

Index

SYMBOLS

- «{}» (curly brackets). *See* curly brackets
- «()». (empty sequence). *See* empty sequence
- «[]» (square brackets). *See* square brackets
- «/». *See also* «\» (backslash)
 - binary operator, 602–606
 - associativity of, 605–606
 - axis steps and, 603
 - context node and, 603
 - context position and, 603
 - context size and, 603, 821, 855
 - document nodes and, 617
 - effect of, 603–604
 - examples, 604–605
 - as higher-order operator, 603, 604
 - last() and, 605
 - position() and, 605
 - steps creating new nodes in, 605
 - type error and, 603
 - different roles of, 623
 - disambiguation of, 543
 - division and, 572
 - in expression context, 538, 621
 - operator, 57, 83, 538
 - deduplication and, 544
 - simple mapping expressions and, 599, 601, 644
 - sorting and, 544
 - in parentheses, 624
 - pattern, 692
 - patterns starting with, 689
 - as root expression. *See* root expressions
 - as rooted path expression. *See* rooted path expressions
 - simple mapping expressions and, 599, 601
 - syntactic ambiguities, 623–624
- «/..», **623**. *See also* empty sequence
- «//»
 - «/descendant::» compared with, 628
 - examples, 627–628
 - expensiveness of, 627
 - infix operator, 627
 - leading, 691
 - pseudo-operator, 602, 626–628
 - «.» and, 544
 - last() function and, 703
 - positional predicates and, 627
 - position() and, 703
 - rooted path expressions and, 625
 - XML databases and, 627
- «\» (backslash), **920**
 - character class escapes and, 920–922
 - in embedded XPath, 528
 - in regex, 916
 - in replace(), 863
- «\C» (multicharacter escape), **233, 921**
- «\c» (multicharacter escape), **233, 237, 921**
- «\D» (multicharacter escape), **233, 921**
- «\d» (multicharacter escape), **233, 921**
- «\l» (multicharacter escape), **233, 921**
 - XML 1.0 /XML 1.1, 921
- «\i» (multicharacter escape), **233, 237, 921**
 - XML 1.0 /XML 1.1, 921
- «\S» (multicharacter escape), **233, 921**
- «\s» (multicharacter escape), **233, 921**
 - matching with whitespace, 921
- «\W» (multicharacter escape), **233, 921**
- «\w» (multicharacter escape), **233, 921**
- «\P{prop}» (complementary escape), **233, 922**
- «\p{prop}» (category escape), **233, 922**
- «'» (entity reference), **528, 532, 533, 939**
- «"» (entity reference), **528, 532, 533**
- «	» (entity reference), **528**
- «\$»
 - as regex metacharacter, 233, 926. *See also* metacharacters
 - in replacement string, 863
 - in variable references, 429, 528, 540
- «.» **symbol**
 - «//» and, 544
 - context item expression, 84, 269, 564, 620, 621
 - ContextItemExpr, 269, 528, 543–544
 - decimal point, 528, 789
 - different roles of, 528
 - function body and, 544
 - regex metacharacter. *See* metacharacters
 - separator character, 528
 - as thousands separator, 301
 - undefined, 544
- «,» **symbol**
 - as decimal point, 301–302
 - as grouping-separator. *See* grouping-separator
 - as operator. *See* comma operator

«?» symbol

«?» symbol

- «cast as» and, 655
 - «castable as» and, 655
 - occurrence indicator, 163, 179, 547, 669
 - empty sequence and, 713
 - optional argument v. argument with, 713
 - quantifier (regex), 917–918. *See also* quantifiers (regex)
- «*?» **quantifier (regex), 232.**
- «+?» **quantifier (regex), 232.**
- xsi:nil and, 673
- «|»
- “or” as, 75
 - Pattern construct and, 690
 - union operator. *See* «union» operator
- «*»
- different roles of, 538
 - expression context, 538
 - multiplicative operator, 83, 133, 200, 538, 571
 - on durations, 579
 - precedence, 572
 - NameTest, 440, 696
 - NodeTest, 617
 - occurrence indicator, 163, 179, 547, 669
 - quantifier (regex). *See* quantifiers
- «*?» **quantifier (regex), 232. *See also* quantifiers (regex)**
- «+»
- additive operator, 83, 133, 216, 554, 571
 - binary, 572
 - date/time plus duration, 578–579
 - precedence, 572
 - sum() and, 889
 - unary, 572, 639
 - character, 528
 - leading sign
 - number() and, 851
 - occurrence indicator, 505, 669
 - quantifier (regex). *See* quantifiers
- «+?» **quantifier (regex), 232. *See also* quantifiers (regex)**
- «-»
- additive operator, 200, 571
 - date/time plus duration, 578–579
 - precedence, 572
 - hyphen. *See* hyphens
 - minus operator
 - binary, 572
 - on date/time values, 580
 - unary, 200, 530, 572
 - minus-sign. *See* minus-sign
 - name character, 528
 - need for whitespace and, 538
- «=» **operator, 186, 200, 203, 538, 555, 571**
- «every» expression and, 649
 - existential properties of, 279
 - GeneralComp, 582
 - not() and, 850–851
 - quantification and, 648–649
 - «some» expression and, 649
- «!=» **operator, 200, 203, 538, 555, 571, 584**
- GeneralComp, 582
 - not() and, 850–851
 - usage/avoidance of, 584, 591
- «<», **200, 203, 204, 538, 555, 571**
- character, 527
 - escaped as «<», 538, 588
 - GeneralComp, 582
- «<=» **operator, 200, 203, 538, 555, 571**
- GeneralComp, 582
- «>», **200, 203, 204, 538, 555, 571**
- escaped as «>», 538
 - GeneralComp, 582
- «>=» **operator, 200, 203, 538, 555, 571**
- GeneralComp, 582
- «>», **444, 538**
- «<», **492, 493, 496, 527, 538**
- «<» escaped as, 538, 588
- «<<» **operator, 57, 571, 594**
- description of, 594
 - document order and, 593, 594
 - NodeComp, 582
- «>>» **operator, 57, 571, 594**
- description of, 594
 - document order and, 593, 594
 - NodeComp, 582
- «@» **(attribute axis), 83, 621, 622. *See also* attribute axis**
- self axis and, 622
- «@*», **622**
- «"» **(attribute value), 64**
- «©» **(copyright symbol), 365**
- «..» **(parent axis), 621, 623. *See also* parent axis**
- XPath 1.0 and, 623
- «^» **(regex metacharacter). *See* metacharacters**
- «!» **(separator in jar URI scheme), 868**
- «³» **(superscript three), 849**
- «²» **(superscript two), 849**
- « », **64, 281, 367, 938. *See also* nonbreaking space characters**
- «&» **ampersand, 425**
- curly braces and, 937
 - in URLs, 937
- «'» **attribute value, 64**
- «#», **fractional part of number and, 790**
- «::» **separator, 83**
- « » **character reference, 142, 143**
- 24:00:00 time value, 743**
- «A», **782**

«Å»

Ångstrom symbol v., 849
character normalization and, 848

«^a» (Italian ordinal), 407

« α , β , γ , δ » Greek numbering sequence, 409

A

AbbrevForwardStep construct, 621

abbreviated axis steps, 599, 621–625

root expressions v., 624. *See also* root expressions

abbreviated path expressions, 599

AbbrevReverseStep construct, 621

absence of prefix (QNames), 533, 766, 866

absence of timezone, 567

abs() function, 200, 714–715

absolute path expressions, 625. *See also* rooted path expressions

absolute URIs, 58, 100, 193, 357, 373, 755

in href attribute, 100, 357

resolve-uri() and, 867–869

absolute value of number, 714

abstract types, 163

accents, 202, 462, 561, 731

and diacritics in NFC, 880, 884

Latin letters with, 534

accountancy

parentheses for negative numbers, 301, 790

rounding rules, 790, 873

actions from meeting, numbering of, 415

Adams, Colin, 962. *See also* Gestalt processor

add_extension_function_library() (Gestalt), 962

addition, of durations, 577, 579. *See also* totals, numeric values/durations

additive operators, 200. *See also* «+»

AdditiveExpr construct, 525, 572

adjacent bullets, 335

adjacent elements, repeating groups of, 338–340

adjacent strings, separator between, 258

adjacent text nodes, 143, 258, 304, 309, 496

adjust-date-time-to-timezone() function, 715–718

adjust-date-to-timezone() function, 715–718

adjusting timezone (xs:date/xs:dateTime), 297

adjust-time-to-timezone() function, 205, 715–718

Adler, Sharon, 26

aggregate functions, 200, 332, 350. *See also* avg(); count(); max(); min(); sum()

aggregating list of numbers (example), 993–995

algorithm for URI resolution, 867

aliasing. *See also* xsi:namespace-alias

of namespace URIs, 396

for XML namespace, 397

#all, 242, 485, 486

alphabet example, 122–123

alphanumeric (term), 407

alternate attribute (<?xml-stylesheet?>), 101

AltovaXML processor, 10, 15, 93, 143, 1215–1220.

***See also* examples, running; XMLESpy development environment**

collection() and, 727, 1217

COM API, 1218–1219

command line interface, 1217–1218

conformance, 1216

control over doc() invoking validation, 671

extension instructions and, 476

extensions/extensibility, 1217

input-type-annotations and, 478

Java API and, 1219–1220

MSXML and, 1216

.NET API and, 1220

validating source document on, 169, 1218

validation request, 166, 1218

whitespace and, 1216

XMLValidator, 1218–1219

xsi:schemaLocation attribute and, 671, 1217

XSLT2 object, 1219

A.M. or P.M. components, 782

ambiguities, of «/», 623–624

Ambroziak, Jacek, 10

ampersand «&», 425

curly braces and, 937

in URLs, 937

ancestor axis (XPath), 83, 609, 1221

definition/description, 1221

ancestor-or-self axis (XPath), 609, 610, 1221

definition/description, 1221

ancestors

innermost, 542

of node (example), 325

outermost, 542

RelativePathPattern and, 693

anchored regex, 236, 829, 863

«and» expression, 186, 538, 552

boolean() and, 722

error semantics for, 1144

AndExpr construct, 525, 594

Ångstrom symbol, «Å» v., 849

annotated with (a schema type), 653, 742

annotations. *See also* type annotations

of attributes, 258–260

of elements, 312–315

input-type-annotations, 99

of source documents, 165

annual leave example, 347–348

anonymous atomic types, 669

anonymous types, 164, 313, 673

ANSI, 937

ant build tool, 67, 86

any level. *See* level attribute

AnyKindTest construct, 616, 670, 698

anyURI. *See* xs:anyURI

APIs (application programming interfaces)

APIs (application programming interfaces), 90

- AltovaXML processor
 - COM API, 1218–1219
 - Java API and, 1219–1220
 - .NET API and, 1220
 - Java. *See* JAXP
 - JavaScript, 430
 - Microsoft, 100, 427
 - Saxon. *See* Saxon
 - vendor-defined, 427
- apostrophes, quotation marks and, 533**
- appearance, order of, 248**
- application detects error, 774**
- application programming interfaces. *See* APIs**
- application/xslt+xml, 100, 470**
- apply operator (in functional programming languages), 604**
- arguments, 545. *See also* arity**
- converting, 547–549
 - automatic, with number() function, 573
 - empty sequence as, 713
 - optional, 712, 713
 - arguments with «?» v., 713
 - required type of, 547, 713
 - type checking of, 1143
- arithmetic operators, 197–198, 200, 454, 571–581.**
- See also* «*»; «+»; «-»; «div»; «mod»; numeric functions**
- atomization of operands. *See* atomization
 - changes in XPath 2.0, 573
 - date calculations, 571, 574, 577–581
 - decimal calculations, 575–576
 - duration calculations, 571, 577–581
 - integer calculations, 574–575
 - numeric calculations, 574–577
 - rules for, 573–574
 - backward compatibility and, 573, 574
 - syntax, 571–572
 - time calculations, 574, 577–581
 - type promotion, 572, 585
 - xs:decimal and, 571
 - xs:double and, 554–555, 571
 - xs:float and, 571
 - xs:integer and, 571
- arithmetic overflow, 719**
- arity, 345, 346, 361, 546, 559, 712, 792, 1221. *See also* arguments**
- definition/description, 1221
- arranging data, in tables, 343–344**
- as attribute, 72, 82, 161, 243**
- occurrence indicators and, 179
 - QNames and, 536
 - required type of parameter and, 427–428
 - schema-aware processor and, 161
 - sequence type descriptors and, 162, 164, 668

- type attribute v., 178–179
 - type declarations and, 161, 514
 - value of variable and, 503, 504
 - of xsl:function, 346, 371
 - of xsl:param, 371, 425, 427–428
 - of xsl:template, 371, 487
 - of xsl:variable, 274, 371
 - of xsl:with-param, 371, 518
- ascending order, 459**
- ASCII**
- copyright symbol and, 365
 - escaping non-, 776
 - UTF-8 as superset of, 365
 - UTF-8 encoding and, 365
 - xsl:output encoding, 424
- ASP page, 430**
- ASP.NET**
- and C# (examples), 1092–1094
 - pages, generating HTML with, 1092–1095
- assertions, 568**
- «treat as» operator, 678
- assignment statements, 506, 985–989**
- conditional initialization and, 989–990
 - order of execution and, 986
- Associating Style Sheets with XML Documents, 444.**
- See also* <?xml-stylesheet?> processing instruction**
- associative access to nodes, 813**
- associative property**
- of binary «/», 605–606
 - of concatenation, 635
- atoms (regex construct), 917–918**
- as back-references, 917
 - character class escapes as, 920
 - as character classes, 917
 - «i» flag and, 925
 - as normal characters, 917
- atomic types, 53, 153, 163, 193–217. *See also* type(s)**
- anonymous, 669
 - built-in. *See* built-in atomic types
 - derived numeric types. *See* derived numeric types
 - derived string types. *See* derived string types
 - major, 193–206
 - minimum set (XSLT 2.0) of, 669–670
 - minor, 193, 206–210
 - primitive. *See* primitive types
 - as required item type, 506
 - SequenceType and, 371
 - simple types v., 191–192
 - sources of, 192
 - untyped atomic values, 193, 215–217
 - user-defined. *See* user-defined atomic types
- atomic values, 52, 54, 71, 152, 153, 162, 163, 164, 178, 189–191, 506, 1221. *See also* booleans; xs:double; xs:integer; xs:string**

- booleans as, 189, 191
- converting, 654–668
- converting, to strings, 878–879. *See also* string()
- definition/description, 1221
- document() applied to, 760–761
- floating-point numbers as, 189, 198
- identity and, 190, 289
- integers as, 189
- matching, sequence type descriptors and, 669–670
- properties of, 190
- returning, 453, 455
- sequences and, 187–188, 322, 351, 633
- sets of, 631
- simple mapping expressions and, 644–645
- strings as, 189, 190
- URLs as, 760
- xs:integer, 191
- xs:untypedAtomic, 191
- AtomicType construct, 655, 668**
- atomization (XPath), 52, 81, 165, 190, 220, 258, 262, 427, 496, 506, 548, 1221**
- and arithmetic operators, 573
- automatic, 743
- «cast as» operator and, 655
- «castable as» operator and, 655
- conditional expressions and, 551
- as conversion, 654
- data() and, 741–742
- definition/description, 1221
- of list-valued attribute, 353
- of operands, 573, 586
 - empty sequence and, 573, 586
 - type error and, 573, 586
 - value comparisons, 586
 - xs:untypedAtomic and, 574, 586, 589
- predicates with nodes and, 638
- runtime error and, 499
- schema validation and, 742
- attribute(), 617, 698**
- KindTest, 617
- matches any attribute node, 673
- NodeTest, 162
- schema-defined types and, 617
- attribute axis, 83, 609, 1221**
- «@» and, 83, 621, 622
- NameTest and, 695
- node tests and, 613
- ordering of nodes on, 620
- PatternAxis and, 695
- in patterns, 683
- positional predicates and, 702
- attribute declarations (in schemas), 1221**
- global/top-level. *See* global attribute declarations
- user-defined, 369
- attribute (N, T), 371**
- attribute names, 255–256. *See also* name(); node-name()**
- changing, 391
- decided at runtime, 255, 260, 264–266
- duplicate, 268, 309
- prefix of, 256
- unprefixed names and, 482, 837
- attribute nodes, 49, 56, 1221–1222. *See also* specific attributes**
- atomization of list-valued, 353
- calculation of value of, 260
- copying, 260–261
 - all, 261, 296
 - unconditionally, 290
- creating, 253. *See also* xsl:attribute
- attribute value template and, 260
- different ways for, 260–261
- string value in, 258
- validating/annotating in, 258–260
- xsl:attribute and, 260
- xsl:copy/xsl:copy-of and, 260–261
- xs:QName and, 261–263
- declarations. *See* attribute declarations
- default, 65
- definition/description, 1221–1222
- document order for, 610
- generation of, before child elements, 117, 257
- KindTests for, 699–700
- of literal result elements, 112–114
- local, 260
- matching, 672–677
- names. *See* attribute names
- namespace declarations and, 610
- namespace prefixes in content of, 392
- namespaces v., 256
- normalization, 64, 142, 532
- order of, 64, 241, 323
 - deep-equal() and, 747
 - undefined, 316
- parentless, 257
 - patterns and, 683, 684, 688–689
- processing, 241
- with QName-valued content, 261, 393
- same name of, 257, 267
- schema-defined types of, 622
- string value, 51
- validation, 255
- vendor-defined, 134
- on xsl:stylesheet, 98–99
- attribute sets (XSLT), 97, 105, 113, 117, 257, 261, 361, 1222. *See also* xsl:attribute-set**
- cycles among, 352–353
- definition/description, 1222
- duplicate, 268
- for generated HTML table elements, 269

attribute sets (XSLT) (continued)

attribute sets (XSLT) (continued)

- literal result elements and, 117
- merging, 267
- for numbering, 270–271
- parameterize attribute values in, 268
- xsl:include with, 375–376

attribute value templates (XSLT), 90, 113, 117, 228, 255, 308, 410, 1222

- backward-compatibility mode and, 122
- curly brackets { } in, 113, 117
- doubling of, 123
- nesting and, 123
- text between, 117
- XPath expressions within, 117, 122, 123
- definition/description, 227, 1222
- escaping of special characters in, 123
- nonrecognition of, 124
- places/context for using, 123–124
- restrictions on, 124–125
- string-join() v., 880
- xsl:attribute v., 122
- in xsl:result-document, 448
- in xsl:sort, 330
- [xsl:]type attribute and, 124

AttributeName construct, 672

AttributeNameOrWildcard construct, 672

attribute-or-top axis, 689

AttributeTest construct, 616, 672, 697

automatic atomization, data () and, 743

automatic conversion, of arguments to numbers, 573

average of numbers/durations, 718

avg() function, 200, 718–719

- one-or-more() and, 854

axis name, defaulting of, 621–622

axis specifier, 695

axis steps, 606–625

- abbreviated. *See* abbreviated axis steps
- binary «/» and, 603
- document order/duplicates and, 602
- evaluation of, 607–608
- examples of, 608–609
- filter expressions v., 542
- filtering and, 606
- node tests in, 606, 613–614
- NodeTests and, 607
- path expressions and, 599
- predicates in, 542, 552, 606, 617
- predicates in filter expressions v., 639–640
- QNames in, 536
- result of, 607
- syntax, 606–607

axis/axes (XPath), 83, 606, 609–612, 1222. *See also* specific axes

- ancestor, 83, 609, 1221
- ancestor-or-self, 609, 610, 1221
- attribute. *See* attribute axis

attribute-or-top, 689

child. *See* child axis

child-or-top, 689

definitions/descriptions of, 606, 609–612, 1222

descendant, 609, 610, 1223

descendant-or-self, 609, 611, 1223

direction of, 242, 323, 620

following, 609, 611, 1225

following-sibling, 83, 609, 611, 1225

forward, 607, 608, 620

list of, 609

namespace. *See* namespace axis

parent, 609, 612, 621, 1227

«..» as, 621, 623

path expressions and, 599. *See also* path

expressions

in patterns, 683

preceding, 609, 612, 1227

preceding-sibling, 83, 609, 612, 1227

reverse, 607, 608, 620, 1227, 1228

self, 609, 612, 622, 836, 1228

AxisStep

NameTests in, 482

primary expressions v., 540

zero or more predicates in, 620

B

back-references, 917, 918

atoms as, 917

captured groups and, 918

character class escape and, 921

«i» flag and, 925

backslash («\$»), 920

character class escapes and, 920–922

in embedded XPath, 528

in regex, 916

in replace(), 863

backward-compatibility mode, 82, 90, 91, 113,

131–134, 377, 479, 496, 1139–1145

attribute value templates and, 122

automatic conversion of arguments to numbers and, 573

comparing strings and, 1141

definition of, 1231

disable-output-escaping and, 1140. *See also*

disable-output-escaping

effective boolean value and, 722

error handling and, 1140–1141

‘first item’ rule, 132, 133, 480, 498, 1142–1143

function conversion rules and, 549

function-available() and, 792

general comparisons and, 588, 589

HTML result tree and, 424

number() and, 852

numeric formats and, 1141

- rules for arithmetic operators and, 573, 574
- sequence numbers and, 405
- serialization and, 1142
- standard conversion rules and, 505, 714
- static context, 554–555
- string() and, 877
- as transition aid, 554
- type checking and, 549
- value comparisons and, 586
- version attribute, 479, 496
 - XSLT 2.0 rules for, 479
- xsl:sort in, 459
- base URIs, 84, 293, 357, 358, 373, 1222. See also xml:base attribute**
 - collection() function and, 562
 - of context node, 720
 - deep-equal() and, 747
 - definition/description, 756, 1222
 - doc() function and, 562, 751
 - document() function and, 757, 760, 763
 - DOM and, 756, 876
 - nodes, 54–55
 - for output documents, 68
 - static context, 562, 868, 876
 - static-base-uri() function and, 562, 868, 876–877
 - of stylesheet, 720, 756, 905
- base64Binary. See xs:base64Binary**
- base-uri() function, 55, 719–720**
 - doc() and, 750
 - resolve-uri() and, 720
- basic XSLT processors, 151, 190. See also schema-aware XSLT processors**
 - built-in types in, 162, 190, 192
 - in-scope schema definitions and, 557
 - xsl:import-schema declarations and, 370
- BasicAttributeTest construct, 672, 698**
- BasicElementTest construct, 672, 697**
- Berners-Lee, Tim, 26**
- binary «/» operator. See «/»**
- binary operators, 602. See also «except»; «intersect»; «union»**
- binary types, 208. See also xs:base64Binary; xs:hexBinary**
- binding (extension functions), 957–962**
 - external objects and, 961–962
 - in Gestalt, 962
 - in Java processors, 959–961
 - in MSXML, 957–959
 - in Saxon, 962, 1205–1207
 - in System.Xml.Xsl processor, 957, 961–962
- block-structured languages, 502**
- BNF, extended, 522, 689**
- boilerplate generation, HTML, 909–910**
- book list**
 - current() and, 736–738
 - document() and, 757–758
 - look-up table in stylesheet example, 761–763
 - embedded stylesheet example, 102–104
 - simplified stylesheet and, 125–127
 - keys as cross-references, 814–816
 - nonunique keys example, 381–383
- boolean constants, 779, 899. See also false(); true()**
- boolean grouping key, 338**
- boolean operators, 594–596. See also «and»; «or»**
 - shortcut semantics, 595–596
- boolean predicates, 618, 638**
 - effective boolean value and, 638
- boolean() function, 200, 354, 551, 552, 595, 619, 638, 721–722. See also effective boolean value**
 - effective boolean value and, 721
 - [fn:] prefix and, 552
 - «if» expression and, 722, 771
 - «some», «every», «and», «or», expressions and, 722
 - xs:boolean() constructor function v., 552
 - xsl:if and, 722
 - xsl:when and, 722
- booleans, 52, 184, 186, 187, 189, 191, 194–195, 1222**
 - atomic values, 191
 - conversions to, 552
 - converting from decimal, 659
 - converting from double, 660
 - converting from float, 661
 - converting from string, 552, 663
 - converting to other types, 657–658
 - definition/description, 1222
 - effective value. *See* effective boolean value
 - «eq» and, 583
 - false value, 189, 191, 194, 195
 - lexical representations of, 552
 - «lt», «le», «ge», «gt», and, 585
 - sequence of nodes compared to, 589
 - true value, 189, 191, 194, 195
 - xs:untypedAtomic value converted to, 552
- Bosak, Jon, 247, 335, 996. See also Shakespeare plays**
- bottom-up process, of tree construction, 73**
- Bouvier, Jacqueline Lee, 1051. See also family tree**
- brackets. See curly brackets; square brackets**
- branches (regex construct), 916**
- browsers, 100. See also Firefox; Internet Explorer; Konqueror; Opera; Safari**
 - different versions of XSLT and, 318
 - escaping URIs and, 776
 - generating HTML in, 1095–1098. *See also* family tree stylesheets running in, 430, 463, 957, 1154–1157
- built-in atomic types, 163, 164, 192–193, 193–217. See also derived numeric types; derived string types; primitive types**
 - basic XSLT processor, 162, 190, 192
 - constructor functions for, 546
 - schema-aware XSLT processor, 190, 192

built-in derived types

built-in derived types, 664, 665. See also

xs:dayTimeDuration; xs:integer;

xs:yearMonthDuration

in basic XSLT processor, 192

conversion rules, 665

numeric, 192, 193, 210–213

string, 192, 193, 213–215

built-in template rules (XSLT), 69, 73, 78–79, 238, 242, 400, 486, 1222

for comment nodes, 700

definition/description, 1222

node kind, 79

built-in types, 161, 259, 312, 371, 1222

definition/description, 1222

in schema-aware processors, 190, 192,

557, 670

Bulgarian language, 409

bullets, adjacent, 335

byte-order-mark, 421, 422, 448, 905

C

C#, 7, 8, 9, 192, 212, 532, 543, 546

ASP.NET examples, 1092–1094

extension functions in, 957

and Saxon on .NET, 957, 1093, 1203–1204

user-defined functions in, 546

C++, 136

C programming language, 187, 502

#include directive in, 374

cached schemas, 369

compiled, 370

calculations, numeric. See arithmetic operators;

numeric calculations; numeric functions

calculus, predicate, 641

calendar argument, 784–787

calendar component, 782

calendars, 784–787

Gregorian, 784, 785, 786

Islamic, 787

ISO, 785, 786

Julian, 786, 787

non-Gregorian, 786

call stack, 429. See also stack frame

call-by-value parameters, 273

caller's context, 349

calling. See also function calls

constructor functions for, 546

core function library for, 546

extension functions, 956–957

extension functions within loop (example), 965–967

functions for, 546

Java methods, 546, 959–961

Java methods from Saxon, 1206

user-defined functions for, 546

vendor-supplied functions for, 546

XSLT functions for, 546

calling conventions, function, 959

canonical form, 64–65

pattern facet and, 667

XML documents converted to, 64–65

Canonical XML, 64

captured groups, 234, 860, 862

back-references and, 918

captured substrings, 860

cardinal numbers, 409, 783

cardinality

required, 505, 547

in sequence types, 653

Carducci String Quartet examples, 488–491

carriage return (whitespace character), 440

carriage return, line feed (CR-LF), 64

Cascading Style Sheets. See CSS

case mappings, 828, 910. See also lower-case();

upper-case()

locale-sensitive, 828

case studies

family tree, 1049–1098

knight's tour, 221–222, 224, 1099–1114

XML specification, 1001–1048

case variants, 925, 926

lower-case() and, 926

upper-case() and, 926

case-insensitive comparison, 279, 828

collation in, 828

«i» flag and, 828, 925

lower-case() and, 828

in regex, 828, 925

case-insensitive mode (regex evaluation), 231, 925

«i» flag and, 925–926

case-order attribute, 456, 459, 460

upper-case/lower-case letters and, 461

case-sensitive names, in XPath expressions, 535

«cast as» operator, 538, 560

atomization and, 655

constructor function and, 655

question mark and, 655

syntax, 655

«castable as» operator, 538

atomization and, 655

number() and, 853

question mark and, 655

syntax, 655

CastableExpr construct, 526, 655

CastExpr construct, 526, 655

casting. See conversions

casts (XPath), 1222. See also conversions

catalogs, 369

categories, character. See character categories

category escape, 922

 charset attribute (<?xml-stylesheet?>)

- CDATA sections, 64, 67, 422, 440, 446, 909, 1222**
 - CDATA-section-elements and, 933
 - character maps and, 941, 942
 - definition/description, 1222
 - disable-output-escaping and, 947
- CDATA-section-elements, 421, 422, 446, 448, 933**
 - CDATA section and, 933
 - XML output method and, 933
- ceiling() function, 200, 723–724**
 - negative zero and, 723
- century, 783**
- changing**
 - context item, 274, 323, 325–326, 543
 - element/attribute name, 391
 - focus, 565
 - namespace URI, 296
 - prefixes, 310–311
- character(s), 62–63. See also specific characters**
 - ampersand. *See* ampersand
 - categories. *See* character categories
 - combining, 849, 880
 - normalization of, 880, 884
 - compatibility variants of, 849
 - control, 63
 - converting. *See* lower-case(); upper-case()
 - first, of string, 884–885
 - ignored in collations, 886, 888
 - invalid, IDs and, 62
 - line ending. *See* line ending characters
 - meta. *See* metacharacters
 - newline. *See* newline characters
 - private use area (Unicode), 287, 943
 - removal of, 897–898
 - replacement, 897. *See also* translate()
 - special. *See* special characters
 - in string value, 63, 884
 - substitution of, 897–898
 - surrogate pair. *See* surrogate pair
 - Unicode. *See* Unicode
 - unreserved, 772
 - whitespace. *See* whitespace
- character attribute (xsl:output-character), 425**
- character blocks, 922–923**
 - (list), 922–923
- character categories, 924**
 - (list), 924–925
- character class escapes, 917, 920–922**
 - as atoms, 920
 - back-reference and, 921
 - backslash «\» and, 920–922
 - in square brackets, 920
- character class expression, 917**
- character classes, 917**
 - atoms as, 917
- character encoding, 447, 527, 905. See also Unicode**
 - ASCII
 - copyright symbol and, 365
 - US, 424
 - UTF-8 as superset of, 365
 - %HH, 422, 447, 771, 772, 776, 811, 936, 938, 940
 - iso-8859–1, 365, 424
 - UTF-8, 63, 64, 906
 - UTF-16, 63
 - in xsl:output, 422
 - in xsl:result-document, 447
- character groups, 919**
 - character ranges in, 919
 - negative group, 919
 - positive group, 919
 - square brackets and, 919
 - subtraction, 919
- character maps (XSLT), 280, 287, 425, 941–945, 1222. See also xsl:character-map**
 - CDATA sections and, 941, 942
 - 'commenting out' elements (example), 943–944
 - composite, 282
 - definition/description, 1222
 - disable-output-escaping v., 948–949
 - entity references and, 942
 - internal DTD subset and, 942
 - limitations of, 944–945
 - merging, 282
 - name of, 281
 - no effect, 282
 - nonbreaking space character and, 943
 - serialization and, 280, 282, 941
 - SGML and, 942
 - use-character-maps and, 447, 935, 941
 - xsl:output-character and, 425
- character maps and, 943**
- Character Model for the World Wide Web 1.0: Normalization, 848, 849**
- character normalization, 848–849**
 - «Å» and, 848
 - normalize-unicode() and, 848–849
- character ranges (regex), 919–920**
 - in character groups, 919
 - as codepoint range, 920
 - hyphen and, 919
 - «i» flag and, 925
 - names for, in Unicode. *See* character blocks
 - positive groups and, 919
 - Unicode codepoints and, 919
- character references (XML), 64, 142, 143, 367, 443, 444, 532, 880, 1222. See also entity references**
 - definition/description, 1222
 - encoding attribute and, 933
 - special characters converted to, 493
- charset attribute (<?xml-stylesheet?>), 101**

charset attribute (HTML meta element)

- charset attribute (HTML meta element), 938, 939, 940, 941**
- chessboard, knight's tour of (case study), 221–222, 224, 1099–1114**
- chevrons, 522, 689, 915**
 - patterns and, 689
 - regex and, 915
 - XPath and, 522
- child axis (XPath), 83, 609, 610, 621, 1222**
 - definition/description, 1222
 - KindTest and, 695
 - NameTest and, 695
 - node() and, 671
 - PatternAxis and, 695
 - in patterns, 683
- child-or-top axis, 689**
- children, of nodes, 55**
 - attributes generated before, 117, 257
 - parent relationship v., 55, 623
- choosing. *See also* `xsl:choose`**
 - prefixes, 262
 - prevention of conflicts and, 308
 - template rules, 242–245, 485–486
 - XPath-level work v. XSLT-level work, 455
- circularity, 246, 267, 282, 378, 485, 501**
 - patterns and, 378, 485, 683
 - variable references and, 683
- circumflex, 920**
- cities grouped by country, 817–818**
- Clark, James, 25, 28, 59, 201, 550, 844, 865, 986.**
 - See also* `xt processor`
 - on DSSSL, 28, 986
- Clark notation, 59, 844, 865**
 - node-name() example, 844–845
 - resolve-QName() example, 865
- class hierarchy**
 - UML, 50, 56–57
 - `xsl:import`, 95
- client side, 100. *See also* browsers**
 - JavaScript, 286, 957
 - script, 957
 - extension functions and, 957
 - transformation
 - `<?xmlstylesheet?>`, 100
- closure (property of languages), 9, 42, 185**
- codepoint collation. *See* Unicode codepoint collation**
- codepoint range, 920**
 - Unicode codepoint and, 920
- codepoint-equal() function, 724**
 - comparing strings and, 724
 - comparing URIs and, 724
 - `xs:anyURI` and, 724
- codepoints (Unicode), 233, 535, 724, 725, 773, 1222**
 - character ranges and, 919
 - definition/description, 1222
 - `string-to-codepoints()` and, 881–882
- codepoints-to-string() function, 725–726. *See also* `string-to-codepoints()`**
- collapsing whitespace, 143, 214**
- collating sequences. *See* collations**
- collation attribute, 329, 377, 456, 457, 460**
- collation units, 731**
 - ignorable, 886, 888
- collation URIs, 327, 378, 457, 460, 470**
 - Saxon, 565, 1207–1208
- collations (XPath), 103, 329, 456, 459–461, 461–462, 1222. *See also* sorting**
 - case-insensitive comparison and, 828
 - characters ignored in, 886, 888
 - codepoint. *See* Unicode codepoint collation
 - decomposition and, 731
 - default. *See* default collation
 - definition/description, 1222
 - domain-specific collating rules in, 461
 - «eq» operator and, 459, 584
 - in Germany, 459
 - identified by URIs, 460, 561
 - implementation-defined, 954
 - in Java, 561, 731–732
 - matching substrings with, 731–732, 876
 - minimal, 886, 888
 - ordering of strings by, 584
 - in Saxon, 1207–1208
 - sorting and, 106, 459
 - static context and, 561–562
 - strong, 378, 461, 462, 562, 584
 - substring matching with, 731–732
 - in Sweden, 459
 - UK phone book and, 462
 - URIs, 460, 560
 - URIs, in Saxon, 561
 - varying by country, 459
 - weak, 378, 461, 562, 584
 - in the XPath context, 84
 - XQuery and, 562
- collection URIs, 727, 1197, 1217**
- collection() function, 44, 440, 477, 568, 726–727. *See also* `doc()` function**
 - AltovaXML and, 727, 1217
 - base URI and, 562
 - document collection and, 568
 - document-uri() and, 727, 764
 - dynamic context and, 567, 568, 726
 - Gestalt, 727
 - Saxon, 727
- collections**
 - of documents, 568, 726
 - default, 726
 - dynamic context and, 567, 568
 - statically known, 562–563
 - XML databases and, 568, 726

1242

- CollectionURIResolver (Saxon), 727, 1197**
 - color example, corporate, 363**
 - COM API**
 - AltovaXML, 1218–1219
 - MSXML, 1147–1153
 - combining character, 849, 880**
 - normalization of, 880, 884
 - combining sets of nodes. See sets of nodes**
 - comma**
 - as decimal point, 301–302
 - as grouping-separator. *See* grouping-separator
 - comma operator («,»), 189, 261, 525, 537, 561, 633, 634–636**
 - ExprSingle and, 634
 - function calls and, 525, 545
 - list construction with, 635
 - max() function and, 635, 831
 - no sorting into document order, 635
 - parentheses and, 525, 542
 - specified order and, 635
 - comma-separated lists, 189**
 - values enclosed in parentheses, 189
 - comma-separated-values file, 44, 906, 907–908**
 - comment(), 671, 698, 700**
 - KindTest, 613
 - «comment» keyword, 616**
 - comment nodes, 49, 64, 285–287, 1222. See also xsl:comment**
 - comment() and, 671
 - debugging stylesheets and, 286
 - deep-equal() and, 746
 - default template rule for, 700
 - definition/description, 1222
 - with hyphen, 285
 - with markup, 287
 - string value, 51, 285
 - string value of new, 285
 - in stylesheet, 287
 - writing to result sequence, 285
 - commented-out JavaScript, 286–287, 943**
 - comments (regex)**
 - «x» flag and, 926
 - comments (XPath)**
 - lexical rules and, 524
 - nested, 529
 - syntactic rules and, 524
 - CommentTest construct, 616, 670, 698**
 - compare() function, 203, 584, 727–729. See also collations; value comparisons**
 - comparing. See also matching**
 - «//» and «/descendant::», 628
 - dates/times, 580
 - node names, 859. *See also* node-name(); QName()
 - nodes. *See* node comparisons
 - numeric values, 583, 585
 - sequences
 - deep-equal() and, 745–748
 - general comparisons and, 588–592
 - single atomic values. *See* value comparisons
 - strings, 84, 724, 727. *See also* codepoint-equal(); compare()
 - backward-compatibility mode and, 1141
 - default collation and, 588, 724, 728, 731
 - incompatibility and, 588
 - substrings, 730–732. *See also* contains() URIs, 724
- ComparisonExpr construct, 525, 582**
- comparisons. See existential comparison; general comparisons; node comparisons; value comparisons**
- compatibility mode. See backward-compatibility mode; forward-compatibility mode**
- compatibility variants, of characters, 849**
- compile time**
 - conditional logic, 97–98
 - prefixes not known at, 263
 - schemas/runtime schemas, 752
 - type errors at, 506
 - type information at, 590
 - URIs not known at, 263
 - use-when evaluation at, 478
- complementary escape, 922**
- complex types (in schema), 154–156, 163, 164, 313, 371, 1223**
 - definition/description, 1223
 - with element-only content, 53, 142, 156–157
 - schema-aware processor and, 499
 - with mixed content, 155
 - schema types as, 653, 742
 - with simple content, 154–155
 - units of measure as, 213
- component, schema, 557, 1228. See also attribute declarations; element declarations; type definitions**
- component specifier, 782**
- composability, 185. See also closure**
- composite character maps, 282**
- composite grouping keys, 334**
- composite sequence numbers, 407, 419**
- computational power, of XSLT, 1099, 1112**
- computational stylesheets (design pattern), 985–1000, 1114. See also knight's tour (case study)**
- concatenation, 730**
 - associative property of, 635
 - normalized strings and, 849
 - separator and, 879. *See also* string-join() of sequences, 261, 634. *See also* comma operator
- concat() function, 123, 204, 545, 729–730**
 - keys and, 730
 - normalize-unicode() and, 849
 - string-join() v., 880
 - xsl:value-of v., 730
- concrete types, 163**

conditional compilation

conditional compilation, 97–98, 112, 127–128

conditional expressions (XPath), 283, 551–553. See

also if-then-else

atomization and, 551

errors and, 552

as sort key, 462

substring() as, 885

xsl:choose v., 552–553

conditional initialization, 989–990

Configuration.buildDocument() (Saxon), 1205

conflict resolution policy, 69, 73, 79–80, 686–688

conflicts, priority and, 440

consistency rules

for schemas, 370

XDM, 310

constants, 506. See also global variables; literal

constants

constraints

identity, 179

integrity, result document, 174

construction of lists, 635. See also comma operator

constructor functions (XPath), 80, 82, 135, 371, 482,

506, 1223

atomic types and, 544, 546

for calling, 546

«cast as» expressions as, 655

default namespace for elements/types and, 560

definition/description, 1223

empty sequence and, 655

function-available() and, 793

in-scope functions as, 559

user-defined type and, 482

xs:boolean(), 552

xs:integer(), 780

xsl:sort and, 459

contained sequence constructor, 503

contains() function, 204, 730–732

substring matching with collation, 731–732

content

element, 441

empty, 53, 742

mixed, 53, 146, 155–156, 441, 499

namespace-sensitive, 60–61, 260, 310, 311

#PCDATA, 142

simple, 441

ContentHandler (SAX), 44, 1174, 1180

context (XPath), 83, 84–85, 125. See also evaluation

context

dynamic. See dynamic context

importance of, 569

patterns and, 683

static. See static context

use-when attribute, 478

validation, 298

context item, 85, 125, 269, 322, 328, 329, 330,

563–564, 1223

changing, 274, 323, 325–326, 543

copying, 287

current() and, 735–736

definition/description, 1223

«every» expression and, 646

in «for» expression, 642–643

global variable and, 735

no

«/» and, 624

rooted path expressions and, 625

path expressions and, 543, 644

predicates and, 543, 565, 620, 639

«some» expression and, 646

stylesheet functions, 350, 736

undefined, 350, 544, 736

xsl:analyze-string and, 232, 736

xsl:apply-templates and, 543, 735

xsl:call-template and, 273

xsl:for-each and, 543, 735

xsl:for-each-group and, 736

xsl:function and, 346

xsl:sort and, 457

context item expressions, 543–544

«.» as, 84, 269, 564, 620, 621

“context item is not a node” error, 232, 624, 625

xsl:analyze-string and, 624

xsl:for-each and, 624

context node, 83, 241, 564, 1223

base URI of, 720

binary «/» and, 603

definition/description, 1223

«for» expression and, 642–643

position of, 241

predicates and, 608

xsl:template and, 486, 487

context position, 85, 125, 269, 328, 329, 330, 563,

1223. See also position() function

binary «/» and, 603

definition/description, 1223

in «for» expression, 644

position() function and, 564, 608, 620, 639,
854–856

predicates and, 608, 620, 639

xsl:apply-templates and, 564

xsl:for-each and, 564

xsl:function and, 346

xsl:sort and, 457

xsl:template and, 486, 487

context size, 85, 125, 269, 328, 329, 330, 563, 1223.

See also last() function

binary «/» and, 603, 821, 855

definition/description, 1223

in «for» expression, 644

- last() function and, 564, 608, 620, 639, 820–824
- predicates and, 608, 620, 639, 821
- xsl:apply-templates and, 564, 821
- xsl:for-each and, 564, 821
- xsl:function and, 346
- xsl:sort and, 457
- xsl:template and, 486, 487
- context variables, 510**
- ContextItemExpr “.”, 269, 528, 543–544**
- context-sensitive values, 507–510**
 - lower-case() and, 828
- continuation marker, 330**
- control characters, 63**
- controlling**
 - sequence of processing, 77
 - serialization, 67, 450
 - validation, 450
- convenience variables, 507**
 - global, 507
 - local, 507
- conversion rules, standard. See standard conversion rules**
- conversions, 82, 207, 244, 506, 653. See also type expressions**
 - of arguments, 547–549
 - automatic, with number() function, 573
 - of atomic values, 654–668
 - atomization as, 654
 - to boolean, 552
 - between derived types, 664–668
 - of durations into number of months/days/seconds, 581
 - failed, 654
 - function. *See* function conversion rules; standard conversion rules
 - of IRIs to URIs, 811–812
 - of non-string values to derived types, 666–667
 - of numbers to strings, 531
 - of operands to strings, 586
 - between primitive types, 656–664
 - list of source/result types, 656–657
 - to strings, 499
 - of strings to booleans, 552
 - strong. *See* strong conversions
 - type, 377
 - type names in, 482
 - of upper-case characters to lower-case characters, 827–828
 - to user-defined atomic type, 371
 - weak, 81–82, 220
 - from xs:date, 207
 - from xs:dateTime, 207
 - to xs:dayTimeDuration, 668
 - to xs:double, 454, 852
 - xs:untypedAtomic and, 574, 589
 - to xs:integer, 667–668
 - xsl:function and, 346
 - xsl:sort and, 459
 - of xs:string to derived types, 665–666
 - of xs:untypedAtomic, 427
 - of xs:untypedAtomic values to booleans, 552
 - to xs:yearMonthDuration, 668
- copying/copies. See also xsl:copy; xsl:copy-of**
 - attributes, 260–261
 - all, 261, 296
 - unconditionally, 290
 - context item, 287
 - data to/from temporary tree, 293
 - deep, 287, 292, 293, 296. *See also* xsl:copy-of recursive, 290
 - exact, 296
 - namespace nodes, 296–297, 311
 - nodes, 57
 - validation of, 297
 - of parent element’s namespaces, 310
 - shallow, 287
 - source document to result, 293, 296
 - with few changes, 290
 - type annotations, 297–298
- copy-namespaces, 289, 296, 392, 474**
- copy-namespaces = “no”, 291**
- copyright symbol (©), 365**
- core function library, 135, 709–913**
 - for calling, 546
 - functions by category (list), 710–712
 - in-scope functions and, 559
 - minimum set of atomic types and, 669
 - standard conversion rules, 713–714. *See also* standard conversion rules
- corporate color example, 363**
- count attribute (xsl:number), 411–412**
- counter, 990**
- count() function, 146, 212, 350, 411, 733–734**
 - last() v., 734
 - xsl:number v., 734
- countries/languages, xsl:decimal-format and, 301. See also format-date()**
- country argument, 787**
- cp1252, 937**
- create_iterator() (Gestalt), 962**
- CR-LF. See carriage return, line feed**
- cross references**
 - in case study, 1007, 1029–1032
 - construction of, 412
 - keys as, 814–816
- CSS (Cascading Style Sheets), 24, 444, 489**
 - IdKeyPattern and, 705
 - rule-based stylesheets and, 980
 - XSLT vs., 24, 982
 - family tree case study, 1080, 1081

curly brackets ({})

curly brackets ({})

- ampersand and, 937
- in attribute value templates, 113, 117
 - doubling of, 123, 915
 - nesting and, 123
 - text between, 117
 - XPath expressions within, 117, 122, 123
- in regular expressions, 231, 915, 916
- #current, 240, 242, 485, 486**
- current date, 785**
- current group, 346, 739**
 - stylesheet function and, 739
- current grouping key, 346, 740**
- current mode (XSLT), 238, 242, 273, 346, 400, 485, 486, 1223**
 - definition/description, 1223
- current template rule (XSLT), 238, 273, 346, 399–400, 1223**
 - definition/description, 1223
 - xsl:apply-imports and, 399
 - xsl:apply-templates and, 399
 - xsl:call-template and, 400
 - xsl:for-each and, 400
 - xsl:next-match and, 399–400
- current-date() function, 738–739**
 - dynamic context and, 566
 - implicit timezone and, 583
- current-dateTime() function, 738–739**
 - dynamic context and, 566
- current() function, 85, 509, 734–738**
 - context item and, 735–736
 - example, 736–738
 - patterns and, 683, 736
- current-group() function, 328, 331, 332, 739. See also xsl:for-each-group**
 - xsl:for-each-group and, 739
 - xsl:sort and, 739
- current-grouping-key() function, 328, 331, 740–741. See also xsl:for-each-group**
- current-time() function, 738–739**
 - dynamic context and, 566
 - GMT and, 717
- customization layer, 368, 429**
- customizing**
 - of presentation, 368
 - of stylesheet, 429
 - Othello example, 433–437
- cycles, 251**
 - among attribute sets, 352–353
 - in graphs (examples), 251–253, 352–353
 - xs:time and, 585
- Cygnin, 1196**
- Cyrillic alphabet, 409**

D

- data dictionaries, 369**
- data model, 42. See also XDM tree model**
 - role of, 185
- databases**
 - products, 67
 - relational, 8, 67. *See also* SQL
 - Unicode, 233, 828, 922
 - XML, 568
 - «//» and, 627
 - collections and, 568, 726
 - unordered() and, 902
- data() function, 52, 741–743. See also type matching rules**
 - atomization and, 741–742
 - test contents of node with, 552
 - typed values and, 741
- data-oriented XML, 53, 977**
- data-type, 456, 458. See also xsl:sort**
- dates, 195–197. See also format-date() function; xs:date; xs:dateTime**
 - arithmetic on, 571, 574, 577–581
 - current, 785. *See also* current-date(); current-dateTime()
 - difference between two, 580
 - formatting, 195, 235, 738–739, 781. *See also* format-date(); format-dateTime()
 - Gregorian, 196
 - ISO format, 785
 - language argument and, 784
 - negative, 196
 - partial date types, 207
 - partially ordered, 196, 585
 - starting instants of
 - simultaneous, 583
 - xs:date minus xs:date and, 580
 - subtracting, 348
 - with timezone, 583
 - value space of, 786
- dates/times, 577. See also xs:dateTime**
 - comparing/subtracting, 580
 - formatting. *See* formatting
 - interval between two, 577
 - plus duration, 578–579
 - «+» and «-», 578–579
 - without timezone, 580
- dateTime. See xs:dateTime**
- date/time functions, list of, 711**
- dateTime() function, 743–744**
- day component, 744, 745**
 - from xs:date and xs:dateTime, 744
 - from xs:duration, 745
- day, New Year's, 787**
- day-from-date() function, 744–745**

- day-from-dateTime() function, 744–745**
- daylight savings time, 787**
- days of month, 744**
 - picture argument and, 782
 - from xs:date, 744
 - from xs:dateTime, 744
- days of week, 581, 782, 786. See also format-date()**
 - language for, 784
 - numbering of, 785
- days-from-duration() function, 745**
- debugging, 161, 388**
 - extension-element-prefixes and, 477
 - stylesheet, 431
 - comments and, 286
- decimal. See xs:decimal**
- decimal arithmetic, 575**
- decimal digits, surplus, 531**
- decimal notation. See xs:decimal**
- decimal places, 531**
 - round-half-to-even() and, 872–873
- decimal point «.», 528, 789**
 - «.» symbol as, 301–302
 - digits after, 790
- decimal-format. See also format-number();
xsl:decimal-format**
 - default, 299, 789
 - examples, 791–792
 - name, 299, 789
- DecimalLiteral symbol, 530**
- decimal-separator**
 - comma as, 301–302
 - picture string, 300
 - xsl:decimal-format, 300
- declaration, XML. See XML declaration**
- declarations (in schemas)**
 - attribute. See attribute declarations
 - element. See element declarations
- declarations (in stylesheets), 89, 90, 94, 104, 149,
298, 318, 1223. See also namespace
declarations; type declarations**
 - definition/description of, 1223
 - duplicate, 97, 374
 - implementor-defined, 106–107
 - import precedence of, 357, 361–362
 - local. See local declarations
 - multiple xsl:import-schema, 370
 - order of, 105, 360, 486
 - overriding, 364
 - xsl:attribute-set, 468
 - xsl:character-map, 468
 - xsl:decimal-format, 298, 468
 - xsl:function, 345, 469
 - xsl:import, 469
 - xsl:import-schema, 368, 469
 - xsl:include, 469
 - xsl:key, 376, 469
 - xsl:namespace-alias, 394, 469
 - xsl:output, 469
 - xsl:param, 469
 - xsl:preserve-space, 439, 469
 - xsl:strip-space, 465, 469
 - in xsl:stylesheet, 468–469
 - XSLT-defined, list of, 105–106
 - xsl:template, 469
 - xsl:variable, 469, 500
- declarations, text. See text declarations**
- declarative language, XSLT as, 8, 40, 979**
- decomposition, 731**
- deduplication, 544. See also distinct-values(); removal;
xsl:for-each-group**
- deep copy, 287, 292, 293, 296. See also xsl:copy-of**
 - recursive, 290
- deep-equal() function, 745–748. See also
saxon:deep-equal()**
 - base URI and, 747
 - comments and, 746
 - comparing sequences with, 745–746
 - «eq» operator and, 746
 - namespace nodes and, 746, 747
 - order of attributes and, 747
 - processing instructions and, 746
 - type annotations and, 747
 - whitespace text nodes and, 747
- #default, 242, 395, 485, 486**
- default attribute values, 65, 481**
 - in DTD, 481
- default collation, 377, 461, 467, 470, 561, 562**
 - comparing strings and, 588, 724, 728, 731
 - «lt», «le», «ge», «gt», and, 584
 - for xsl:sort, 461
- default-collation attribute, 99, 159, 467**
 - effect/usage, 470
 - on literal result elements, 468
 - [xsl:], 112, 470
 - xsl:for-each-group and, 470
 - xsl:key and, 470
 - xsl:sort and, 470
- default-collation() function, 748**
- default collection of documents, 726**
- default decimal-format, 299, 789**
 - examples, 791–792
- default (unnamed) mode, 238, 242, 485, 486**
- default namespace declarations, 480–482, 536, 1223**
 - as default attribute values in DTD, 481
 - definition/description, 1223
 - xpath-default-namespace and, 480–482, 615
- default namespaces, 58, 308, 534**
 - for elements, 481, 556, 615
 - constructor functions and, 560
 - for functions, 135, 482, 545, 556, 560, 655, 793

default namespaces (continued)

default namespaces (continued)

- and QNames without prefixes, 536, 766, 866
- static context, 556
- for types, 556, 615
 - constructor functions and, 560
- undeclaration, 58
- XPath, 556, 696. *See also* xpath-default-namespace NameTest and, 696

default priorities, 250, 401, 484, 486

- of patterns, 686–687, 691

default separator, 258, 497

default template rules. *See* built-in template rules

default values

- defined in schema, 313
- for elements and attributes, 313
- as empty sequence, 504
- of parameter, 244, 273, 427, 428–429, 487
- xsl:param with, 432–433
- as zero-length string, 503

defaulting, of axis name, 621–622

default-validation, 99, 297, 315, 467, 470–471

- literal result elements and, 470, 471
- xsl:copy and, 471
- xsl:copy-of and, 471
- xsl:element and, 471
- xsl:result-document and, 471

deferred evaluation, of variables, 388

definitions, type. *See* type definitions

Definitive XML Schema (Walmsley), 151

delimiter, 532

delimiting quotation marks, 528

- doubling of, 532

deprecation

- of disable-output-escaping, 280, 492, 493, 946–947
- of namespace axis, 61, 601

depth, of nodes, 274

depth-first traversal, 250

dereferenced URIs, 357, 373, 567, 755, 757

derived by restriction (types), 664

derived numeric types, 192, 193, 210–213

derived string types, 193, 213–215

derived types

- in basic XSLT processor, 192
- built-in, 664, 665
 - conversion rules, 665
- conversions between, 664–668
- converting non-string values to, 666–667
- converting xs:string to, 665–666
- numeric, 192, 193, 210–213
 - type hierarchy, 211
- string, 192, 193, 213–215
- user-defined, 664

«/descendant::», «//» compared with, 628

descendant axis (XPath), 609, 610, 1223

descendant-or-self axis (XPath), 609, 611, 1223

descending order, 459

design patterns (stylesheets)

- computational, 985–1000, 1114. *See also* knight's tour (case study)
- fill-in-the-blanks, 38, 973–976
- navigational, 976–980, 1114. *See also* family tree (case study)
- rule-based, 74, 246, 980–985, 1047, 1113. *See also* XML specification (case study)
- visitor, 250

development environment

- Kernow, Saxon and, 10, 13–14, 1195
- oXygen, 10, 12, 13, 1195
- Stylus Studio, 10, 12, 13, 16–17, 1195
- XMLSpy, 10, 12, 13, 15–16, 91, 479, 1195, 1215–1216

diacritics, 848

- and accents in NFC, 880, 884

diagnostic functions. *See* error(); trace()

diagnostics, 107

- trace() and, 896–897
- xsl:message and, 388, 897

dialect of XSL, WD-xsl as, 467

dictionaries, data, 369

difference between two dates, 580

difference operator. *See* «except» operator

diffspec.xml overlay stylesheet, 1003, 1041–1044

digit placemaker, optional, 789

Digit symbol, 530

- as regular expression, 530
- square brackets for, 530

digital signatures, 396

digits. *See also* groups of digits

- after decimal point, 790
- decimal, 531
- Indic Arabic, 302
- in regular expressions, 233, 921
- significant, 790
- Unicode, 408, 782
- in xsl:decimal-format, 302
- in xsl:number, 408–409

Dijkstra, Edsger, 985, 986

direction, of axis, 242, 323, 620

disable-output-escaping, 280, 909, 945–949

- backward-compatibility mode and, 1140
- CDATA sections and, 947
- character maps v., 948–949
- deprecation of, 280, 492, 493, 946–947, 1140
- Firefox and, 946
- grouping and, 947
- xsl:text and, 492, 493, 945
- xsl:value-of and, 496, 497, 945

disabling namespace inheritance, 312

disabling output escaping, 945–949

disambiguation of «/», 543

- distinct-values() function, 249, 333, 435, 459, 749–750**
 - duplicate values and, 749–750
 - «eq» operator and, 749
 - hyperlink output and, 435
 - index-of() and, 750
 - transitivity and, 587
 - undefined order of result, 640
 - XQuery and, 750
 - xsl:for-each-group v., 750
- «div» operator, 200, 538, 554, 571**
 - durations divided by durations, 580–581
 - durations divided by numbers, 579
 - exact division, 572
 - «mod» and, 572, 575
 - precedence, 572
 - precision of result, 575
- divide-and-conquer recursion, 993**
- division**
 - decimals and, 575
 - of durations, 577, 579, 580–581
 - by zero, 574, 575
- division operators, 572. See also «div»; «idiv»**
 - «/» v., 572
- doc-available() function, 750–753**
- DocBook stylesheet suite, 433**
- doc() function, 44, 55, 85, 323, 440, 477, 568, 750–753**
 - base URI and, 562, 751
 - base-uri() and, 750
 - document-uri() and, 764
 - document() v., 750
 - dynamic context and, 567, 568, 751
 - evaluation context and, 751
 - implementation, 752
 - relative URI and, 751
 - resolve-uri() and, 750, 751, 868–869
 - schema validation and, 752
 - unparsed-text() and unparsed-text-available() v., 905
 - validation invoked by
 - (AltovaXML) control over, 671
 - (Saxon) control over, 671
 - Web documents and, 751
 - XQuery and, 751
 - xsi:schemaLocation and, 558, 671
- DOCTYPE declaration, 67, 422, 446, 931**
 - public identifier in, 446
 - system identifier in, 422, 447, 933, 938
- doctype-public, 421, 422, 446, 448, 933**
 - HTML output method and, 938
 - public identifier and, 933, 938
 - XML output method and, 933
- doctype-system, 421, 422, 447, 448, 933**
 - HTML output method and, 938
 - XML output method and, 933
- document (''), 321, 760**
- document() function, 44, 55, 68, 84, 85, 138, 321, 323, 373, 440, 477, 510, 568, 754–763**
 - applied to atomic values, 760–761
 - applied to nodes, 757–758
 - base URIs and, 757, 760, 763
 - doc() v., 750
 - dynamic context and, 567, 568
 - «is» operator and, 756
 - modes and, 758
 - resolve-uri() and, 763
 - schema validation and, 757
 - stylesheet analysis with, 758–759
 - temporary trees and, 756
 - unparsed-text() and unparsed-text-available() v., 905
- document(s), 1223. See also transformation process, XSLT; XDM tree model**
 - built in memory, 752
 - collection of, 568, 726. *See also* collection() function
 - default, 726
 - definition/description, 1223
 - dynamic context and, 567, 568
 - input. *See* source documents
 - locating text within, 413–415
 - mappings from URIs to, 568
 - multiple result. *See* multiple result documents
 - multiple source. *See* multiple source documents
 - nodes. *See* document nodes
 - order. *See* document order
 - output. *See* result documents
 - parsing, 43, 752, 756–757
 - result. *See* result documents
 - secondary, 510, 692, 761
 - source. *See* source documents
 - statically known, 562–563
 - technical/legal, sequence numbers for, 418
 - temporary. *See* temporary documents
 - trivial, 514
 - well-formed. *See* well-formed documents
- document element (XML), 48, 1223**
- document nodes, 47, 49, 289, 454, 1223–1224**
 - binary «/» and, 617
 - definition/description, 1223–1224
 - document-node() and, 671
 - KindTests for, 700–701
 - process of forming content of, 304
 - root nodes other than, 624, 625, 688, 870
 - rooted expressions and, 623
 - string value, 51
 - trees with no, 624
 - trees with root other than, 624, 625, 688
 - value of parameter as, 306
 - in XDM data model, 304
- Document Object Model. See DOM**

document order

document order, 57–58, 228, 241, 323, 378, 1224

- «<<», «>>» and, 593, 594
- for attributes, 610
- definition/description, 228, 1224
- duplicates and, 378, 601, 602
 - binary «/» operator in, 602
 - node identity in, 602
 - path expressions and, 644
- «for» expression and, 642
- forward axes and, 608, 620
- no sorting into
 - comma operator and, 635
- of nodes, 57–58
 - on attribute axis, 620
 - on namespace axis, 620
- operators for testing nodes before/after in, 57, 593, 594
- path expressions and, 644
- reordering sequences in, 602
- reverse, 608, 620
 - reverse axes and, 608, 620, 1228
- sorting sequences into, 57, 241, 602
- specified order, comma operator and, 635

Document Style Semantics and Specification Language.

See **DSSSL**

‘document type,’ 671

document type declaration, 933, 938. *See also*

DOCTYPE declaration

Document Type Definitions. *See* DTDs

DocumentBuilder (JAXP class), 1168–1169

DocumentBuilderFactory (JAXP class), 1167–1168

document-formatting stylesheets, 1047. *See also*
rule-based stylesheets

document-level integrity checks, 179

document-level validation, 293, 303

document-node(), 698

- document nodes and, 671
- KindTest, 617, 700
- NodeTest, 162

DocumentTest construct, 616, 670, 697

document-uri() function, 764

- collection() and, 727, 764
- doc() and, 764
- URIs and, 764

dollar sign («\$»)

- as regex metacharacter, 233, 926. *See also*
metacharacters
- in replacement string, 863
- in variable references, 429, 528, 540

DOM (Document Object Model), 8, 11, 22, 963–965

- base URI and, 756, 876
- Document, 359
 - stylesheet parameters in form of, 430
- expressions in, 682
- extension functions and, 963–965

- JAXP support for, 1166–1169

- in MSXML, 1147–1152

- node manipulation and, 961

- result tree via, 420

- Saxon and, 1200, 1205

- tree, 44

- XDM tree v., 42, 963–965

Domain Name System, 755

domain-specific collating rules, 461

DOMLocator (JAXP class), 1171–1172

DOMSource (JAXP class), 1172–1173

dot-all mode (regex evaluation), 231

- «s» flag and, 917, 926

dotless I (Turkish), 827, 910

double. *See* xs:double

double markup, 942

double precision numbers, 198–201

- values for, 198–199

double quotes, 64

- string literals and, 532

DoubleLiteral symbol, 530

double-precision floating-point numbers, 198, 531, 576.

See also **xs:double**

doubling

- of curly brackets, 123, 915

- of delimiting quotes, 532

- of square brackets, 781

download example files, xxxiii–xxxiv

DSSSL (Document Style Semantics and Specification Language)

- Clark on, 28, 986

DTDs (Document Type Definitions), 24, 27, 471, 1224

- default attribute value in, 481

- default namespace declaration as, 481

- definition/description, 1224

- exclude-result-prefixes and, 471

- internal subset, 930

- character maps and, 942

- type ID and, 803

- unparsed entity declaration in. *See* unparsed entities

- validation with, 752, 755, 757, 805

- ID attributes and, 62

- namespace prefixes and, 857

- prefix-from-QName and, 857

- xs:NOTATION and, 209

duplicate(s). *See also* distinct-values();

groups/grouping; xsl:for-each-group

- attribute names, 268, 309

- attribute set names, 268

- declarations, in stylesheets, 97, 374

- distinct-values() and, 749–750

- document order and. *See* document order

- empty sequence with no, 607

- grouping keys, 328–329

- groups, removal of, 328

- IDs, 62
 - items, removal of, 333
 - namespace declarations, 119, 309, 473
 - nodes, 328
 - node identity and, 602, 629
 - removal of, 378, 544, 603
 - values
 - in group-by attribute, 328
 - removal of, 749–750
 - durations, 205–206. See also xs:dayTimeDuration; xs:duration; xs:yearMonthDuration**
 - add/subtract, 577, 579
 - arithmetic operators and, 571, 574, 577–581
 - average of, 718
 - components of. *See* days-from-duration(); hours-from-duration(); minutes-from-duration(); months-from-duration(); seconds-from-duration(); years-from-duration()
 - converted to number of months, days, or seconds, 581
 - divided by durations, 580–581
 - divided by numbers, 577, 579
 - equality of, 584
 - functions for (list), 711
 - multiplying, 577, 579
 - negative, 578, 580
 - normalized, 745
 - plus date/time, 578–579
 - ratio between two, 577
 - totals of, 889–890. *See also* sum()
 - dynamic construction, of QName, 535. See also QName(); xs:QName**
 - dynamic context (XPath), 84, 85, 346, 553, 563–568, 1224. See also static context**
 - collection() function and, 567, 568, 726
 - current-date() and, 566
 - current-dateTime() and, 566
 - current-time() and, 566
 - definition/description, 1224
 - doc() function and, 567, 568, 751
 - document() function and, 567, 568
 - documents/collections and, 567–568
 - focus. *See* focus
 - function implementations and, 566
 - implicit-timezone() and, 567
 - static context v., 554
 - variable values, 566
 - dynamic errors, 80, 649, 1224**
 - definition/description, 1224
 - in «some» and «every» expressions, 649
 - Dynamic HTML, 286. See also JavaScript**
 - dynamic loading of assemblies, in .NET, 962**
 - dynamic sort keys, 462–463**
 - dynamic typing, 186, 187, 221–224, 559**
 - «treat as» operator and, 679
 - dynamically scoped variables, 429**
 - dynamically typed languages, JavaScript/VBScript as, 959**
 - dyn:evaluate() function, 463**
- ## E
- early normalization, 848**
 - ECMAScript, 953**
 - EDI message, 42, 906**
 - effective boolean value, 283, 551, 555, 1224. See also boolean() function**
 - of «and» operands, 595
 - backward compatibility and, 722
 - boolean() and, 721
 - definition/description, 1224
 - empty sequence and, 778
 - «if» expression and, 555
 - not() and, 850
 - numeric/boolean predicates and, 638
 - of «or» operands, 595
 - predicates and, 619, 702
 - quantified expressions and, 646
 - rules for forming, 552, 722
 - «satisfies» expression and, 646
 - in use-when, 478
 - in xsl:if, 354
 - in xsl:when, 516
 - effective value (XSLT), 299, 378, 468, 587, 1224**
 - definition/description, 1224
 - effective version, 318**
 - Eiffel, 10, 136**
 - add_extension_function_library() (Gestalt), 962
 - evaluate_item(), 962
 - extension functions in, 957, 962
 - Gestalt and, 957
 - element-available() function, 128, 131, 140, 141, 319, 764–770**
 - extension elements and, 764–770
 - extension instructions and, 766
 - multiple result documents (example), 767–769
 - system-property() and, 892
 - use-when attribute and, 140–141, 478, 766
 - [xsl:]extension-element-prefixes and, 766
 - xsl:fallback and, 766
 - XSLT 1.0 processor and, 766
 - XSLT instructions and, 764–770
 - XSLT versions and, 766–769
 - element(), 698, 700, 713**
 - KindTest, 617, 700
 - matches any element node, 673
 - NodeTest, 162, 164
 - schema-defined types and, 617

element(s)

element(s), 227–519, 1224. See also specific elements

- annotation of, 312–315
- creation of, 307–311. *See also* `xsl:element`
- default namespace for. *See* default namespace
- definition/description, 1224
- document, 48
- empty
 - HTML output method and, 936
 - tags, 65
 - XHTML output method and, 939
- encapsulating properties of, 348
- grouping/ordering of, 229
- as IDs, 803
- KindTests for, 698–699
- literal result. *See* literal result elements
- matching, 672–677
- names. *See* element names
- namespace prefixes in content of, 392
- nillable, 183
- nilled, 183
- order of, 64
- outermost, 48
- parentless, 55, 87
 - patterns and, 683, 684, 688–689
- QName-valued, 393
- as root nodes, 688
- string value, 51, 742
- top-level. *See* top-level elements
- unrecognized, 318
- validation of, 176–179, 289, 312–315

element (N, T), 371

element content, 441

- schema types and, 742

element declarations (in schemas), 152, 156, 159, 164, 171, 173, 177, 314, 1224

- definition/description, 1224
- global. *See* global element declarations
- top-level. *See* global element declarations
- user-defined, 369

element names. See also name(); node-name()

- changing, 391
- decided at runtime, 308
- local name, 308
- namespace URI, 308
- in path expressions, 482
- in patterns, 482
- prefix, 308
- QNames as, 536
- in SequenceType syntax, 482
- in `xsl:preserve-space`, 440, 482
- in `xsl:strip-space`, 465, 482

element nodes, 1224. See also element(s)

ElementName construct, 672

ElementNameOrWildcard construct, 672

element-only content model, 53, 142, 156–157, 506, 878

ElementTest construct, 616, 671, 672, 697

elimination. See removal

embedded languages, 39

- XPath as. *See also* evaluation context
 - in Java, 553
 - in Perl, 553
 - in XForms, 553, 709
 - in XML Schema, 553
 - in XPointer, 553, 682
 - in XSLT, 22, 39–40, 527, 553

embedded stylesheets (XSLT), 94, 102–104, 466, 1224

- browser support, 102
- definition/description, 1224
- example, in Firefox, 102–104
- Internet Explorer and, 102
- in `<?xml-stylesheet?>` processing instruction, 469

empty content, 53, 742

empty elements (XML)

- HTML output method and, 936
- tags, 65
- XHTML output method and, 939

empty sequence («()»), 51, 53, 54, 57, 187, 459, 566, 623, 1224

- «/..» and, 623
- as argument, 713
- atomization of operands and, 573, 586
- constructor functions and, 655
- default value as, 504
- definition/description, 1224
- effective boolean value and, 778
- empty() and, 770
- «every» expression and, 647
- with no duplicates, 607
- null value in SQL and, 586
- one-or-more() and, 853
- parentheses for, 542
- result of axis step as, 607
- setting of version = “2.0” and, 1143–1144
- «some» expression and, 647
- string functions and, 710
- sum() and, 634, 889
- type labels and, 634
- zero-length strings and, 203

empty table cells, generation of, 344

empty template rule, 261

empty() function, 552, 648, 770–771

- empty sequence and, 770
- zero-length string and, 770

empty-sequence(), 653, 669

encapsulation

- of properties of elements, 348
- of relationships, 348

encoded characters. See character encoding

- encode-for-uri() function, 771–772**
 - escape-html-uri() and, 772
 - iri-to-uri () and, 772
 - percent encoding in URIs and, 771–772
 - unreserved characters and, 772
- encoding attribute, 933. *See also* character encoding**
 - character references and, 933
 - HTML output method and, 938
 - JAXP and, 941
 - text output method and, 940
 - of XML declaration, 933
 - XML output method and, 933
 - of xsl:output, 421–422, 941
 - of xsl:result-document, 447–448
- encoding information, external, 905**
- encoding, percent. *See* escaping special characters**
- end tags, 49, 1229**
- end-of-line. *See* line ending; newline**
- ends-with() function, 204, 773–774**
 - matches() and, 774
 - replace() and, 774
 - substring matching with collation, 731–732
 - Unicode codepoint collation and, 773
- entities (XML), 440, 1224**
 - definition/description, 1224
 - external parsed. *See* external parsed entities
 - unparsed. *See* unparsed entities
 - XML document. *See* well-formed documents
- ENTITIES attribute, 903**
- ENTITY attribute, 903**
- entity boundaries, 64**
- entity references (XML), 143, 880, 903, 1224. *See also* character references**
 - character maps and, 942
 - definition/description, 1224
 - external, 64, 720
 - URI of, 54
 - HTML, 281
 - predefined, 444
 - special characters converted to, 493
 - XML output method and, 930
- envelope/payload, 909. *See also* unparsed-text()**
- environment variables, 427**
 - system-property() and, 892
- “eq” operator, 329, 377, 378, 458, 538, 571, 581**
 - collations and, 459, 584
 - deep-equal() and, 746
 - “ne” v., 584
 - operand types for, 583–584
 - transitivity and, 587
 - ValueComp, 582
- equal sort keys, 324**
- equality. *See also* “eq” operator**
 - of durations, 584
 - transitive, 587
- equations, numbering of, 407, 415**
- era component, 782**
- error codes, 775**
 - (list), 1123–1138
 - user-defined, 775
- error(s)**
 - “/” root expression and, 624
 - application detects, 774
 - attribute/namespace preceded by node, 309
 - circularity, 246, 267, 282, 378, 485, 501
 - conditional expressions and, 552
 - “context item is not a node,” 232, 624, 625
 - division by zero, 574, 575
 - dynamic, 80, 649, 1224
 - dynamic typing. *See* dynamic typing
 - in “every” expression, 649–650
 - match attribute and, 485, 688
 - recoverable, 1140
 - reporting, xsl:message and, 388
 - rooted path expressions and, 625
 - rounding, 575, 576
 - runtime, atomization and, 499
 - in “some” expression, 649–650
 - static, 80, 1229
 - static typing. *See* static typing
 - try/catch mechanism and, 80, 752, 774
 - type. *See* type errors
 - types and detection of, 53, 186, 221
 - vendor-defined declarations and, 107
 - xsl:message and, 388, 775
 - XSLT, 80
 - dynamic, 80
 - static, 80
- error() function, 774–775**
 - xsl:message and, 775
- error-handling scheme, 775**
 - backward-compatibility mode and, 1140–1141
- ErrorListener (JAXP interface), 1173**
- escaped spaces, “x” flag and, 927**
- escape-html-uri() function, 775–777**
 - encode-for-uri() and, 772
 - HTML/XHTML serialization and, 776, 938
 - percent encoding in URIs and, 775–777
 - URI escaping conventions and, 775, 776, 938
- escapes, character class. *See* character class escapes**
- escape-uri-attributes, 421, 422, 447, 448. *See also* encode-for-uri()**
 - HTML output method and, 938
 - JavaScript and, 938
 - percent encoding in, 422
 - serialization property, 776, 936
 - XHTML output method and, 940
 - in xsl:output, 422
 - in xsl:result-document, 447

escaping special characters

escaping special characters. See also backslash; character references; disable-output-escaping; entity references

in attribute value templates, 123
non-ASCII, 776
in processing instructions, 445
in regex, 232, 915, 920–921, 927
in replacement string, 223
in string literals, 532
in URIs. *See* percent encoding
in XML output, 493, 497
in XPath, 527

evaluate() extension function

dyn:, 463
saxon:, 135, 541, 546, 1210–1214

evaluate_item() (Gestalt), 962

evaluating templates, 486–487. See also template(s)
evaluation context (XPath), 553–568, 639. See also dynamic context; static context

doc() and, 751

even-numbered nodes, 241

«every» expression, 39, 540, 552, 646–650

«=» operator and, 648
boolean() and, 722
context item and, 646
empty sequence and, 647
errors in, 649–650
dynamic, 649
focus within, 565
range variables in, 646
semantics of, 647–648

exact copy, 296

exactly-one() function, 777–778

static type checking and, 777

example files, downloads, xxxiii–xxxiv

examples

«/», 627–628
aggregating list of numbers, 993–995
alphabet, 122–123
ancestors of node, 325
attribute name at runtime, 264–266
attribute set
for generated HTML table elements, 269
for numbering, 270–271
axis steps, 608–609
binary «/», 604–605
book list. *See* book list
boolean operators, 596
calling extension functions within loop, 965–967
cities grouped by country, 817–818
comma, as decimal point, 301–302
comma operator, 635–636
comma-separated-values file, 907–908
comments
commented-out JavaScript, 286–287, 943

date and time in, 286

with markup, 287

XSLT processor identification in, 286

context-sensitive values, 507–510

converting attributes to child elements, 315–316

current(), 736–738

cycles in graphs, 251–253, 352–353

default decimal-format, 791–792

embedded stylesheets, in Firefox, 102–104

envelope/payload, 909

«every» expression, 648

extension functions

with uncontrolled side effects, 967–970

VBScript in MSXML stylesheet, 958–959

extension instruction, 139–141

family tree, 805

family tree (case study), 1049–1098

fill in the blanks stylesheet, 974–976

filter expressions, 640

flat XHTML documents, 340–342

formatting list of names, 356

forward compatibility, 319

function call, 550–551

general comparison, 592

group-adjacent, 335–340

grouping consecutive elements by name, 336–338

Hello world, 10–19

HTML [OPTION] element with SELECTED attribute, 263–264

HTML table, generation of, 116

import precedence

of global variables, 364–365

of template rules, 365–368

Indic Arabic, 302

«instance of» operator, 678

JAXP transformations, 1187–1193

key() pattern, 706–707

KindTests, 617

knight's tour (case study), 221–222, 224, 1099–1114

locating text within document, 413–415

look-up table in stylesheet, 761–763

memory, 137

message, 388–389

localized, 389–390

modes, 247–251, 490–491

multilevel grouping, 333–335

multiphase transformation, 511–513

multiple result documents, 451–452, 767–769

music transformation, 5–7

namespace-uri-for-prefix(), 840–841

NameTests, 615–616

NaN and infinity, 302

navigational stylesheet, 977–980

Naxos Breakfast Supplies, 380

- node tests, 613–614
- no-namespace, 372
- nonunique keys, 381–383
- numeric arithmetic, 576–577
- path expressions, 600–601
- PatternSteps, 703–704
- period, as thousands separator, 301
- picture argument, 787–788
- poem
 - attribute set for numbering in, 270–271
 - formatting of, 35–38
 - import precedence of template rules in, 365–368
 - multiple result documents, 451–452, 767–769
 - numbering lines of, 417–418
 - validation of output document, 171–174
 - validation of source document, 167–170
- push processing, 74–76
 - controlling sequence of processing, 77
 - selecting nodes explicitly, 77–78
- recursion
 - process sequence of nodes in, 275–278
 - process sequence of strings in, 278–280
- rendering elements, 239
- repeated output (with `xsl:copy-of`), 294–296
- repeating groups, of adjacent elements, 338
- `resolveQName()`, 865
- rooted path expressions, 626
- sales report, 977–980
- schema for XSLT 2.0 stylesheets, 674–677
- Shakespeare plays, 336–340
- simplified stylesheet, 125–127
- single-level grouping, 331–333
- soccer, 294–296
 - multiphase transformation, 511–513
- «some» expression, 648
- sorting on result of calculation, 463
- square root extension function, 959–961
- states (U.S.), 284
- static typing, 221–224
- stripping whitespace-only text nodes, 442
- stylesheet
 - `xsl:import` in, 95–98
 - `xsl:include` in, 91–95
- stylesheet analysis with `document()`, 758–759
- table (HTML), 116
 - arranging data in, 343–344
 - attribute set for, 269
 - template rules, 488–490
- temporary document, 86
- temporary trees, 320–321
- «to» operator, 637
- tunnel parameters, 433–437
- Unicode normalization, 847–848
- union type, 372
- value comparison, 587
 - vendor portability, 319–320
- Venn operators, 630
- well-balanced XML fragment, 48
- whitespace, 493–494
- word count, 19–21
- XML specification (case study), 1001–1048
- XML tree, 46–48
 - `<?xml-stylesheet?>`, 444–445
 - JAXP transformation, 1189–1190
- `xsl:analyze-string`
 - multiple-match, 236–237
 - single-match, 235–236
- `xsl:namespace-alias`, 398–399
- `xx:node-set()` extensions, 794–796
- examples, running, 12–17**
 - AltovaXML, 15, 169–170, 277, 347–348, 352–353, 382–383, 994–995, 1217–1218
 - browser, 13
 - Firefox, 102–104
 - Gestalt, 382–383
 - Java method, 796
 - Kernow, 13–14
 - Saxon, 139, 382–383, 398–399, 433–437, 476–477, 799, 959–961, 1087–1090, 1092–1095
 - Saxon on Java from command line, 14–15, 103
 - Saxon on .NET, 15
 - Saxon version 6, 767–769
 - `saxon:evaluate()`, 1212–1214
 - Saxon-SA, 169, 172–173, 769, 995, 1200–1201
 - Stylus Studio, 16–17
 - XMLSpy, 15–16, 1215–1216
- Excel spreadsheets, up-conversion to XML from, 343**
- «except» operator, 57, 261, 353, 538, 628–631**
 - «is» operator and, 593
 - path expressions and, 599, 628. *See also* path expressions
 - XPath 1.0 and, 631
- excess whitespace. *See* whitespace-only text nodes**
- exclamation point «!» (in jar URI scheme), 868**
- exclude-result-prefixes, 104, 358, 374, 467, 471–475**
 - DTDs and, 471
 - effect/usage, 471–475
 - [`xsl:`], 112, 118, 119, 315, 358, 392
 - redundant namespace nodes and, 394
- existential comparison, 279, 590–592. *See also* general comparisons**
- existential quantifiers (logical), 646. *See also* «some» expression**
- exists() function, 552, 622, 648, 778–779. *See also* boolean(); empty(); not()**
- expanded QName, 50–51, 201, 1224–1225. *See also* xs:QName**
 - definition/description, 1224–1225
 - local-name-from-QName() and, 826

expanded QNames (continued)

expanded QNames (continued)

- namespace URIs and, 50–51, 840–841
- original prefix in, 58
- prefixes from, 857–858
- QName() and, 858–859

expected type, of parameter, 431

explicit priorities, 401

exponential notation (scientific notation), 190, 530, 531

- for floating-point numbers, 198, 530
- format-number() and, 660
- number() and, 851

Expr construct, 525

expression language, XPath as, 185, 553

expressions (XPath), 8, 82–84, 228, 524–529, 1225.

See also specific expressions

- arbitrary ordering of, 901–902. *See also* unordered()
- arithmetic, 454
- conditional. *See* conditional expressions
- constructed at runtime, 541
- context. *See* context
- conventional programming, 39
- curly brackets and, 117, 122, 123
- definition/description, 228, 1225
- in DOM, 682
- extension functions and, 526
- filter. *See* filter expressions
- names in. *See* name(s)
- ordering results of, 901. *See also* unordered()
- parenthesized, 542–543
- path. *See* path expressions
- patterns v., 681–682
- primary. *See* primary expressions
- grammatical construction of, 396
- quantified, 646
- regular. *See* regular expressions
- root. *See* root expressions
- rooted path. *See* rooted path expressions
- in Saxon, 1201–1202
- SELECT statement v., 568, 632
- sequence. *See* sequence expressions
- sequence type descriptors in, 165
- side effects and evaluation of, 526
- simple mapping. *See* simple mapping expressions
- step, 57
- stylesheet parameter in form of, 430
- syntax. *See* syntax
- test, 283
- type. *See* type expressions
- type-safe, 777
- as Unicode characters, 527
- unprefixed names in, 481–482
 - xpath-default-namespace and, 481–482
- for values of double-precision numbers, 199

- in XML Schema 1.1, 682

- within XPointer, 553, 682

ExprSingle construct, 525, 545

- «,» and, 634

EXSLT library, 135, 346, 463, 546, 631, 955

extended BNF, 522, 689

extensibility, XSLT, 134–141, 953–971. *See also*

extension functions; extension instructions

- design principles, 134
- extension functions, 953–971
- in Gestalt, 962
- in Java processors, 959–961
- mechanisms for, 134
- in MSXML, 957
- portable extension functions, 970–971. *See also* portable stylesheets
- in Saxon, 959, 962, 1205–1208
- in Saxon on .NET, 962
- in System.Xml.Xsl, 961

Extensible Stylesheet Language. *See* XSL

Extensible Stylesheet Language: Transformations. *See* XSLT

extension(s)

- portable, 970–971
- proprietary, 319
- Saxon, 1208–1210
- user-defined, 318
- vendor-specific. *See* vendor-specific extensions

extension attributes, 954, 1225

- Saxon, 1209
- serialization and, 1225

extension declarations, 106–107, 954

extension elements, 317, 467, 475, 1225. *See also*

xsl:fallback

- element-available() and, 764–770
- ### extension functions, 106, 111, 134–138, 709, 953–971, 954, 1225
- alternatives to, 138
 - binding, 957–962. *See also* binding
 - in C#, 957
 - calling, 956–957
 - calling, within loop, 965–967
 - client side script and, 957
 - definition/description, 1225
 - DOM and, 963–965
 - in Eiffel, 957, 962
 - evaluate() extension function. *See* evaluate()

- extension function
- expressions and, 526
- and external data to stylesheets, 956
- function-available() and, 792, 793
- in-scope functions as, 560
- in Java/JavaScript, 134–135, 137, 956
 - XSLT 1.1 and, 135
- in JScript, 134, 709, 957

- languages for writing, 957
 - linked to stylesheets
 - Java and, 136–137
 - namespaces for, 471
 - in .NET languages, 957
 - node manipulation and, 961
 - performance and, 956
 - in Python, 957
 - random number generator and, 955
 - reasons for using, 955–956
 - Saxon, 959, 962, 1209–1210
 - saxon:evaluate(), 135, 541, 546, 1210–1214
 - side effects and, 138, 244, 549, 595, 955, 956
 - example, 967–970
 - square root calculation (example), 959–961
 - stylesheet parameters v., 430
 - trigonometric calculations and, 531, 955
 - user-written. *See* user-written extension functions
 - in VBScript, 134, 957
 - and VBScript in MSXML stylesheet (example), 958–959
 - wrong reasons for using, 956
 - extension instructions, 106, 111–112, 138–141, 228, 954, 1225**
 - AltovaXML and, 476
 - definition/description, 138, 228, 1225
 - element-available() and, 766
 - example, 139–141
 - extension-element-prefixes and, 475
 - implementation-defined, 954
 - literal result element v., 766
 - portable stylesheets and, 138–141
 - Saxon, 139, 140, 475–476, 1209
 - Xalan and, 140
 - xsl:fallback and, 769
 - XSLT, 138–141
 - extension namespace, 138**
 - instructions in, 317
 - extension types, 954**
 - extension-element-prefixes, 98, 107, 138, 358, 374, 467, 475–477**
 - debugging and, 477
 - effect/usage, 475–477
 - extension instructions and, 475
 - Saxon and, 476–477
 - [xsl:], 111, 113, 118, 138, 317
 - element-available() and, 766
 - external data to stylesheet, extension functions and, 956**
 - external encoding information, 905**
 - external entity references, 64, 720**
 - URI of, 54
 - external file, in form of string, 904. *See also* unparsed-text()**
 - external functions. *See* extension functions**
 - external objects, 961**
 - binding extension functions with, 961–962
 - external parsed entities, 48, 87, 929–930**
 - rules for, 930
 - well-formed XML documents v., 930–931
 - XML output method and, 929–932
- ## F
- facet(s), 1225**
 - converting non-string values to derived types and, 666
 - converting xs:string to derived types and, 665, 666
 - definition/description, 1225
 - pattern, 236, 665
 - canonical form and, 667
 - converting non-string values to derived types and, 667
 - regular expressions and, 667, 829
 - whiteSpace, 143, 214, 663, 665, 846
 - failed conversions, 654**
 - failure, validation, 166, 259, 1230**
 - fallback processing, 27, 80, 231. *See also* xsl:fallback**
 - forward-compatibility mode and, 448
 - xsl:analyze-string and, 231, 317
 - false value**
 - booleans and, 189, 191, 194, 195
 - false() function, 779. *See also* true()**
 - boolean constant and, 779
 - family tree (case study), 1049–1098**
 - creating data file, 1058–1072
 - creating schema for GEDCOM 6.0, 1053–1054
 - displaying data of, 1072–1098
 - GEDCOM 6.0 schema, 1054–1058
 - GEDCOM data model, 1050–1054
 - generating HTML
 - with ASP.NET pages, 1092–1095
 - in browser, 1095–1098
 - from servlet, 1087–1092
 - model for, 1050
 - publishing static HTML, 1086–1087
 - stylesheet for, 1073–1085
 - family tree example, 805**
 - figures, numbering sequences for, 415**
 - fill-in-the-blanks stylesheets (design pattern), 38, 973–976**
 - filter expressions, 540, 638–640**
 - axis steps v., 542
 - changing focus with, 565
 - last() and, 822
 - predicates in, 552, 637–638
 - predicates in axis steps v., 639–640
 - FilterExpr construct, 539, 638**
 - filtering, axis steps and, 606**
 - filtering items in sequence**
 - predicates and, 542

final result trees

final result trees, 74, 448, 1225. See also result trees

- definition/description, 1225
- temporary tree v., 1225
- URIs for identifying, 449

finding. See locating

finite nonzero values, 198

Firefox, 42, 68, 100

- disable-output-escaping and, 946
- embedded stylesheets example, 102–104
- HTML boilerplate generation and, 910
- <?xml-stylesheet?> processing instruction, 101

first character of string, 884–885

'first item' rule, 132, 133, 480, 498, 1142–1143

first or last item in list (position()), 856

first-class objects, 251

first-order operators, 537–538

fixup, namespace. See namespace fixup

flags, 230, 231, 829, 862, 894, 925–927. See also «i»

- flag; «m» flag; «s» flag; «x» flag
- nilled, 477
- whitespace and, 231, 926

flat XHTML documents example, 340–342

float. See xs:float

floating-point arithmetic, 576

- IEEE 754 standard, 198, 576

floating-point numbers, 189

- as atomic values, 189, 198
- double-precision, 198, 531, 576. See also xs:double
- exponential notation for, 198, 530
- single-precision, 208. See also xs:float

floor() function, 200, 779–780

- «idiv» and, 780
- rounding and, 780
- xs:integer() constructor function v., 780

flow-of-control instructions, 979

FLWOR expression (XPath 1.0), 602, 647–648

[fn:] prefix, 545, 710. See also specific unprefixd

function names

- boolean() function and, 552

focus, 85, 563–566, 821, 854–855. See also context item; context position; context size

- changing, 565
- within «every» expression, 565
- within «for» expression, 565, 855
- host languages and, 564, 565
- parts of, 821, 854
- for predicate evaluation, 620
- within «some» expression, 565

fold function, 251

following axis, 609, 611, 1225

following siblings, 116, 499, 501

following-sibling axis, 83, 609, 611, 1225

footnotes, numbering of, 407, 410, 415, 417

«for» expression, 21, 39, 540, 640–644

- context item in, 642–643

- context node and, 642–643

- context position in, 644

- context size in, 644

- document order and, 642

- focus within, 565, 855

- mapping sequences with, 641–642

- multiple input sequences and, 643–644

- path expressions v., 644

- range variable in, 644

- sequences and, 640

for loop, 274

ForExpr construct, 525, 640

- keywords and, 525

format attribute

- in xs:number, 407

- in xs:result-document, 446, 448

format string, 407–408

- xs:number and, 405, 407–408

format-date() function, 93, 195, 235, 781–788

- day of the week and, 581

- year component, 782

format-dateTime() function, 781–788

format-number() function, 298, 531, 576, 788–792.

See also xs:decimal-format; xs:number

- changes in specification for, 1141

- exponential notation and, 660

- round-half-to-even() and, 873

- xs:decimal-format and, 789

- xs:number v., 403, 406, 791

format-time() function, 781–788

formatting

- date and time. See format-date(); format-dateTime(); format-time()

- format-number() and, 403

- list of names, 356

- numbers, 403, 408–410, 788–791. See also

- xs:number

- numeric output, 531. See also format-number();

- xs:number

- tables. See XML specification (case study)

- xs:date, 781

- xs:dateTime, 781

- xs:decimal. See format-number()

- xs:double. See format-number()

- xs:float. See format-number()

- xs:decimal-format and, 403

- zeros, 409

formatting tokens, 407, 782

- «A» and, 782

forward axes, 607, 620, 1225

- document order and, 608, 620

forward reference, to local variable, 502

forward-compatibility mode, 110, 113, 125, 130–131,

319, 321, 479

- example, 319

- forward-compatibility mode (continued)**
 - fallback processing and, 448
 - function-available() and, 794
 - version attribute, 479
 - XSLT 1.0 rules for, 479
 - XSLT 2.0 rules for, 479
 - in XSLT 1.0, 130–131
 - xt and, 320
- forward-compatibility problems, HTML, 766**
- 4XSLT processor, 136, 957**
 - Python and, 957
- Fourthought, 66**
- fractional part of number, 790**
 - «#» and, 790
- fractional seconds component, 782, 783**
- fragments**
 - identifier, 756, 868, 905
 - result tree, 87, 320, 454. *See also* document nodes
 - well-balanced XML, 48, 1230–1231
- free memory example, 137**
- French quotation marks, 522**
- Frijters, Jerome, 1195**
- from attribute (xsl:number), 412, 417**
- fully-normalized, 849**
- funcproto.xsl stylesheet module, 1046–1047**
- function(s), 1225. *See also specific functions***
 - aggregate, 200, 332, 350
 - arguments. *See* arguments
 - arity of, 345, 346, 361, 546, 559, 712, 792, 1221
 - availability of, 792. *See also* function-available()
 - for calling. *See* calling; function calls
 - by category (list), 710–712
 - constructor. *See* constructor functions
 - conversion rules. *See* function conversion rules;
 - standard conversion rules
 - core library. *See* core function library
 - date/time
 - (list), 711
 - default namespace for. *See* default namespace
 - defined by host language, 546
 - definition/description, 1225
 - diagnostic. *See* error(); trace()
 - duration, list of, 711
 - extension (external). *See* extension functions
 - higher-order, 250–251, 272
 - implementations. *See* function implementations
 - in-scope. *See* in-scope functions
 - libraries. *See* function libraries
 - local parameters to, 425, 427
 - for locating nodes
 - (list), 711
 - mathematical, 135, 200
 - mathematical concept of, 987
 - names. *See* function names
 - namespace. *See* function namespace, standard
 - nodes created by, 550
 - with distinct identity, 988
 - not in namespaces, 560
 - numeric, 32
 - list of, 710
 - overloading of, 546
 - parameters. *See* function parameters
 - pure, 34, 566, 987
 - on QNames
 - (list), 711
 - recursive, 347, 350–351, 862, 992
 - result of, 346
 - returning context information
 - (list), 711
 - returning information on XSLT environment, 546
 - (list), 712
 - returning properties of nodes
 - (list), 711
 - with same name, 546, 559
 - in same namespace, 349
 - from Saxon, 135
 - on sequences, 72
 - (list), 711
 - with side effects, 549–550
 - signature. *See* function signature
 - square root, 346, 347
 - example, 959–961
 - stable, 726
 - standard. *See* core function library
 - static type asserted by. *See* exactly-one();
 - one-or-more(); zero-or-one()
 - string. *See* string functions
 - strings returned from, 497–498
 - stylesheet. *See* stylesheet functions
 - system-defined functions, 793–794
 - tail-recursive, 351, 353, 862, 966
 - templates as, 988
 - uniquely identified, 546, 559, 712
 - on URIs
 - (list), 711
 - user-defined. *See* user-defined functions
 - vendor-supplied. *See* vendor-supplied functions
 - written in XPath, 546
 - XForms-specific, 544, 553, 709
 - XSLT, 546
 - (list), 712
- function body**
 - «.» and, 544
- function calling conventions, 959**
- function calls, 82, 220–221, 345, 427, 544–551**
 - «.» operator and, 525, 545
 - arguments in, 545
 - converting arguments in, 547–549
 - core function library and, 546
 - example, 550–551

function calls (continued)

function calls (continued)

- identifying functions in, 545–546
- IdKeyPattern and, 704
- function conversion rules, 505, 548. See also standard conversion rules**
 - backward-compatibility mode and, 549
 - casting rules and, 548
- function implementations, 566**
- function libraries, 135. See also core function library**
 - constructor functions in, 559, 709. *See also* constructor functions
 - EXSLT, 135, 346, 463, 546, 631, 955
 - FXSL, 251, 347, 546, 955
 - mathematical, 135, 200, 346
 - third-party, 546, 955
 - Walmsley's (functx), 546, 955
- function names, 135, 345. See also QName**
 - default namespace URI for, 482, 545, 710
 - node tests and, 528
 - as QName, 545
 - unique function identity and, 546, 559, 712
 - unprefixed, 482, 710. *See also specific unprefixed function names*
 - unprefixed names not available as, 546
- function namespace, standard, 135, 482, 545, 556, 560, 655, 793**
 - default function namespace as, 482, 655, 793
- function parameters, 82, 346, 425, 426**
 - function signature and, 559
- function signature, 220, 547, 559**
 - core function library and, 712–713
- functional programming languages, 250, 350, 429, 985–989**
 - apply operator in, 604
 - dynamically scoped variables in, 429
 - knights tour (case study) and, 1113
 - Lisp, 429
 - map operator in, 604
 - Scheme, 28
 - side effects in, 449
- function-available() function, 128, 130, 131, 138, 478, 792–796. See also element-available()**
 - backward compatibility mode and, 792
 - constructor functions and, 793
 - extension functions and, 792, 793
 - forward-compatibility mode and, 794
 - Java method and, 796
 - portable stylesheets and, 970
 - static context and, 793
 - stylesheet functions and, 792, 793
 - system-defined functions and, 793–794
 - system-property() and, 892
 - use-when and, 478
 - vendor/third-party extensions and, 794
 - xsl:function and, 793

- [xsl:]use-when and, 792
- xx:node-set() extensions and, 794–796

FunctionCall construct, 544

functx, 546, 955

FXSL, 251, 347, 546, 955, 956

G

gDay. See xs:gDay

«ge» operator, 538, 571, 581

- default collation and, 584
- NaN and, 586
- ordered data types and, 584–586
- ValueComp, 582

genealogy, 1049–1050

- family tree case study, 1049–1098
- GEDCOM. *See* family tree (case study)

general comparisons, 588–592. See also «[»; «]»; «!|=» operator; «=» operator; «[=» operator; «]=» operator

- backward-compatibility mode and, 588, 589
- changes in XPath 2.0, 588–589
- examples, 592
- existential comparison in, 590–592
- NaN and, 589
- rules for, 589–590
- untyped atomic values and, 588
- value comparison operators v., 582, 588

GeneralComp construct, 582

general-purpose stylesheet module, 97

generate-id() function, 293, 450, 594, 797–800. See also id(); «is» operator; key()

- compare nodes by identity and, 797
- hyperlinks and, 797
- id() and, 800
- «is» operator and, 797, 817

geographical coordinate systems, Open GIS schema for, 557

German language, 409

- «^B» in, 827

Germany, collations in, 459

Gestalt processor, 10, 136, 340, 359

- add_extension_function_library(), 962
- binding extension functions in, 962
- collection() and, 727
- create_iterator(), 962
- Eiffel and, 957
 - extension functions in, 957
- evaluate_item(), 962
- extensibility in, 962
- running example, 382–383
- XM_XPATH_FUNCTION_LIBRARY, 962
- XM_XPATH_SYSTEM_FUNCTION, 962

gHorribleKludge types, 207

- global attribute declarations, 179, 259, 371, 557**
 - KindTests and, 699
 - schema for XSLT 2.0 stylesheets and, 676
 - sequence type descriptors and, 673–674
 - type annotations and, 673, 699
- global element declarations, 371**
 - schema-element and, 674, 699
 - sequence type descriptors and, 673–674
 - substitution group and, 674
 - type annotations and, 673, 699
- global parameters, 68, 82**
 - match attribute and, 485
 - xsl:param and, 425, 427
- global tree-valued variable, 294**
- global type definition, 54, 315, 371**
 - named, 313
 - simple, 259, 313
- global variables, 81, 108, 321, 362, 438, 506, 512, 1225**
 - context item and, 735
 - convenience, 507
 - definition/description, 1225
 - import precedence of, 364–365, 502
 - match attribute and, 378, 485
 - principal source document and, 506, 626
 - scope of, 501
 - tunnel parameters *v.*, 431
 - use attribute and, 378
 - xsl:apply-templates in, 245
 - xsl:perform-sort and, 438
 - xsl:variable and, 501
- glossary/terms, 1221–1231**
- gMonth. See xs:gMonth**
- gMonthDay. See xs:gMonthDay**
- GMT (Greenwich Mean Time), 717**
- Goldstein, Robert F., 27**
- goto statements, 986**
- grammar, XPath, 521**
- graphs**
 - cycles in, 251–253, 352–353
 - navigating, 275
- greedy quantifiers (regex), 232, 863, 894, 917**
- Greek language**
 - « α , β , γ , δ » for numbering sequence, 409
 - character block, 922
 - polymorphic* (many shapes) from, 186
 - « Σ » (sigma), 827
 - tunnel parameters and, 429
 - zero sign, 460
- Greenwich Mean Time. See GMT**
- Gregor XSLT processor, 10**
- Gregorian calendar, 784, 785, 786**
 - Julian calendar and, 786
- Gregorian dates, 196**
- group theory, 42**
- groups (regex). See captured groups; subexpressions**
- groups (substitution). See substitution groups**
- groups/grouping, 327–331. See also xsl:for-each-group**
 - of consecutive elements by name, 336–338
 - current, 346, 739. *See also* current-group()
 - disable-output-escaping and, 947
 - initial item of each, 328, 330
 - multilevel, 333–335
 - multiple
 - items belonging to, 335
 - nested, 334
 - of nodes, key() and, 813
 - order of first appearance, 328
 - with patterns, 740
 - population and, 327, 330, 739
 - processing order of, 328
 - repeating
 - of adjacent elements, 338–340
 - single-level, 331–333
 - sorting, 330–331
 - grouping keys and, 330–331
 - sorting items in each, 331
- groups of digits, 409**
 - grouping-separator for, 409, 790
 - grouping-size for, 409
 - in xsl:decimal-format, 300, 790
 - in xsl:number, 409
- group-adjacent attribute, 327, 328, 329, 740**
 - examples, 335–340
 - group-by *v.*, 329, 338
 - group-starting-with *v.*, 340
- group-by attribute, 327, 328–329, 740**
 - duplicate values, 328
 - group-adjacent *v.*, 329, 338
 - single-level grouping (example), 331–333
- group-ending-with attribute, 3, 327, 330**
 - arranging data in tables with, 343–344
 - pattern, 330
 - up-conversion to XML from Excel spreadsheets with, 343
 - usage, 342–344
- grouping keys, 327, 328, 384–385, 816–817**
 - boolean, 338
 - comparison of, based on type, 329
 - composite, 334
 - current, 346, 740
 - current-grouping-key() and, 740–741
 - duplicate, 328–329
 - example, 817–818
 - sorting groups and, 330–331
 - xsl:for-each-group and, 384, 816
- grouping-separator, 409, 791**
 - picture string, 300
 - thousands and, 791
 - xsl:decimal-format, 300

grouping-separator (continued)

grouping-separator (continued)

- in `xsl:decimal-format`, 300, 790
- in `xsl:number`, 409

grouping-size, 409

group-starting-with attribute, 327, 329–330

- flat XHTML documents example, 340–342
- group-adjacent v., 340
- pattern, 329–330, 340

«gt» operator, 538, 571, 581

- default collation and, 584
- NaN and, 586
- ordered data types and, 584–586
- ValueComp, 582

guillemets, 522

gYear. See `xs:gYear`

gYearMonth. See `xs:gYearMonth`

H

half-to-even rule, 873. See also round-half-to-even()

hash joins, 643

hash table, `xsl:key` and, 379

head-tail recursion, 275, 993

Hebrew language, 410

Hebrew numbering, 410, 783

Hello world example, 10–19

- explanation, 17–19
- running from XMLSpy, 15–16, 1215–1216

hexadecimals, 725

hexBinary. See `xs:hexBinary`

hierarchies. See also class hierarchy; type hierarchy

- navigating, 275
- node position in, 407
- parallel
 - interleaved markup and, 942
- of sequence numbers, 418
- in XHTML, 340

higher-order functions, 250–251, 272

higher-order operators, 538–539, 603, 604

- binary «/» as, 603, 604

holiday resort example, 798–799

host languages, 190, 527, 528, 553

- C#, 543
- escape conventions, 527
- focus and, 564, 565
- functions defined by, 546
- Java, 528, 543

hotel list example, 798–799

hour component

- (12 hours), 782
- (24 hours), 782
- `xs:dateTime`, 800
- `xs:duration`, 801
- `xs:time`, 800

hours-from-dateTime() function, 800–801

hours-from-duration() function, 801

- hours component (`xs:duration`) and, 801

hours-from-time() function, 717, 800–801

href attribute

- in HTML, generating, 411, 412
- of `<?xml-stylesheet?>`, 100
 - Firefox and, 101
 - Internet Explorer and, 101
- of `xsl:import`, 357
- of `xsl:include`, 373
- of `xsl:result-document`, 446

HTML. See also XHTML

- 4.0 specification, 775, 776
- attributes whose value is URI, 936
- Dynamic, 286
- empty elements, 936
- entity reference, 281
- evolution of, 26–27
- forward-compatibility problems, 766
- generating (family tree case study)
 - with ASP.NET pages, 1092–1095
 - in browser, 1095–1098
 - from servlet, 1087–1092
- generating href attribute in, 411, 412
- meta element (`charset` attribute), 938, 939, 940, 941
- [OPTION] element with `SELECTED` attribute, 263–264
- output method. See HTML output method
- processing instructions in, 444
- result tree, 423
- selection box, 263
- serialization. See HTML output method; serialization
- [table] elements, attribute sets for, 269
- table example, 116
- valid, HTML output method and, 936, 937
- version, version attribute and, 939

HTML boilerplate generation, 909–910

HTML element, 940

HTML output method, 42, 43–44, 423, 444, 936–939.

See also serialization

- empty elements and, 936
- `escape-html-uri()` and, 775, 938
- `escape-uri-attributes` and, 938
- `include-content-type` and, 938
- indent attribute and, 938
- namespace declarations and, 932
- output rules, 930–932
 - XML Namespaces 1.1 and, 931
 - XML Namespaces and, 931
- script in stylesheet v. script in, 137
- valid HTML and, 936, 937
- XHTML output method v., 939
- XML output method v., 936–937

html tidy utility, 947

HTML/XHTML serialization

- percent encoding in URIs and, 776, 937

implementation-defined behavior

- HTTP header, 934**
 - external encoding information in, 905
 - media-type and, 934, 939, 940
 - HTTP request, 430**
 - hyperlinks**
 - in case study. *See* XML specification (case study)
 - ID attribute values and, 800
 - output, 435, 450, 797. *See also* generate-id()
 - hyphens. *See also* minus-sign**
 - allowed with square brackets, 920
 - character range and, 919
 - comment nodes with, 285
 - ignored in collations
 - substring-after() and, 886
 - substring-before() and, 888
 - minus-sign *v.*, 538, 572
 - subtraction (regex), 919
 - in XML names, 538
- I**
- «i» flag, 828, 925–926**
 - atom and, 925
 - back-references and, 925
 - character range and, 925
 - IANA (Internet Assigned Numbers Authority), 460, 820**
 - IBM mainframes, line ending characters on, 63**
 - IBM XSLT processor, 10**
 - id attribute, of <?xml-stylesheet?>, 98, 469–470**
 - identifier, fragment, 756, 868, 905**
 - identity, 593**
 - atomic values and, 190, 289
 - new, 293
 - node
 - document order/duplicates and, 602, 629
 - functions and, 988
 - node comparisons by, 591, 593, 797
 - «is» operator and, 591, 593
 - of sequences, 187, 633
 - identity constraints, 179**
 - identity template, 290, 296**
 - variant of, 290
 - identity transform (JAXP), 1169, 1177, 1185**
 - id() function, 62, 379, 704, 802–804**
 - generate-id() and, 800
 - ID attribute and, 802
 - key() *v.*, 804
 - literal and, 704–705
 - variable reference and, 704
 - xs:IDREF and, 802
 - xs:IDREFS and, 802
 - «idiv», 538**
 - floor() and, 780
 - integer division, 572, 573
 - «mod» and, 572, 573
 - numeric predicates and, 639
 - precedence, 572
 - xs:integer() constructor function and, 780, 872
 - IDKeyPattern construct, 704–705**
 - CSS and, 705
 - function calls and, 704
 - IDREF attribute, 62. *See also* xs:IDREF**
 - constraints, 251, 293, 297, 305
 - input-type-annotations and, 477
 - values, 179
 - idref() function, 62, 804–806**
 - xs:IDREF and, 805
 - xs:IDREFS and, 805
 - IDREFS attribute, 62. *See also* xs:IDREFS**
 - IDs, 61–62, 179, 1225. *See also* xs:ID**
 - candidate, 802
 - constraints, 251, 293, 297, 305
 - definition/description, 1225
 - DTDs and, 62
 - duplicated, 62
 - elements as, 803
 - hyperlinks and, 800
 - id() and, 802
 - input-type-annotations and, 477
 - invalid characters, 62
 - keys *v.*, 379
 - PathPattern and, 690, 691
 - validation and, 803, 805
 - XML, 379
 - whitespace-separated list of, 804
 - XML Schemas and, 61
 - IEEE 754 standard, 198, 576, 884**
 - «if» expression, 278, 354, 521**
 - boolean() and, 722, 771
 - effective boolean value and, 555
 - parentheses and, 528
 - substring() *v.*, 885
 - IfExpr construct, 525, 551**
 - keywords and, 525
 - if-then-else, 282, 283**
 - ignorable characters, in collations, 886, 888**
 - ignorable hyphens/spaces (in collations)**
 - substring-after() and, 886
 - substring-before() and, 888
 - IKVMC cross-compiler, 1195**
 - illustrations, numbering sequences for, 415**
 - immutability of sequences, 633**
 - implementation-defined behavior, 954–955**
 - collations. *See* collations
 - extension attributes, 954
 - extension declarations, 106–107, 954
 - extension functions. *See* extension functions
 - extension instructions. *See* extension instructions
 - extension types, 954
 - localization attributes, 954

implementation-dependent features

- implementation-dependent features, 955**
- implicit timezone, 567, 580, 583, 585, 718, 738, 806–807**
 - current-date() and, 583
- implicit-timezone() function, 716, 806–807**
 - dynamic context and, 567
- import precedence, 359–362, 1225. See also xsl:import**
 - of declarations, 357, 361–362
 - definition/description, 1225
 - determining, 359–361
 - effect of, 361–362
 - of global variables, 364–365, 502
 - for multiple xsl:import-schema declarations, 370
 - named templates, 485
 - priority and, 239, 243, 362, 400, 441, 484, 486
 - of template rules, 79, 94, 365–368, 484
 - xsl:apply-imports and, 239
 - xsl:attribute-set and, 267, 361
 - xsl:call-template and, 272
 - xsl:character-map and, 281, 361
 - xsl:decimal-format and, 299, 361
 - xsl:function and, 346, 361
 - xsl:import and, 94, 148, 357, 358, 362
 - xsl:import-schema and, 362, 370
 - xsl:include and, 148, 362
 - xsl:key and, 362
 - xsl:namespace-alias and, 362
 - xsl:output and, 362, 423
 - xsl:param and, 362
 - xsl:preserve-space and, 362
 - xsl:strip-space and, 362
 - xsl:template and, 243, 362
 - xsl:variable and, 362
- import tree, 238**
- importing. See also xs:import; xsl:apply-imports; xsl:import; xsl:import-schema**
 - schemas, 180–181, 368–372
 - with no target namespace, 482
 - XQuery and, 369
 - stylesheets, 237, 357–368
 - types from schema, 793
- include-content-type**
 - HTML output method and, 938
 - meta element and, 938, 940
 - XHTML output method and, 940
 - in xsl:output, 421, 422
 - in xsl:result-document, 447, 448
- #include directive, 374. See also xsl:include**
- including stylesheets, conditionally at runtime, 374**
- incompatibility**
 - comparing sequence of nodes to booleans, 589
 - comparing strings and, 588
- incremental stylesheet processor, 988**
- increments other than one, for numbering sequences, 412**
- indefinite precision, xs:decimal and, 575**
- indent attribute, 421, 422, 448, 934**
 - HTML output method and, 938
 - mixed content and, 934
 - XML output method and, 934
 - xml:space attribute and, 934
- indentation, 147, 934**
- index, created using modes, 490**
- index, used by xsl:key, 379**
- index-of() function, 807–808**
 - distinct-values() and, 750
 - items found in sequence with, 807
- Indic Arabic digits, 302**
- «INF», 190, 576, 660**
- «-INF», 190, 576, 660**
- infinity, 199**
 - positive/negative
 - «INF», «-INF», 199, 576, 660
 - xsl:decimal-format, 300, 302, 790
- infix operators, 526**
 - «//» as, 627
- information item, 62**
- Information Set, XML, 46, 62, 64, 65**
 - W3C Recommendation, 62
- inheritance, 393**
 - in object-oriented languages, 363
- inherit-namespaces, 289, 311. See also namespace inheritance**
 - = “no”, 291, 310
 - undeclare-prefixes and, 935
 - [xsl:], 113, 120, 310, 393
- initial context item, 543**
- initial context node, 68**
- initial item, of each group, 328, 330**
- initial mode (template processing), 68**
- initial node, of each group, 330**
- initial sequence, in sorting process, 457**
- initial template, 68, 1225**
 - definition/description, 1225
 - in knight’s tour example, 1102–1104
- initialization, conditional, 989–990**
- inline schema, 368**
 - document, 175, 369
- innermost ancestors, 542**
- input documents. See source documents**
- input-type-annotations, 99, 467, 477–478**
 - effect/usage, 477–478
 - ID/IDREF and, 477
 - Saxon and, 478
- in-scope functions, 559–560**
 - as constructor functions, 559
 - core library and, 559
 - as user-defined functions, 560
- in-scope namespaces, 555–556, 808, 864, 1225–1226**
 - definition/description, 1225–1226

- in-scope schema definitions, 557–558**
 - basic XSLT processors and, 557
 - xsl:import-schema and, 557
- in-scope variables, 558–559**
- in-scope-prefixes() function, 61, 601, 808–809**
 - namespace axis and, 808, 839. *See also* namespace axis
 - namespace nodes and, 808–809
 - namespace undeclaration and, 809
 - namespace-uri-for-prefix() and, 809
- insert-before() function, 212, 810**
- inserting items, 810**
- insignificant whitespace, 142, 465**
- instance of (a sequence type), 653**
- «instance of» operator, 165, 217, 371, 538, 548, 654, 668, 677–678**
 - sequence types and, 668
- instance() function (XForms), 544**
- InstanceofExpr construct, 526, 677**
- instants in time, xs:dateTime and, 580**
- instructions, 69, 89, 90, 108–125, 1226. *See also* extension instructions; processing instructions; xsl:instruction**
 - definition/description, 317–318, 1226
 - in extension namespace, 317
 - flow-of-control, 979
 - literal result elements v., 317
 - not recognized by XSLT processor, 318
 - in sequence constructors, 766
 - unknown, 318
 - as user-defined extensions, 318
 - as vendor-specific extensions, 318
 - XSLT, 108–110, 764
 - (full list), 109–110, 765
 - definition/description, 766
 - element-available() and, 764–770
 - in XSLT namespace, 317, 765
- IntegerLiteral symbol, 530**
- integers, 189. *See also* xs:integer**
 - arithmetic operators and, 574–575
 - division, 572, 573. *See also* «idiv»
 - maximum size, 574
 - overflow, 574
 - in reverse order, 637. *See also* reverse() function
- integrity constraints, result document, 174**
- Intel XSLT processor, 9, 10**
- interleaved markup, 942, 995–996**
- Intermediate Language, 1195**
- intermediate results, in variables, 510**
- internal DTD subset, 930**
 - character maps and, 942
- International Resource Identifiers. *See* IRIs**
- internationalization, 389**
- Internet Assigned Numbers Authority. *See* IANA**
- Internet Explorer, 68**
 - embedded stylesheets and, 102
 - HTML boilerplate generation and, 910
 - WD-xsl and, 467
 - <?xml-stylesheet?> processing instruction, 101
- interoperability, 460, 470, 567, 568**
- «intersect» operator, 57, 253, 353, 381, 538, 628–631**
 - node is member of given set and, 630
 - path expressions and, 599, 628. *See also* path expressions
 - XPath 1.0 and, 631
- IntersectExceptExpr construct, 526, 629**
- intersection operator. *See* «intersect» operator**
- interval between two dates/times, 577**
- invented name**
 - of namespace prefix, 256
 - as type annotation, 313, 314
- IRIs (International Resource Identifiers), 23**
 - converted to URIs, 811–812
 - URIs v., 812
- iri-to-uri() function, 811–812. *See also* encode-for-uri(); escape-html-uri(); escape-uri-attributes**
 - encode-for-uri() and, 772
 - percent encoding in URIs and, 811–812
 - percent signs and, 812
- «is» operator, 56, 538, 571, 593**
 - definition/description, 593
 - document() and, 756
 - «except» operator and, 593
 - generate-id() and, 797, 817
 - identical nodes and, 591, 593
 - NodeComp, 582
 - xsl:for-each-group and, 593
- is-id property, 802**
- is-idref property, 805**
- Islamic calendar, 787**
- ISO 639, 819**
- ISO 3166, 819**
- ISO 8601 durations, format for, 236, 785**
- ISO, as calendar name, 785, 786**
- ISO format for dates and times, 785**
- ISO week numbers, 787**
- iso-8859–1, 365, 424**
- Italian language, 405**
 - ordinals («^a», «^o»), 407
- item, context. *See* context item**
- item types, 219, 1226**
 - definition/description, 1226
 - required, 505, 547
 - in sequence types, 653
- item() item type, 653, 669, 713**
- items, 187–188, 189, 1226. *See also* atomic values; ‘first item’ rule; node(s); sequence(s)**
 - inserting, 807

ItemType construct

ItemType construct, 668

iteration

- over sequence, 324
- recursion v., 992–993

IXMLDOMDocument (MSXML interface), 1148–1149

IXMLDOMDocument2 (MSXML interface), 1148–1149

IXMLDOMNode (MSXML interface), 1150–1151

IXMLDOMNodeList (MSXML interface), 1151

IXMLDOMParseError (MSXML interface), 1151–1152

IXMLDOMSelection (MSXML interface), 1152

IXPathNavigable (.NET interface), 1159

IXSLProcessor (MSXML interface), 1152–1153

IXSLTemplate (MSXML interface), 1153–1154

J

JAR archive, 868

Jasmine, 30

Java, 187, 502

- APIs. *See* JAXP
- collation support, 731
- collations in, 561
- extension functions in, 134–135
 - linked to stylesheets, 136–137
- host language, 528
- mapping types to XPath, 961
- processors
 - binding in, 959–961
 - extensibility in, 959–961
- Saxon from command line on, 14–15, 103, 1196–1199
- Standard Edition 6, 13
- string comparisons in, 460
- system properties, 139
- user-defined functions in, 546
- Xalan-J and, 957
- XPath within, 553
- XQuery API for, 961

Java API, Altova and, 1219–1220

Java API for XML Processing. *See* JAXP

Java Architecture for XML Binding. *See* JAXB

Java methods

- calling, 546, 959–961
- calling from Saxon, 1206
- function-available() and, 796

Java object, wrapped, 960

Java Server Page, 430

Java servlet, 430. *See also* servlets

JavaScript, 107, 187, 502. *See also* Dynamic HTML;

ECMAScript; JScript

- API, 430
- client side, 286, 957
- commented-out, 286–287, 943
- as dynamically typed language, 959
- escape-uri-attributes and, 938

extension functions in, 134–135, 137

script and, 957

user-defined functions in, 546

Xalan-J and, 957

java.text.Collator, 561

javax.xml.parsers (JAXP package), 1164–1166

javax.xml.transform (JAXP package), 1171–1186

Saxon and, 1199

javax.xml.validation (JAXP package), 1199

javax.xml.xpath (JAXP package), 1199

JAXB (Java Architecture for XML Binding), 961

JAXP (Java API for XML Processing), 68, 86, 100, 359, 427, 1163–1193

DOM and, 1166–1169

encoding attribute and, 941

identity transform, 1169, 1177, 1185

parser API, 1164–1169

pipelines in, 1190–1193

result trees and, 421

SAX and, 1164–1166

Saxon and, 1163

serialization attributes as output properties in, 1181–1182

transformation API, 1169–1186

Saxon and, 1163, 1199–1202

transformations (examples), 1183–1193

documents in memory, 1188–1189

SAX pipeline, 1190–1193

supplying parameters/output properties, 1188

using files, 1187–1188

<?xml-stylesheet?> processing instruction, 1189–1190

URIResolver, 138, 567, 751, 755, 965, 1186

versions, 1163

Xalan and, 1163

JDK 1.1 specification, 788, 790

jEdit, 1196

special characters and, 941

joins

hash, 644

with multiple range variables, 647

optimization of

with keys, 644

with Saxon-SA, 644

relational, 643

«some» expression and, 647

unordered() and, 902

JScript. *See also* ECMAScript; JavaScript

extension functions in, 134, 709, 957

MSXML and, 957

transform document (example), 1154–1157

Julian calendar, 786, 787

Gregorian calendar and, 786

K**Kennedy family tree. See family tree (case study)****Kennedy, John F., 1049, 1051****Kernow development environment, Saxon and, 10, 13–14, 1195****key definitions, 812, 813****key values, 813****key() function, 85, 325, 377, 384, 704, 812–818**

associative access to nodes with, 813

grouping nodes and, 813

id() v., 804

literal and, 704–705

patterns in xsl:key and, 683

variable reference and, 704

xsl:key and, 812

key() pattern (example), 706–707**keys, 376. See also xsl:key**

concat() function and, 730

as cross-references, 814–816

grouping. *See* grouping keys

IDs v., 379

locating nodes by value with, 814

match attribute and, 377

multiple definitions for same, 383–384

multiple named, 383

multivalued, 381

name of, 377

node selection in any document with, 378

nodes with same, 378

optimization of joins with, 644

PathPattern and, 690

for same nodes, 383

simple, 379–380

sort. *See* sort key(s)

values of

use attribute and, 377

keywords

«comment», 616

ForExpr construct and, 525

IfExpr construct and, 525

«node», 617

«processing-instruction», 616

QuantifiedExpr construct and, 525

«text», 616

KindTests, 613, 616–617, 668, 670, 697–701

attribute(), 617

for attribute nodes, 699–700

simple type definition and, 700

child axis and, 695

comment(), 613, 700

for document nodes, 700

document-node(), 617, 700

element(), 617, 700

for element nodes, 698–699

examples of, 617

global attribute declarations and, 699

matching nodes and, 670–672

node(), 613

node kind and, 697

NodeTests as, 613

PatternStep and, 697–701

processing-instruction(), 613, 616, 700

schema-defined type and, 613, 698

schema-element(), 700

schema-element and, 699

substitution groups and, 701

syntax, 697–698

template rule priority and, 701

text(), 613, 700

type annotation and, 697

type hierarchy and, 701

Kleene operators, 669. See also «*»; «+»; «?»**knight's tour (case study), 221–222, 224, 1099–1114**

algorithm, 1100–1104

displaying final board, 1105–1106

finding possible moves, 1107–1109

finding the route, 1106–1107

initial template, 1202–1204

observations, 1112–1113

placing the knight, 1104–1105

running stylesheet, 1112

selecting best move, 1111–1112

trying possible moves, 1109–1110

Konqueror, 68**L****labels, type. See type labels****lang attribute. See also xml:lang**

of xsl:number, 409

of xsl:sort, 456, 460

lang() function, 819–820

xml:lang and, 819

language argument, 784**language code, 819****language-dependent numbering and sorting, 134****languages (natural)**

Bulgarian, 409

for days of week, 784

German, 409

«^B» in, 827

Greek, 186, 429

« α , β , γ , δ » for numbering sequence, 409

character block, 922

polymorphic (many shapes) from, 186« Σ » (sigma), 827

tunnel parameters and, 429

zero sign, 460

Hebrew, 410

languages (natural) (continued)

languages (natural) (continued)

- Italian, 405
 - ordinals («^a», «^o»), 407
 - lang() and, 819–820
 - for months of year, 784
 - non-English, 409
 - Oriental, 827
 - preferred, 410
 - Russian, 409
 - Spanish, 731, 732, 781
 - Swedish, 561, 848
 - xsl:decimal-format and, 301
 - xsl:message and, 389
 - xsl:number and, 409
 - xsl:sort and, 456, 460
- languages (programming), 546. See also specific programming languages**
- functional. See functional programming languages
 - read-only, XPath as, 188, 526
 - templating, 973
 - for writing extension functions, 957
- last or first item in list (position()), 856**
- last specified template rule, 362**
- last() function, 85, 241, 323, 333, 820–824. See also context size; count(); position(); xsl:number**
- «//» operator and, 703
 - binary «/» and, 605
 - context size and, 564, 608, 620, 639, 820–824
 - count() v., 734
 - filter expressions and, 822
 - in patterns, 685, 823
 - positional predicates and, 702
 - within select expression of xsl:sort, 823
 - xsl:apply-templates and, 734, 821
 - xsl:for-each and, 734, 821
- Latin letters, accented, 534**
- Latin scripts, non-, 491, 534**
- lax (validation =), 173, 259, 297, 305, 371**
- top-level schema declaration and, 314
- layer, customization, 368, 429**
- layer, stylesheet, 360**
- lazy evaluation, 138, 487, 988**
- memory and, 637
 - «to» operator and, 637
- LDAP directory, 42, 192**
- «le» operator, 538, 571, 581**
- default collation and, 584
 - NaN and, 586
 - ordered data types and, 584–586
 - ValueComp, 582
- leading «//», 691**
- leading minus-sign, 530**
- leading whitespace, 142, 493**
- normalize-space() and, 493, 845–846
- leading zeros, 300, 783, 790**
- leap seconds, 578**

least common type, 831

legal documents, sequence numbers for, 418

length, of strings, 880–881. See also zero-length strings

level attribute (xsl:number), 404

- level=any, 404, 415–417
 - numbering lines of poem (example), 417–418
 - rules, 407
- level=multiple, 404, 418–419
 - recursive structures and, 419
 - rules, 407
- level=single, 404
 - example, 413–415
 - rules, 406
 - usage, 411–412

lexical analysis, 528–529. See also tokenizing

lexical constructs (XPath), 527–539. See also comments; name(s); numeric literals; operators; string literals

lexical QName, 50, 201, 228, 534, 1226. See also xs:QName

- definition/description, 228, 1226
- local part, 50
- prefix, 50
- QName() and, 858–859

lexical representations, of booleans, 552

lexical rules (XPath), 521, 524

- comments and, 524
- syntactic rules v., 524
- whitespace and, 524

lexical tokens

- whitespace between, 524

LF. See linefeed

libraries. See function libraries

libxslt processor, 10

line ending characters, 63, 142, 440, 939, 940. See also newline characters

linefeed (LF), 64, 440

- carriage-return, 64

linked tree (in Saxon), 1205

Lisp programming language, 429, 986

list(s)

- comma-separated, 189
 - values enclosed in parentheses, 189
- construction of, 635. See also comma operator
- first or last item in, 856
- of names, formatting, 356
- of numbers, 407–408
 - aggregating (example), 993–995
- whitespace-separated, 113
 - of IDs, 804
 - normalize-space() and, 846

list types, 53, 54, 163, 164, 1226

- as content of node, 164
- definition/description, 1226
- simple types as, 653, 670

list-valued attribute, atomization of, 353

1268

literal(s), 530. See also numeric literals; string literals

id() function and, 704–705
key() function and, 704–705

literal constants

xs:decimal, 191, 530
xs:double, 191, 530
xs:integer, 191, 530
xs:string, 191. *See also* string literals

Literal construct, 539

literal result elements, 69, 73, 111–121, 176, 228, 260, 262, 476, 536, 1226

attributes on, 116–117, 255
 xsl:attribute v., 255
 xsl:prefixed, 112–114
content, 114
default-collation on, 468
default-validation and, 470, 471
definition/description, 228, 1226
effect, 114
exclude-result-prefixes on, 468
extension instruction v., 766
extension-element-prefixes on, 468
format, 112
HTML table example, 116
instructions v., 317
named attribute set and, 117
namespace inheritance, 120
namespace nodes on, 118–119, 392, 394
notation for description of, 112
position, 112
usage, 114–116
use-attribute-sets attribute of, 268
use-when on, 468
validation, 176
version attribute on, 468
xpath-default-namespace on, 468
xsl:attribute and, 117
xsl:element v., 306, 308, 315
xsl:namespace and, 121
[xsl:]type attribute of, 370

literal string value, assigned to variable, 504

literal text, 492

local declarations, 298

of elements, 315

local name

of attribute, 260
of element, 308

local parameters

to functions, 425, 427
to templates, 425, 427

local part, of lexical QName, 50

local part, of name of node, 824

local type definition, 54, 315

local variables, 81, 487, 512, 1226

convenience, 507
definition/description, 1226

forward reference to, 502
with same name, 502
scope of, 70, 501, 502
tunnel parameters v., 431
xsl:variable and, 501

Locale Data Markup Language, 392

locale-sensitive case mappings, 828

localization, 107, 301

attributes, implementation-defined, 954
of messages, 389–390, 410

local-name, NameTests and, 615

local-name-from-QName() function, 202, 826–827

xs:NCName and, 826
xs:QName and, 826

local-name() function, 51, 824–826. See also name(); namespace-uri()

LocalPart construct, 534

locating

items in sequence, 807–808
nodes
 key() and, 814
 list of functions for, 711
 schemas, 314
 catalog/repository and, 314
 xsi:schemaLocation and, 314
stylesheet module, 357–359
text within document (example), 413–415

look-up tables, 107, 321, 510, 753

messages as, 510
in stylesheet, 761–763

loops, nested, 643

lower-case() function, 21, 204, 827–828

case variants and, 926
case-insensitive comparison with, 828

lower-case/upper-case letters, 461. See also case-order attribute

«lt» operator, 458, 538, 571, 581

default collation and, 584
max(), min() and, 831
NaN and, 586
ordered data types and, 584–586
ValueComp, 582

M

«m» flag, 926

«\$» and, 926
«&» and, 926
multiline mode and, 926
string mode and, 926

major atomic types, 193–206. See also atomic types

major-to-minor order, 242, 324, 456. See also sort key(s)

mandatory parameter, 428

“A Manifesto for Adding SGML Intelligence to the World-Wide Web,” 27

map operator (in functional programming languages)

map operator (in functional programming languages), 604

mapping Java to XPath types, 961

mapping sequences, 641–642

mappings, case, 827. *See also* lower-case(); upper-case()

mappings, from URIs to documents, 568

markup

- comments with, 287
- double markup, 942
- interleaved, 942, 995–996
- parallel, 995–996
- standoff, 995–996

match attribute

- errors and, 485, 688
- of xsl:key, 377
- of xsl:template, 68, 484–485
 - errors and, 485
 - global parameters and, 485
 - global variables and, 378, 485
 - pattern, 484
 - template rules and, 487
 - xsl:apply-templates and, 484

match patterns. *See* patterns

matches() function, 204, 234, 828–830

- ends-with() and, 774
- regex and, 828–830
- regex syntax and, 915–927
- replace() v., 898

matching. *See also* comparing

- atomic values, 669–670
- elements/attributes, 672–677
- newline characters in regex, 917, 926
- nodes, 670–672
- spaces, in regex, 921, 924
- substrings with collations, 731–732, 876
 - minimal, 886, 888
- by type, 219–221, 981
- words, in regex, 921

matching document-node() function

- well-formed document and, 671, 700

mathematical concept of function, 987

mathematical concept of variable, 989

mathematical function library, 135, 200, 346

math:sqrt() function, 346

max() function, 200, 275, 830–832

- comma operator and, 635, 831
- «lt» operator and, 831
- one-or-more() and, 854

maximum size of integer, 574

maximum/minimum values in sequence, 820–822

media attribute (<?xml:stylesheet?>), 101

- Firefox and, 101
- Internet Explorer and, 101

media-type, 934

- fragment identifier and, 756
- HTML output method and, 939
- HTTP header and, 934, 939, 940
- of stylesheet modules, 470
- and unparsed-text(), 905
- XHTML output method and, 940
- XML output method and, 934
- in xsl:output, 421, 422
- in xsl:result-document, 447, 448

MEI. *See* Music Encoding Initiative

memory

- document built in, 752
- free, 137
- holding documents in
 - JAXP interface, 1188–1189
- lazy evaluation and, 637

merging attribute sets, 267

merging character maps, 282

messages, 386–390. *See also* xsl:message

- examples, 388–390
- localized, 389–390, 410
- as look-up data, 510
- in native language, 389
- order of, 388

meta element, 422, 447

- HTML (charset attribute), 938, 939, 940, 941
- XHTML (charset attribute), 939, 940, 941

metacharacters (regex)

- «\$», 233
- «m» flag and, 926
- «.», 232, 917, 926
- «s» flag and, 917, 926
- «\», 233, 920
- «m» flag and, 926
- (list), 232

meta-stylesheets, 393, 396

method (serialization parameter), 929

method attribute, 421, 446, 448

micropipeline, 88, 402

Microsoft

- APIs, 100, 427
- cp1252, 937
- MSXML processor. *See* MSXML processor
- .NET framework, 9, 15
- System.Xml.Xsl processor. *See* System.Xml.Xsl processor
- WD-xsl and, 467

midnight (time value), 743, 800

MIME type, 422, 447, 470, 756, 905, 934, 939, 940.

See also media-type

min() function, 200, 275, 830–832

- «lt» operator and, 831
- one-or-more() and, 854

minimal matching substring, 886, 888

- minor atomic types, 193, 206–210. *See also***
 - atomic types**
 - binary, 208
 - partial date, 207
 - single-precision floating point, 208
 - xs:duration, 205, 208–209
 - xs:NOTATION, 180, 190, 209–210
- minimum/maximum values in sequence, 830–832**
- minus-sign, 538. *See also* «-»; additive operators;**
 - hyphens; unary minus operators**
 - hyphen v., 538, 572
 - leading, 530
 - xml:decimal-format, 300, 789, 790
- minutes component**
 - in format-time(), 782
 - from xs:date, 832
 - from xs:dateTime, 832
 - from xs:duration, 832–833
- minutes-from-dateTime() function, 832**
- minutes-from-duration() function, 832–833**
- minutes-from-time() function, 832**
- mixed content, 53, 146, 155–156, 441, 499**
 - indent attribute and, 934
 - schema types and, 742
- «mod» operator, 200, 417, 538, 571**
 - «div» and, 572, 575
 - «idiv» and, 572, 575
 - precedence, 572
 - remainder from integer division and, 572, 575
- mode attribute, 242, 484, 486**
- modes (compatibility). *See* backward-compatibility mode; forward-compatibility mode**
- modes (regex evaluation)**
 - case-insensitive, 231, 925
 - «i» flag and, 925–926
 - dot-all, 231, 926
 - «s» flag and, 917, 926
 - multiline, 231, 918
 - «m» flag and, 926
 - string, 926
- modes (template processing), 68, 78, 88, 242, 274, 1226**
 - current, 238, 242, 273, 346, 485, 486
 - default (unnamed), 238, 242, 485, 486
 - definition/description, 1226
 - document() and, 758
 - examples, 247–251, 490–491
 - index creation with, 490
 - initial, 68
 - named, 249
 - not sticky, 242
 - sticky, 243
 - table of contents and, 490
 - template rules and, 242, 402
 - tunnel parameters and, 431
 - xml:apply-templates and, 247–251, 484, 486
 - xml:include and, 374
- modular structure, of stylesheets, 90–98**
- modules, stylesheet. *See* stylesheet(s)**
- month component**
 - in format-date(), 782
 - language for, 784
 - from xs:date, 833–834
 - from xs:dateTime, 833–834
 - from xs:duration, 834
- month-from-date() function, 212, 833–834**
- month-from-dateTime() function, 833–834**
- months-from-duration() function, 834**
- MSXML processor (Microsoft), 9, 66, 137, 143, 567, 767, 1147–1158**
 - AltovaXML and, 1216
 - binding extension functions in, 957–959
 - in browser (example), 1095–1098
 - COM API and, 1147–1153
 - DOM in, 1147–1152
 - extensibility in, 957
 - handling whitespace nodes, 567, 1158
 - JScript and, 957
 - transform document (example), 1154–1158
 - normalization of text nodes, 1158
 - objects in, 1148–1154
 - parser, 388, 440
 - xml:message and, 388
 - restrictions, 1158
 - transformNode(), 941
 - VBScript and, 957
 - versions, 1148
- msxml:node-set(), 321**
- msxsl:script, 107, 134, 137, 477**
- Muench, Steve, 816**
- Muenchian grouping method, 816**
- multicharacter escapes, 921**
- multilevel grouping (example), 333–335**
- multiline mode (regex evaluation), 231, 918**
 - «m» flag and, 926
- multiphase transformations, 68, 87–88, 90, 402, 991–992**
 - example, 511–513
 - intermediate results of, 510
 - pipelines and, 991–992. *See also* pipelines
- multiple input sequences, 643–644**
 - «for» expression and, 643–644
- multiple level. *See* level attribute**
- multiple named keys, 383**
- multiple result documents, 44–45, 435, 445, 450**
 - examples, 451–452, 767–769
 - element-available() and, 767–769
 - XSLT 1.0 processors and, 44–45, 319–320
- multiple source documents, 44–45, 326, 482, 510, 692**
 - «/» pattern and, 692

multiple xsl:import-schema declarations

multiple xsl:import-schema declarations, 370
multiplication of durations, 577, 579
multiplicative operators, 200. *See also* «*»; «div»; «idiv»; «mod»
MultiplicativeExpr construct, 526, 572
multivalued keys, 381
Music Encoding Initiative (MEI), 6
music transformation, XSLT and, 5–7
musical symbols, 63
MusicXML, 6–7

N

name(s)

attribute. *See* attribute names; name attribute
characters, matching in regex, 921
decimal-format, 299, 789
element. *See* element names
function. *See* function names
list of
 formatting, 356
local. *See* local name
of nodes, 46, 47, 50–51. *See also* local-name();
 name(); node-name()
 comparing, 859
 local part of, 824
 name() and, 835–837
of parameters, 428–429
same
 user-defined functions and vendor-supplied functions
 with, 346
timezone, 787
type. *See* type names
unprefixed. *See* unprefixed names
of variables. *See* variables
name(s) (in XPath expressions), 534–537. *See also*
 LocalPart; **NCName**; **NCNameChar**; **Prefix**; **QNames**
 case-sensitive, 535
name attribute, 485
 name of key and, 377
 name of namespace node and, 390
 xsl:call-template and, 124, 272, 485
named attribute sets. *See* attribute sets
named decimal format, 299
named global types, 313
named modes, 249
named templates, 95, 107, 124, 362, 1226
 definition/description, 1226
 evaluated at runtime, 272
 import precedence, 485
 invoking, 271
 recursive, 350
 stylesheet function as wrapper for, 349
 stylesheet functions v., 347, 349–350
 as wrapper for stylesheet function, 349

name() function, 51, 835–837. *See also* local-name();
 namespace-uri(); **node-name()**
 namespace prefix and, 835
 principal node kind and, 836
 «self» axis and, 836
namespace(s) (.NET assembly)
 System.Xml, 138
namespace(s) (XML), 23–24, 1226. *See also*
 Namespaces 1.1, XML
 aliasing for, 397
 attribute nodes v., 256
 declarations. *See* namespace declarations
 default. *See* default namespaces
 definition/description, 1226
 extension, 138
 for extension functions, 471
 for functions, 547
 functions in same, 349
 in-scope. *See* in-scope namespaces
 null, 58, 866
 starting with http://www.w3.org, 345
 target, 180, 370
 undeclarations. *See* undeclarations, namespace
 in XDM tree model, 59–60
 XML Schema, 99, 135
namespace alias, 121, 263. *See also*
 xsl:namespace-alias
 stylesheet generation and, 121
 for xsi namespace, 183
namespace attribute
 xsl:attribute, 255
 xsl:element, 308
 xsl:import-schema, 180
namespace axis, 61, 601, 609, 611, 808, 1226. *See*
 also in-scope-prefixes(); **namespace-uri-for-prefix()**
 definition/description, 1226
 deprecation, 61, 601
 in-scope-prefixes() and, 808, 839
 namespace declarations and, 611
 namespace nodes and, 611, 617
 order of, 611
 namespace-uri-for-prefix() and, 808, 839
 node tests and, 613
 ordering of nodes on, 620
namespace declarations, 56, 58, 358, 472, 555, 1226
 attribute nodes and, 610
 attribute value templates and, 124
 default. *See* default namespace declarations
 definition/description, 1226
 duplicate, 119, 309, 473
 generating, 256
 namespace axis and, 611
 namespace nodes v., 391, 932
 with output document, 262
 QNames and, 535

- scope, 58
- from serializer, 391
- unwanted/unnecessary, 471, 473–474, 475
- WD-xsl and, 467
- XML output method and, 932
- XPointer, 556
- XQuery, 556
- xsl:attribute and, 391
- on xsl:stylesheet, 99, 466–467
- XSLT namespace, 466, 765
- namespace fixup, 256, 262, 289, 291, 296, 310–311, 391, 392, 474–475, 1226**
 - definition/description, 310–311, 1226
 - namespace nodes acquired with, 310–311, 392, 393
 - prefixes changed with, 310–311
 - prefixes in result document and, 397
 - prefix-from-QName() and, 857
 - schema validation and, 393
 - temporary trees and, 932
 - XML output method and, 932
- namespace <http://www.w3.org/2001/XMLSchema> (xs namespace prefix), 135, 190**
- namespace inheritance, 120, 310, 311–312**
 - disable, 312
- namespace nodes, 49, 56, 59, 262, 289, 471, 1226**
 - acquiring, 392–393
 - element as child of another element and, 392–393
 - literal result elements and, 392
 - with namespace fixup, 310–311, 392, 393
 - xsl:copy and, 392
 - xsl:copy-of and, 392
 - xsl:namespace and, 393
 - adding, with namespace fixup, 310–311
 - construction of
 - xsl:namespace and, 390
 - copied from source document, 474
 - copying, 296–297, 311
 - creation of, 60
 - deep-equal() and, 746, 747
 - definition/description, 1226
 - getting rid of extraneous, 291
 - in-scope-prefixes() and, 808–809
 - on literal result elements, 118–119, 394
 - name attribute and, 390
 - namespace axis and, 611, 617
 - namespace declarations v., 391, 932
 - namespace-sensitive content and, 60–61
 - namespace-uri-for-prefix() and, 839–840
 - order of, 393
 - redundant, 394
 - string value, 51
 - select attribute and, 391
- namespace prefixes, 58, 121, 1226–1227**
 - in content of elements/attributes, 392
 - definition/description, 1226–1227
 - DTD-based validation and, 857
 - invention of, 256
 - name() and, 835
 - in names of variables, 541
 - namespace URI corresponding to given, 839
 - NameTests and, 615, 696, 697
 - QNames, 262, 535
 - in scope, 808–809
 - in xs:QName, 857
- namespace URIs, 50–51, 58, 1227**
 - aliasing of, 396
 - changing, 296
 - corresponding to given prefix, 839
 - default
 - for function names, 545
 - definition/description, 1227
 - equality of, 58
 - expanded QNames and, 50–51, 840–841
 - <http://www.w3.org/XML/1998/namespace>
 - xml prefix mapped to, 311
 - as part of element name, 308
 - relative, 55
 - zero-length strings and, 391
- namespace, XML, 472**
- namespace, XSLT, 466, 471, 475**
- Namespaces 1.1, XML, 23–24, 119, 443, 471**
 - overview, 58–59
 - Recommendation, 24, 58, 474, 534
 - standard, 522
 - undeclarations. *See* undeclarations, namespace
 - XML output method and, 930
- namespace-sensitive content, 60–61, 260, 310, 311**
- namespace-uri-for-prefix() function, 61, 839–841. *See also* in-scope-prefixes(); resolve-QName()**
 - in-scope-prefixes() and, 809
 - namespace axis and, 839. *See also* namespace axis
 - namespace nodes and, 839–840
 - namespace undeclaration and, 840–841
- namespace-uri-from-QName() function, 202, 841–842. *See also* local-name-from-QName(); QName()**
- namespace-uri() function, 51, 615, 837–839. *See also* local-name(); name()**
 - xs:anyURI and, 838
- NameTests, 439, 613, 614–616, 696–697**
 - «*», 440, 696
 - attribute axis and, 695
 - in AxisStep, 482
 - child axis and, 695
 - examples of, 615–616
 - forms of, 440
 - local-name and, 615
 - namespaces and, 615, 696, 697
 - variants of, 697
 - XHTML, 697
 - NCName as, 440

NameTests (continued)

NameTests (continued)

- no namespace prefix and, 615, 697
- NodeTests as, 613
- PatternStep and, 696–697
- principal node kind and, 613, 614, 696
- priority and, 440
- RSS and, 697
- schema-aware processor and, 696
- XPath default namespace and, 696

NaN (not-a-number), 80, 199, 302, 406, 454, 1226

- definition/description, 1227
- «eq» and, 329
- equal to itself, 459
- general comparisons and, 589
- «lt», «le», «ge», «gt», and, 586
- number() function and, 573, 852
- string converting from double and, 660
- sum() and, 889
- xs:double and, 586, 852
- xs:float and, 586
- xsl:decimal-format, 300, 790

narrative XML, 53

native language, messages in, 389. See also languages

navigating

- graphs, 275
- hierarchies, 275
- trees, 189

navigational stylesheets (design pattern), 976–980, 1114. See also family tree (case study)

- procedural v., 979

Naxos Breakfast Supplies example, 380

NCName, 534. See also xs:NCName

- as NameTest, 440

NCNameChar, 534

NDATA (non-XML data), 209, 903, 906

«ne» operator, 538, 571, 581

- «eq» v., 584
- operand types for, 583–584
- ValueComp, 582

negative dates, 196

negative durations, 578, 580

negative group (character group), 919

negative infinity

- «-INF», 199, 576, 660
- xsl:decimal-format, 300, 302, 790

negative numbers, 199

- parentheses for, 301, 790
- sub picture for, 301

negative subpicture, 790

negative zero, 199, 576

- ceiling() and, 723

nested comments, 529

nested groupings, 334

nested loops, 643

nested sequence constructors, 73–74

nesting

- curly brackets and, 123
- sequences and, 188, 633

.NET framework

- Altova and .NET API, 1220
- dynamic loading of assemblies in, 962
- languages, extension functions in, 957
- Saxon on, 15
 - C# and, 957
- System.Xml namespace in, 138
 - classes, 567
- System.Xml.Xsl processor. *See* System.Xml.Xsl processor
- XmlResolver, 138, 567, 751, 755, 965

network, of related objects, 351. See also graphs

New Year's day, 787

newline characters, 63, 532

- (x0A), 917, 926
- line ending in XML, 63
- matching in regex, 917, 926
- serializing, 939, 940
- in string literal, 532

NFC (normalization form C), 848

- accents and diacritics in, 880, 884
- normalize-unicode() and, 848, 881

nilability, 182–183. See also xsi:nil

- nilable elements, 183
- nilled elements, 183
- nilled flag, 477

nilled property, 843

nilled() function, 842–843

- schema validation and, 842
- xsi:nil and, 843

node(), 617, 685, 698, 713

- child axis and, 671
- KindTest, 613
- NodeTest, 162

node(s), 49–57, 1227

- ancestors of. *See* ancestors
- associative access to, 813
- attribute. *See* attribute nodes
- base URI, 54–55
- children of. *See* children, of nodes
- classification methods, 50
- comment. *See* comment nodes
- comparisons. *See* node comparisons
- context. *See* context node
- copied, 57
 - validation of, 297
- creation of
 - from functions, 550
 - XSLT instructions/templates and, 550
- definition/description, 1227
- depth of, 274
- document. *See* document nodes

1274

- document order, 57–58
- document() applied to, 757–758
- duplicate, 328
 - removal of, 378, 544, 603
- element. *See* element(s)
- even-numbered, 241
- extension functions and, 961
- hierarchic position of, 407
- identity
 - document order/duplicates and, 602, 629
 - functions and, 988
- keys for same, 383
- kinds. *See* node kinds
- matching, sequence type descriptors and, 670–672
- names of, 46, 47, 50–51. *See also* local-name(); name(); node-name()
 - comparing, 859
 - local part of, 824
 - name() and, 835–837
- namespace. *See* namespace nodes
- numbering of, 403
- odd-numbered, 241
- parent. *See* parent nodes
- parentless. *See* parentless nodes
- peer, numbering of, 406
- processing instruction. *See* processing instructions
- returning references to, 453, 455
- root. *See* root nodes
- with same key, 378
- in sequences, 187–188, 633. *See also* sequence(s)
- sets of. *See* sets of nodes
- string value. *See* string value
- test contents of, 552
- text. *See* text nodes
- type annotations of. *See* type annotations
- typed values of. *See* typed values
- UML class hierarchy, 50, 56–57
- URIs held in, 757
- node comparisons, 593–594. *See also* «[[»; «]]»; «is»**
 - changes in XPath 2.0, 594
 - by identity, 591, 797
- node kinds, 46, 47, 49, 607, 1227**
 - built-in template rule, 79
 - definition/description, 1227
 - NameTest and, 697
 - PathPattern and, 690
 - PatternStep and, 694
 - principal, 613, 614
- node tests, 528, 599, 613–614**
 - attribute axis and, 613
 - in axis steps, 606
 - examples of, 613–614
 - function names and, 528
 - namespace axis and, 613
- NodeComp construct, 582**
- NodeInfo interface (in Saxon), 1205**
- NodeName construct, 698**
- node-name() function, 51, 843–845**
 - expanded name of node and, 843–844
 - QName() and, 859
 - xs:QName and, 843–844
- node-set() extension (example), 794–796**
- node-set() extension function, 87**
- node-sets, 32, 39, 186, 601, 629, 633. *See also* sequence(s)**
- NodeTests**
 - «*», 617
 - attribute(), 162
 - axis steps and, 607
 - document-node(), 162
 - element(), 162, 164
 - as KindTests, 613
 - as NameTests, 613
 - node(), 162
- no-namespace schema (example), 372**
- non-ASCII characters, 811, 812**
 - escaping, 776
- nonbreaking space characters, 38, 142, 148, 281, 367, 732, 880, 938**
 - « », 64, 281, 367, 938
 - character maps and, 943
 - normalization of, 880
- noncomparable types, 329**
- non-consecutive numbering sequences, 416**
- non-document tree, 626**
- nonempty node sequence, testing for, 276**
- non-greedy quantifiers (regex), 232, 863, 917**
- non-Gregorian calendar, 786**
- non-Latin scripts, 534**
 - weights in, 491
- non-schema-aware processors. *See* basic XSLT processors**
- nonspacing characters. *See* nonbreaking space characters**
- nonunique keys (example), 381–383**
- non-validating XML parser, 803, 805, 904**
- non-XML data (NDATA), 209, 903, 906**
- normal characters (regex), 917**
 - atoms as, 917
- normalization**
 - attribute value, 64, 142, 532
 - of combining character, 880, 884
 - duration value, 745
 - early, 848
 - forms, 847, 848
 - of nonspacing characters, 880
 - of strings, 849
 - term, 144

normalization (continued)

normalization (continued)

of Unicode characters, 422, 447, 561, 731, 847–849
whitespace, 528, 663, 665

normalization form C. *See* NFC

normalization-form, 421, 422, 447, 448, 934

HTML output method and, 939
normalize-unicode() and, 934
XML output method and, 934

normalize-space() function, 144, 148, 204, 215, 665, 845–846. *See also* concat(); substring-after(); substring-before()

leading/trailing whitespace and, 493, 845–846
whitespace and, 493, 845–846
whitespace-separated list of values and, 846
xsl:strip-space v., 846
xs:normalizedString v., 144, 215, 846
xs:whiteSpace facet and, 846

normalize-unicode() function, 204, 847–849. *See also* serialization

character normalization and, 848–849
concat() and, 849
NFC and, 848, 881
normalization-form and, 934
string-join() and, 849
string-length() and, 880–881

not sticky modes, 242

not-a-number. *See* NaN

NOTATION. *See* xs:NOTATION

notation, XPath, 522–523

not() function, 135, 552, 850–851. *See also* boolean(); false(); true()

«! =» operator and, 850–851
«=» operator and, 850–851
effective boolean value and, 850

Novatchev, Dimitre, 251, 347, 955

FXSL library of, 251, 347, 546, 955

null namespace (XML), 58, 866

null values, 57, 203

SQL concept of, 182
empty sequence and, 586

number(s), 201. *See also* infinity; NaN; xs:decimal;

xs:double; xs:float; xs:integer; xsl:number; zero(s)
absolute value of, 714
arithmetic. *See* arithmetic operators
average of. *See* avg()
cardinal, 409, 783
comparing, 583, 585
promotion and, 585
constant. *See* literal constants
converted to strings, 531
double precision. *See* double precision numbers
formatting, 403, 408–410, 788–791
fractional part of, 790
«#» and, 790
functions. *See* numeric functions

Greek, 409

Hebrew, 410, 783

list of, 407–408

negative. *See* negative numbers

ordinal, 409, 783

Italian («^a», «^o»), 407

predicates as. *See* numeric predicates

Roman numeral, 409, 410, 782

rounding. *See* rounding numbers

section, 405

sequence. *See* sequence numbers

term, 1227

totals of, 889–890. *See also* sum()

as words, 783

number of digits displayed, 300

number of items in sequence, 733

number types. *See* numeric types

number() function, 200, 454, 458, 588, 851–853. *See also* string()

arguments automatically converted to numbers with, 573, 851

backward-compatibility mode and, 852

«castable as» operator and, 853

conversion to xs:double and, 852

exponential notation and, 851

leading «+» sign and, 851

NaN and, 573, 852

numeric comparison and, 588

predicates and, 853

xs:double and, 851

numbering

of days of week, 785

of equations, 407, 415

of weeks of year, 785

numbering schemes, 784

numbering sequences. *See also* sequence numbers

for actions from meeting, 415

for equations, 407, 415

for figures, 415

for footnotes, 407, 410, 415, 417

Greek, 409

Hebrew, 410, 783

for illustrations, 415

in increments other than one, 412

language-dependent, 134

for lines of poem, 417–418

for nodes, 403

non-consecutive, 416

for peer nodes, 406

Roman numeral, 409, 410, 782

for siblings, 406, 411–412

starting at one, 409

starting at value other than one, 412

for tables, 415

vendor-defined, 134

- as words, 409, 783
 - XSL Formatting and, 410
 - numerals, Roman, 409, 410, 782**
 - numeric calculations. *See also* arithmetic operators;**
 - numeric functions**
 - precision of, 531
 - «div» and, 575
 - indefinite, 575
 - xs:decimal and, 575
 - numeric character references. *See* character references**
 - numeric comparison operators, 200**
 - numeric equality, 200, 583**
 - numeric formats, 1141. *See also* xs:decimal; xs:double;**
 - xs:float; xs:integer**
 - backward-compatibility mode and, 1141
 - numeric functions. *See also* specific numeric functions**
 - abs(), 200, 714–715
 - avg(), 200, 718–719
 - ceiling(), 200, 723–724
 - floor(), 200, 779–780
 - format-number(), 298, 531, 576, 788–792
 - list of, 710
 - max(), 200, 275, 830–832
 - min(), 200, 275, 830–832
 - number(), 200, 454, 458, 588, 851–853
 - round(), 200, 870–872
 - round-half-to-even(), 200, 212, 531, 576, 872–873
 - sum(), 200, 889–890
 - numeric literals, 530–532**
 - decimal places and, 531
 - XML Schema and, 531
 - numeric order, 585**
 - numeric output, formatting, 531. *See also***
 - format-number(); xsl:number**
 - numeric overflow, 659, 661**
 - numeric predicates, 608, 618, 619, 638**
 - effective boolean value and, 638
 - «idiv» and, 639
 - number() and, 853
 - position() and, 619, 638
 - numeric ranges, 636–637. *See also* «to» operator**
 - numeric type promotion, 81, 427, 458, 506, 548, 713, 1228**
 - arithmetic operators and, 572, 585
 - xs:decimal, 572, 585
 - xs:double, 572, 585
 - xs:float, 572, 585
 - xs:integer, 572, 585
 - numeric types, 506, 549, 713. *See also* xs:decimal;**
 - xs:double; xs:float; xs:integer**
 - derived, built-in, 192, 193, 210–213
 - numeric values. *See* numbers**
 - NumericLiteral construct, 539**
- O**
- «0» (Italian ordinal), 407**
 - object-oriented programming**
 - inheritance in, 363. *See also* xsl:import
 - super() in, 237, 400
 - occurrence indicators, 163, 179, 538, 547, 669, 713.**
 - See also* «*»; «+»; «?»**
 - as attribute and, 179
 - SequenceType and, 505
 - OccurrenceIndicator construct, 668, 669**
 - odd-numbered nodes, 241**
 - ODF specification, 4**
 - omit-xml-declaration, 421, 422, 447, 448, 935**
 - in XHTML output method, 940
 - XML declarations and, 421, 422, 447, 935, 940
 - in XML output method, 935
 - in xsl:output, 421, 422
 - in xsl:result-document, 447, 448
 - one-or-more() function, 853–854**
 - avg() and, 854
 - empty sequence and, 853
 - max() and, 854
 - min() and, 854
 - static type checking and, 853
 - type errors and, 853
 - OOXML specification, 4**
 - OpenGIS schema, 557**
 - Opera, 68, 100**
 - opera example, 507–510**
 - operand types for value comparisons, 582–586**
 - operands**
 - atomization of. *See* atomization
 - conversion to strings, 586
 - as sequences, 588
 - operations, polymorphic, 186**
 - operator precedence, 537–538**
 - «[]» predicate operator and «/» path operator, 618
 - parentheses and, 542
 - production rules and, 523
 - operator priority. *See* operator precedence**
 - operators, 537–539. *See also* specific operators**
 - additive. *See* additive operators
 - arithmetic. *See* arithmetic operators
 - boolean. *See* boolean operators
 - difference. *See* «except» operator
 - first-order, 537–538
 - general comparison. *See* general comparisons
 - higher-order, 538–539
 - infix, 526, 627
 - intersection. *See* «intersect» operator
 - Kleene, 669
 - multiplicative, 200
 - node comparison. *See* node comparisons
 - numeric comparison, 200

operators (continued)

operators (continued)

- numeric equality, 200
- precedence of. *See* operator precedence
- predicate, 618
- prefix, 526
- priority of. *See* operator precedence
- pseudo-. *See* pseudo-operator
- reserved words and, 538
- set, 629. *See also* Venn operators
- tokens as, 538
- type, 538
- unary minus, 200
- union. *See* «union» operator
- value comparison. *See* value comparisons
- Venn. *See* Venn operators
- whitespace and, 538

optimization, 107. *See also* lazy evaluation; performance

- of joins
 - with keys, 644
 - with Saxon-SA, 644
- performance, 67, 350
- tail-call, 487, 966, 993
- types and, 186, 431

optimizers, 140, 177, 350, 351, 449, 497, 586, 618, 619, 643, 686

- trace() and, 896

optional digit placemaker, 789

optional tunnel parameters, 430

«or» expression, 186, 537, 552

- boolean() and, 722
- error semantics for, 1144

Oracle XSLT processor, 9, 957

- extension functions in Java, 957

order

- of appearance, 248, 330
- arbitrary, of expressions, 901–902
- ascending, 459
- of attributes, 64, 241, 323
 - deep-equal() and, 747
 - undefined, 316
- of data types, 584–586
- of declarations (in stylesheets), 105, 360, 486
- descending, 458, 459
- document. *See* document order
- of elements, 64
- of execution, 80, 323
 - assignment statements and, 986
- of groups, 328
- major-to-minor, 242, 324, 456
- of messages, 388
- of namespace nodes, 393
- numeric, 585
- population (grouping), 327, 330, 739
- of results of expressions, 901. *See also* unordered()
 - reverse, 324, 458. *See also* reverse()

- specified, comma operator and, 635
- of strings, 584
- undefined. *See* undefined order

ordinal numbers, 409, 783

- Italian
 - «^a», 407
 - «^o», 407

OrExpr construct, 525, 594

Oriental scripts, 827

Othello (Shakespeare play). *See also* Shakespeare plays

- interleaved markup, 995–999
- modes example, 247–251
- rule-based stylesheet example, 981–985
- tunnel parameters example, 433–437
- word count example, 19–21

outermost ancestors, 542

outermost element, 48

output definition, 423

output documents. *See* result documents

output methods, 1227. *See also* HTML output method; serialization; text output method; XHTML output method; XML output method

output properties (JAXP), 1181, 1182. *See also* serialization, attributes

OutputKeys (JAXP class), 1173–1174

outputs, 41

- hyperlink, 435
- numeric, 531
- post-process, 235

OutputURIResolver (Saxon), 449, 1197

output-version, 447, 448

overflow, 575, 576

- arithmetic, 719
- integer, 574
- numeric, 659, 661

overlay stylesheet modules, 320, 368

- XML specification case study, 1041–1044
 - diffspec.xml, 1003, 1041–1044
 - REC-xml.xml, 1003, 1044

overloading, of functions, 546

override attribute (xsl:function), 346–347

override/overriding, 94, 136, 268

- declarations, 364
- template rules, 237
- xsl:param, 431

oXygen development environment, 10, 12, 13, 1195

P

page footers, 293

page headers, 293

parallel hierarchies, interleaved markup and, 942

parallel markup, 995–996

parallel processing, 245, 309, 323, 330

- parameters, 68, 82, 1227. See also xsl:param**
 - for attribute values in named attribute sets, 268
 - call-by-value, 273
 - declaration of, 82
 - default value of, 244, 273, 427, 428–429, 487
 - definition/description, 1227
 - evaluation, 244
 - expected type of, 431
 - global. *See* global parameters
 - lazy evaluation, 138, 487
 - local. *See* local parameters
 - mandatory, 428
 - names, 428–430
 - scope of, 429
 - uniqueness of, 429
 - passing on, 243, 244
 - stylesheet. *See* stylesheet(s)
 - template. *See* template parameters
 - tunnel. *See* tunnel parameters
 - type declarations of, 161
 - type (required) of, 425, 427–428, 431, 487
 - values of
 - as document node, 306
 - xsl:with-param and, 517–518
 - xsl:apply-imports, 238
 - xsl:call-template, 272–273
- parameters (serialization), 929**
 - «method», 929
- parent axis, 609, 612, 621, 1227**
 - «..» as, 621, 623
 - definition/description, 1227
- parent nodes, 55, 623**
 - children relationship v., 55, 623
 - RelativePathPattern and, 693
- parentheses, 542–543**
 - «/» in, 624
 - «,» operator and, 525, 542
 - empty sequence and, 542
 - «if» expression and, 528
 - for negative numbers (accountancy), 301, 790
 - operator precedence and, 542
 - predicates and, 542
 - in regex, 234, 918
- parenthesized subexpressions (regex), 234, 918**
- ParenthesizedExpr construct, 542–543**
- parentless nodes**
 - attributes, 257
 - elements, 55, 87
 - patterns and, 683, 684, 688–689
 - PatternStep and, 695
- parsed entity references, 64**
 - external, 48, 87
- parsers**
 - MSXML3, 388, 440. *See also* MSXML processor
 - pull, 8
 - SAX, 8, 1164
 - XML, 8, 752
 - non-validating, 803, 805, 904
 - validating, 62, 752, 904
 - XML 1.1, 104, 440
 - XPath, 527
- parsing documents, 43, 752, 756–757**
- part explosion, 251**
- partial date types, 207**
 - xs:gDay, 207
 - xs:gMonth, 207
 - xs:gMonthDay, 207
 - xs:gYear, 207
 - xs:gYearMonth, 207
- partially ordered dates, 196, 585**
- particles (in schemas), 1227. See also complex types; element declarations**
- partitioning, 328**
- passing on parameters, 243, 244**
- path expressions, 39, 57, 83–84, 189, 322, 350, 599–632, 1227**
 - «//» and. *See* «//»
 - abbreviated. *See* abbreviated path expressions
 - abbreviated axis steps and, 599
 - absolute, 625. *See also* rooted path expressions
 - axes and, 599
 - axis steps and. *See* axis steps
 - binary «/» and. *See* «/»
 - changes in XPath 2.0, 601–602
 - changing focus with, 565
 - context item and, 543, 644
 - definition/description, 1227
 - distinguish simple mapping expressions from, 645
 - document order and, 644
 - element names in, 482
 - examples of, 600–601
 - «except» operator and, 599, 628
 - «for» expressions v., 644
 - simple mapping expressions v., 645
 - «intersect» operator and, 599, 628
 - nodes in tree and, 599
 - predicates in, 639
 - predicates in filter expressions v., 639–640
 - rooted. *See* rooted path expressions
 - SELECT statement v., 568, 632
 - sets of nodes and, 599
 - simple mapping. *See* simple mapping expressions
 - steps and, 83, 599
 - syntax, 526
 - «union» operator and, 599, 628
 - variable reference as first component in, 626
 - Venn operators and. *See* Venn operators
- PathExpr construct, 526, 625**
 - PathPattern in, 690

PathPattern construct

PathPattern construct, 690–692

- ID attribute and, 690, 691
- key value and, 690, 691
- kinds of, 691
- node kind and, 690
- in PathExpr, 690
- Pattern and, 690

Pattern construct, 690

- «|» and, 690
- PathPattern and, 690
- UnionExpr and, 690

pattern facet, 236, 665

- canonical form and, 667
- converting non-string values to derived types and, 667
- regular expressions and, 667, 829

PatternAxis construct, 695–696

- attribute axis and, 695
- child axis and, 695
- PatternStep and, 695–696

pattern-matching processing, 78. See also push processing technique

patterns (stylesheet design), 973–1000. See also design patterns

patterns (XSLT), 68, 214, 228, 377, 483, 681–708, 1227

- «/», 692
- algorithm for matching, 685
- axes in, 683
 - attribute, 683
 - child, 683
- changes in XSLT 2.0, 682–683
- chevrons and, 689
- circularity and, 378, 485, 683
- conflict resolution, 69, 73, 79–80, 686–688
- context and, 683
- current() and, 683, 736
- default priority of, 686–687, 691
- definition
 - formal, 683–685
 - informal, 685–686
- definition/description, 228, 1227
- element names in, 482
- in elements, 681. *See also* xsl:for-each-group; xsl:key; xsl:number; xsl:template
- expressions v., 681–682
- group-ending-with attribute, 330
- grouping with, 740
- group-starting-with attribute, 329–330, 340
- key(), 706–707
- last() function in, 685, 823
- match attribute, 484
- match nodes according to schema-defined type, 683
- match nodes by name, 696
- parentless nodes and, 683, 684, 688–689
- position() function and, 685

- predicates in, 683, 685
 - runtime errors and, 688
- QNames as, 536
- starting with «/», 689
- syntax of, 483, 689–707
- template rules and, 681, 686, 696
- union, 384, 400, 419, 686
- unprefixed names in
 - xpath-default-namespace and, 481–482
- in XProc, 682
- in xsl:for-each-group, 681
- in xsl:key, 681
 - key() and, 683
- in xsl:number, 681
- in xsl:template, 483, 484, 681

pattern-separator (xsl:decimal-format), 789

- picture string, 300

PatternStep construct, 694–704

- examples, 703–704
- KindTest and, 697–701
- NameTest and, 696–697
- node kind and, 694
- node name and, 694
- parentless nodes and, 695
- PatternAxis and, 695–696
- predicates and, 694, 696, 701–704
 - several, 703
- schema type and, 694

#PCDATA content, 142

PDF (Portable Document Format), 44

peer nodes, numbering of, 406

percent encoding (in URIs), 771, 772, 776, 811, 936, 938, 940

- encode-for-uri() and, 771–772
- escape-html-uri() and, 775–777
- escape-uri-attributes. *See also* escape-uri-attributes
 - in xsl:output, 422
 - in xsl:result-document, 447
- HTML/XHTML serialization and, 776, 937
- iri-to-uri() and, 811–812
- in xs:anyURI type, 194
- in xsl:result-document, 447

percent signs, iri-to-uri() and, 812

percent symbol (xsl:decimal-format), 300, 790

- picture string, 300

%HH encoding. See percent encoding

performance. See also lazy evaluation; optimization

- extension functions and, 956
- named modes and, 249–250
- optimization, 67, 350

period. See «.»

Perl, 187, 204, 232, 829. See also regular expressions

- regex and, 829, 915
- XML Schema and, 829, 915
- XPath within, 553

- per-mille (xsl:decimal-format), 300, 790**
 - picture string, 300
- phone book, UK, 462**
- PHP, 430, 1087**
- picture argument (format-date()), 781–783**
 - examples, 787–788
 - square brackets and, 781
- picture string (xsl:decimal-format), 300, 789–791**
 - negative subpicture, 790
 - subpicture, 789
 - symbols in, 300
- pieces (regex construct), 916**
- pipelines, 67, 86, 448, 510**
 - in JAXP, 1190–1193
 - micro, 88, 402
 - multiphase transformations and, 981–982
 - processor, 956. *See also* XProc
 - XProc, 992
- PITarget, 49, 51, 443**
- PITest, 616, 670, 698**
- place marker, 405, 406–407**
- plus operator. *See* additive operators**
- P.M. or A.M. components, 782**
- poem examples**
 - attribute set for numbering in, 270–271
 - formatting of, 35–38
 - import precedence of template rules in, 365–368
 - multiple result documents, 451–452, 767–769
 - numbering lines of, 417–418
 - validation of result document, 171–174
 - in XHTML, 171–173
 - validation of source document, 167–170
- polymorphic operations, 186, 221**
- population (grouping), 327, 330, 739**
- population order (grouping), 327, 330, 739**
- Portable Document Format. *See* PDF**
- portable extensions, 970–971**
- portable stylesheets, 127–141, 970–971**
 - conditional compilation, 127–128
 - extensibility, 134–141
 - extension functions, 134–138
 - extension instructions, 138–141
 - function-available() and, 970
 - system-property() and, 970
 - version compatibility, 128–134
 - backward-compatibility mode, 131–134
 - forward-compatibility mode, 130–131
 - whitespace nodes in, 146–147. *See also* whitespace
- position. *See also* context position**
 - of context node, 241
 - of node in hierarchy, 407
 - of node in source document, 406
 - in result document, 416
- positional predicates, 607, 620, 702**
 - «//» and, 627
 - attribute axis and, 702
 - last() and, 702
 - position() and, 702
- positional qualifiers, 242**
- position() function, 85, 146, 212, 241, 323, 333, 411, 416, 459, 564, 854–856. *See also* last(); xsl:number**
 - «//» operator and, 703
 - binary «/» and, 605
 - context position and, 564, 608, 620, 639, 854–856
 - current position and displaying, 856
 - testing, 856
 - first or last item in list and, 856
 - numbering and, 856
 - numeric predicates and, 619, 638
 - patterns and, 685
 - positional predicate and, 702
 - predicates and, 856
 - xsl:number v., 856
 - xsl:preserve-space and, 441
- positive group (character group), 919**
 - character range and, 919
- positive infinity**
 - «INF», 199, 576, 660
 - xsl:decimal-format, 300, 302, 790
- positive numbers, formatting, 789**
- positive zero, 198, 576**
- Post Schema Validation Infoset (PSVI), 46, 65**
- post-process output, 235**
- precedence. *See* import precedence; operator precedence**
- preceding axis, 609, 612, 1227**
- preceding siblings, 242, 323, 324, 406**
- preceding-sibling axis, 83, 609, 612, 1227**
- precision argument, 872**
- precision of numeric calculations, 531**
 - «div» and, 575
 - indefinite, 575
 - xs:decimal and, 575
- predefined entity references, 444**
- predicate calculus, 641**
- Predicate construct, 618, 638**
- predicate logic, XPath syntax, 39**
- predicate operator «[]», 618**
 - path operator «/» and, 618
- predicate«[last()]», 639**
- PredicateList construct, 618**
- predicates, 83, 85, 599, 617–621, 1227**
 - added to PrimaryExpr, 540
 - in axis steps, 542, 552, 606, 617
 - predicates in filter expressions v., 639–640
 - boolean. *See* boolean predicates
 - context item and, 543, 565, 620, 639

predicates (continued)

predicates (continued)

- context node and, 608
- context position and, 608, 620, 639
- context size and, 608, 620, 639, 821
- definition/description, 1227
- effective boolean value and, 619, 702
- in filter expressions, 552, 638
 - predicates in axis steps v., 639–640
- focus for evaluation of, 620
- with nodes, atomization and, 638
- numeric. *See* numeric predicates
- parentheses and, 542
- patterns in. *See* patterns
- PatternStep and, 694, 701–704
- PatternStep and several, 703
- positional, 607, 620, 702
- position() and, 856
- self, 639
- square brackets ([]) in, 84, 638, 701, 856
- zero or more, 620

Prefix, 534

prefix(es). *See also* **unprefixed names**

- absence of, in QNames, 533, 766, 866
- of attribute name, 256
- changing, 310–311
- choosing, 262, 308
- of element name, 308
- from expanded QNames, 857–858
- [fn:], 545, 710. *See also* *specific unprefixed function names*
- lexical QName, 50
- namespace. *See* namespace prefixes
- not known at compile time, 263
- reserved, 256
- in result document, 397–398
 - namespace fixup and, 397
- xml, 256
 - mapped to namespace URI
 - http://www.w3.org/XML/1998/namespace, 190, 311, 345

prefix operators, 526

prefix-from-QName() function, 857–858. *See also*

- local-name-from-QName();**
- namespace-uri-from-QName()**

- DTD-based validation and, 857
- namespace fixup and, 857
- xs:NCName and, 857

presentation, customizing of, 368

presentation modifier, 782

preserve (validation =), 173, 178, 260, 289, 293, 297, 305

- xs:anyType and, 313

preserving whitespace, 143, 214, 439–442

- primary expressions, 526, 539–551.** *See also* **context item expressions; function calls; numeric literals; parenthesized subexpressions; string literals; variable references**
 - AxisStep v., 540
- primary weight, 461**
- PrimaryExpr construct, 539**
 - predicates added to, 540
- primitive types, 189–190, 1227–1228.** *See also* **built-in atomic types**
 - converting between, 654–664
 - list of, 190, 1228
- principal node kind, 613, 614, 1228**
 - definition/description, 1228
 - name() and, 836
 - NameTests and, 613, 614, 696
- principal source document, 246, 477, 626**
 - global variables and, 506, 626
- principal stylesheet module, 90–91, 480**
- priority(ies), 239, 243, 362, 1228**
 - conflicts and, 440
 - default. *See* default priorities
 - definition/description, 1228
 - explicit, 401
 - import precedence and, 239, 243, 362, 400, 441, 484, 486
 - NameTests and, 440
 - operator. *See* operator precedence
 - template rule, 79
 - KindTests and, 701
- priority attribute, 79, 484, 485–486, 489**
- private use area, Unicode, 287, 943**
- procedural stylesheet, navigational stylesheet v., 979**
- procedure call (analogy with template call), 271**
- processContents, 314**
- processing**
 - attribute nodes, 241
 - instruction nodes, 49
 - string value, 51
 - parallel, 245, 309, 323, 330
 - pattern-matching, 78
 - pull. *See* pull processing
 - push. *See* push processing technique
 - recursion and. *See* recursive/recursion
 - redundant, 313
 - rule-based, 487–488. *See also* template rules
 - sequential, 245
- processing environment, 456, 890**
 - system-property() and, 890–893
- processing instructions, 49, 51, 289, 1228.** *See also* **<?xml-stylesheet?> processing instruction; xsl:processing-instruction**
 - deep-equal() and, 746
 - definition/description, 1228
 - escaping of special characters in, 445

1282

- in HTML, 444
 - terminator for, 444
 - name of, 443
 - nodes, 1228
 - selecting, 616
 - string value of, 51, 443
 - processing-instruction(), 616, 671, 698, 700**
 - keyword, 616
 - KindTest, 613, 616, 700
 - processing model, XSLT, 41–88**
 - error handling, 80
 - expressions, 82–84
 - dynamic context, 84, 85
 - static context, 84
 - multiple inputs/outputs, 44–45
 - output formats, 43–44
 - parameters, 82
 - system overview, 41–45
 - temporary documents, 85–88
 - transformation process, 67–80
 - overview, 41–45
 - variables, 81–82
 - XDM tree model, 8, 42–43, 45–67
 - processor, incremental stylesheet, 988**
 - processors, XSLT**
 - AltovaXML. *See* AltovaXML processor
 - basic. *See* basic XSLT processors
 - 4XSLT, 136, 957
 - Gestalt. *See* Gestalt processor
 - Gregor, 10
 - IBM, 10
 - Intel, 9, 10
 - Java, binding in, 959–961
 - libxslt, 10
 - MSXML. *See* MSXML processor
 - .NET framework, 9
 - non-schema-aware. *See* basic XSLT processors
 - Oracle, 9, 957
 - product name of, 891
 - running within Web servers, 430
 - Saxon. *See* Saxon
 - schema-aware. *See* schema-aware XSLT processors
 - System.Xml.Xsl. *See* System.Xml.Xsl processor
 - Xalan-J. *See* Xalan-J processor
 - xsl:is-schema-aware and, 478, 891
 - xsl:product-name and, 891
 - xsl:product-version and, 891
 - xsl:supports-backwards-compatibility and, 892
 - xsl:supports-namespace-axis and, 892
 - xsl:supports-serialization and, 892
 - XSLT 1.0. *See* XSLT 1.0
 - XSLT 2.0, 10
 - XSLTC, 10
 - xsl:vendor and, 478
 - xsl:vendor-url and, 891
 - xt, 319, 320, 550
 - product name, of XSLT processor, 891**
 - production rules, 522–524**
 - notation, 522–523
 - operator precedence and, 523
 - type checking and, 523
 - grammatical construction of XPath expression, 396**
 - programming languages. *See* functional programming languages; languages (programming)**
 - progressive rendering, 29, 35**
 - promotion, 1228**
 - numeric. *See* numeric type promotion
 - of xs:anyURI to xs:string, 81, 427, 506, 548, 713, 838
 - “A Proposal for XSL,” 28–29**
 - proprietary extensions, 319. *See also* vendor-specific extensions**
 - pseudo-attributes, <?xml-stylesheet?>, 100–101, 444**
 - pseudo-operator (“//”), 602, 626–628**
 - “,” and, 544
 - examples, 627–628
 - positional predicates and, 627
 - rooted path expressions and, 625
 - PSVI. *See* Post Schema Validation Infoset**
 - public identifiers, 446, 902–904. *See also* doctype-public; unparsed-entity-public-id()**
 - doctype-public and, 933, 938
 - pull parsers, 8**
 - pull processing, 78, 246, 324, 490**
 - punctuation tokens (xsl:number), 407**
 - pure functions, 34, 566, 987**
 - push processing technique, 74–78, 246, 324, 490**
 - controlling the sequence of processing, 77
 - example, 74–76
 - selecting nodes explicitly, 77–78
 - Python, 136, 957**
 - extension functions in, 957
 - 4XSLT processor and, 957
- ## Q
- QName() function, 858–859. *See also* xs:QName**
 - node-name() and, 859
 - QNames, 135, 440, 534, 1228. *See also* xs:QName**
 - as attribute and, 536
 - in axis step, 536
 - dynamic construction, 535
 - as element names, 536
 - error codes as, 775
 - expanded. *See* expanded QNames
 - function names as, 545
 - functions on
 - (list), 711
 - lexical. *See* lexical QNames
 - as literal result element name, 536
 - namespace declarations and, 535
 - namespace prefixes and, 262, 535

QNames (continued)

QNames (continued)

- as patterns, 536
 - resolving, 864–866
 - in SequenceType, 536
 - stylesheet generation and, 261
 - type attribute and, 536
 - as type names, 536
 - unprefixed, 536, 766, 866
 - default namespace and, 536, 766, 866
 - xpath-default-namespace and, 536
 - xsl:preserve-space and, 536
 - xsl:strip-space and, 536
 - in XSLT stylesheets, 535
- QName-valued attributes, 261, 393**
- QName-valued elements, 393**
- quantification, «=» operator and, 649**
- quantified expression, 646**
- QuantifiedExpr construct, 525, 646**
- keywords and, 525
- quantifiers (logical). See also «every» expression;**
- «some» expression**
 - existential, 646
 - universal, 646
- quantifiers (regex), 232, 916–917**
- «*?», 916–917
 - «*», 916–917
 - «+?», 916–917
 - «+», 916–917
 - «?», 916–917
 - curly brackets and, 917
 - greedy, 232, 863, 894, 917
 - non-greedy, 232, 863, 917
- question mark «?»**
- «cast as» and, 655
 - «castable as» and, 655
 - occurrence indicator, 163, 179, 547, 669
 - empty sequence and, 713
 - optional argument v. argument with, 713
 - quantifier (regex), 917–918. *See also* quantifiers (regex)
 - xsi:nil and, 673
- quotation marks, 533. See also double quotes;**
- single quotes**
 - apostrophes and, 533
 - delimiting, 528
 - French, 522

R

Raggett, Dave, 947

- random number generator, 955**
- range variables, 641, 1228**
 - definition/description, 1228
 - in «every» expression, 646
 - in «for» expression, 644

- joins with multiple, 647
 - in «some» expression, 646
- RangeExpr construct, 525, 636**
- value comparisons and, 582
- ranges, numeric, 633, 636–637. See also «to» operator**
- ratio between two durations, 577**
- read-only language, XPath as, 188, 526**
- Recommendations (W3C)**
- defined, 30
 - serialization specification, 44, 929
 - standards vs., 30
 - XML Information Set, 62
 - XML Namespaces 1.1, 24, 58, 474, 534
 - <?xml-stylesheet?> processing instruction, 99, 444
 - XPath 2.0, 82
 - XPath expressions, 8
 - XSLT 1.0, 28, 30, 479
 - XSLT 2.0, 5, 32–33, 479
- recoverable errors, 1140**
- recursive/recursion, 992–993. See also computational stylesheets**
- deep copy, 290
 - divide-and-conquer, 993
 - functions, 347, 350–351, 862, 992
 - head-tail, 275, 993
 - iteration v., 992–993
 - level=multiple and, 419
 - processing
 - list of values, 274–275
 - sequence of nodes, 275–278
 - sequence of strings, 278–280
 - speed/efficiency, 278
 - summary, 999–1000
 - tail-recursive functions, 351, 353, 862, 966
 - templates, 274–280, 350, 487
 - xsl:call-template and, 274–280
- REC-xml.xsl overlay stylesheet, 1003, 1044**
- redundant namespace nodes, 394**
- redundant processing, 313**
- regex. See regular expressions**
- regex construct, 916**
- regex-group() function, 234, 860–861. See also xsl:analyze-string**
- xsl:analyze-string and, 860–861
- regular expressions (regex), 21, 204, 351, 915–927, 1228. See also matches() function; replace() function; tokenize() function; xsl:analyze-string**
- anchored, 236, 829, 863
 - backslash in, 916
 - case-insensitive comparison in, 828, 925
 - chevrons and, 915
 - comments in, «x» flag and, 926
 - constructs in, 233
 - curly brackets and, 231, 915, 916
 - definition/description, 1228

- Digit as, 530
- digits in, 233, 921
- escaping special characters in, 232, 915, 920–921, 927
- flags. *See* flags
- matches zero-length string, 863
- matches() and, 828–830
- metacharacters. *See* metacharacters
- parentheses in, 234, 918
- parenthesized subexpressions, 234, 918
- pattern facet and, 667, 829
- Perl and, 829, 915
- quantifiers. *See* quantifiers
- replace() and, 862–864
- square brackets in, 919–920
- syntax, 915–927
 - brief summary, 232–234
 - disallowed constructs, 927
 - XML Schema and, 829, 915
- tokenize() and, 894–895
- whitespace in, 926
- words matching in, 921
- in xsl:analyze-string, 123, 232–234
- xsl:matching-substring and, 386
- xsl:non-matching-substring and, 402
- zero-length strings and, 231, 863
- relational databases, 8, 67. *See also* SQL**
- relational joins, 643**
- relationships, encapsulation of, 348**
- relative reference, 867–868**
 - of (""), 868
 - relative URI v., 867
- relative URIs, 54, 100, 193, 293, 357**
 - definition/description, 756
 - doc() and, 751, 755
 - in href attribute, 100, 359, 373
 - as links between output documents, 449
 - as namespace URI, 55
 - relative reference v., 867
 - resolving, 720, 755
- RelativePathExpr construct, 602, 625**
- RelativePathPattern construct, 693–694**
 - ancestor node and, 693
 - attribute-or-top axis and, 689
 - child-or-top axis and, 689
 - parent node and, 693
 - steps and, 694
- Relax NG, 25**
- remainder from integer division, 572, 575. *See also* «mod» operator**
- removal**
 - of characters, 897–898
 - of duplicate groups, 328
 - of duplicate items, 333
 - of duplicate nodes, 378, 544, 603
 - of duplicate values, 749–750
 - of whitespace-only text nodes, 143, 146, 441–442, 465, 493
- remove() function, 212, 861–862**
 - tail of sequence and, 861
- rendering elements example, 239**
- reordering sequences in document order, 602**
- repeating groups, of adjacent elements, 338–340**
- replace() function, 204, 234, 862–864**
 - backslash in, 863
 - ends-with() and, 774
 - regex syntax and, 915–927
 - translate() v., 864, 898
 - xsl:analyze-string v., 864
- replacement characters, 897. *See also* translate() function**
- replacing all occurrences of substrings, 862**
- replacing whitespace, 143, 214**
- reporting errors, xsl:message and, 388**
- required attribute (xsl:param), 238, 244, 273, 428, 487**
- required cardinality, 505, 547**
- required item type, 505, 547**
 - atomic type as, 506
 - as numeric type, 506
- required tunnel parameters, 430**
- required type**
 - of arguments, 547
 - of parameters, 425, 427–428, 487
 - standard conversion rules and, 505
- requirements, for XSLT 2.0, 32**
- reserved for XQuery, typeswitch expression, 546, 669, 680**
- reserved prefix, 256**
- reserved words, 40, 528, 615**
 - operators and, 538
- resolve-QName() function, 864–866. *See also* in-scope-prefixes(); QName()**
- resolvers for URIs, 369, 755–756**
 - CollectionURIResolver (Saxon), 727, 1197
 - OutputURIResolver (Saxon), 449, 1197
 - URIResolver (JAXP), 138, 567, 751, 755, 965, 1186
 - XmlResolver (.NET), 138, 567, 751, 755, 965
- resolve-uri() function, 867–869**
 - absolute URIs and, 867–869
 - base-uri() and, 720
 - doc() and, 750, 751, 868–869
 - document() and, 763
 - static-base-uri() and, 868
- resolving QNames, 864–866**
- resolving relative URIs, 720, 755**
- Result (JAXP interface), 1174**
- result documents. *See also* xsl:result-document**
 - base URIs for, 68
 - integrity constraints, 174

result documents (continued)

result documents (continued)

- multiple, 44–45, 445, 450
 - XSLT 1.0 processors and, 319–320
- with namespace declaration, 262
- position in, 416
- prefixes in, 397–398
- relative URIs as links between, 449
- in transformation process, 41–42
- validation of, 170–174
 - XHTML, 170
- well-formed, 174
- xsi:noNamespaceSchemaLocation in, 174
- xsi:schemaLocation in, 174
- xsl:output and, 420

result sequence, 253

result trees, 1228

- definition/description, 1228
- DOM, 420
- final, 74, 448
 - URIs for identifying, 449
- fragments, 87, 320, 454. *See also* document nodes
- HTML, 423
- JAXP and, 421
- with same URI, 450
- sequential building of, 257
- serialization of. *See* serialization
- validation
 - with validation/type attributes, 932
- XHTML, 423–424
- xsl:apply-templates and, 245
- xsl:result-document and, 445

returning

- any values, 453
- atomic values, 453, 455
- references to existing nodes, 453, 455

reusable components, 363. *See also* stylesheet(s)

revalidating copied subtree, 298

reverse axes, 607, 1227, 1228

- definition/description, 1228
- reverse document order and, 608, 620, 1228

reverse document order, reverse axes and, 608, 620, 1228

reverse order

- of integers, 637
- of original sequence, 324, 458, 869

reverse() function, 324, 637, 869. *See also* unordered()

RFC 2045, 208

RFC 2396, 867

- relative reference of zero-length string, 868

RFC 3986, 771, 811, 1230

- algorithm for URI resolution in, 867
- relative reference of zero-length string, 868

RFC 3987, 811

RFC 4790, 460

Rich Text Format. *See* RTF

Roman numerals, 409, 410, 782

root expressions («/»), 623–625

- abbreviated axis steps v., 624
- document node and, 623
- errors and, 624
- no context item and, 624
- root() v., 870
- type of, 624

root nodes, 47, 624, 1228

- definition/description, 1228
- element nodes as, 688
- other than document node, 624, 625, 688, 870
- root() and, 870

rooted path expressions, 542–543, 625–626

- «//» and, 625
- «/» as, 542–543, 625–626
 - «[]» and, 618
- errors and, 625
- examples of, 626
- no context item and, 625

root() function, 624, 870. *See also* root expressions

- «/» expression v., 870

round() function, 200, 870–872

- xs:integer() constructor function and, 872

round-half-to-even() function, 200, 212, 531, 576, 872–873

- format-number() and, 873
- unwanted digits and, 660

rounding errors, 575, 576

rounding numbers, 723, 870–872. *See also* ceiling();

- round()
 - floor() and, 780
 - round-half-to-even() and, 872–873
 - significant digits and, 790

rounding rules, 790, 893

RSS, NameTests and, 697

RTF (Rich Text Format), 44

Rubik's cube, 42. *See also* transformation process, XSLT

rule-based processing, 487–488. *See also* template rules

rule-based structure, of XSLT, 35–38, 41

rule-based stylesheets (design pattern), 74, 246, 980–985, 1047, 1113. *See also* XML specification (case study)

- CSS and, 980

rules

- rounding, 790, 873
- standard conversion. *See* standard conversion rules

running examples. *See* examples, running

running total, 900

runtime

- attribute name decided at, 255, 260, 264–266
- element name decided at, 308

- errors
 - atomization and, 499
 - predicates in patterns and, 688
 - schemas/compile time schemas, 752
 - sort key as runtime parameter, 836
 - stylesheets included conditionally at, 374
 - template name evaluated at, 272
 - type errors at, 506
 - XPath expression constructed at, 541
- Russian language, 409**
- S**
- «Σ» (sigma), 827
 - «s» flag, 926
 - «.» and, 917, 926
 - dot-all mode and, 917, 926
 - s9api interface (Saxon), 1199, 1202–1203**
 - Safari, 100**
 - sales report example, 977–980**
 - same keys**
 - multiple definitions for, 383–384
 - nodes with, 378
 - same name**
 - attributes with, 257, 267
 - functions with, 546, 559
 - local variables with, 502
 - user-defined functions and vendor-supplied functions with, 346
 - xsl:key declarations with, 383–384
 - same namespace, functions in, 349**
 - same nodes, keys for, 383**
 - same schema, different versions of, 370**
 - same URI, result trees with, 450**
 - Sarissa, 68**
 - «satisfies» expression, 646
 - SAX**
 - ContentHandler, 44, 1174, 1180
 - JAXP support for, 1164–1166
 - parsers, 8, 1164
 - Saxon, 9, 10, 30, 1195–1214. See also examples, running**
 - arbitrary length integers and, 212
 - B, 10, 1195
 - Saxon-SA v., 644
 - calling Java methods from, 1206
 - collation URIs, 565, 1207–1208
 - collations, 561, 731, 1207–1208
 - collections, 727, 1197
 - command line interface, 1196–1199
 - control over doc() invoking validation, 671, 1198
 - DOM and, 1200, 1205
 - early versions, 30
 - extensibility, 959, 962, 1205–1208
 - extension attributes, 1209
 - extension functions, 959, 962, 1209–1210
 - extension instructions, 139, 140, 475–476, 1209
 - extension-element-prefixes and, 476–477
 - extensions, 1208–1210
 - functions from, 135
 - graphical user interfaces, 10
 - input-type-annotations and, 478
 - on Java, 1195
 - from command line, 14–15, 103, 1196–1199
 - extension functions and, 957
 - JAXP transformation API and, 1163, 1199–1202, 1200
 - Kernow and, 10, 13–14, 1195
 - linked tree, 1205
 - on .NET, 15, 1195, 1203–1205
 - binding in, 962
 - C# and, 957, 1093, 1203–1204
 - extensibility, 962
 - NodeInfo interface, 1205
 - OutputURIResolver, 449, 1197
 - s9api interface, 1199, 1202–1203
 - SA, 10, 1195
 - optimization of joins with, 644
 - Saxon-B v., 644
 - schema validation API, 1201
 - validating result document on, 172–173
 - validating source document on, 169
 - SQL extension, 111, 475, 476
 - substring matching using collation, 731–732
 - tiny tree, 1205
 - trace(), 896
 - TransformerFactory and, 1200–1201
 - tree models, 1205
 - unordered() and, 901, 902
 - validation request, 166, 671, 1197
 - version 6, 8, 319, 767, 809, 1195
 - on Xalan-J, 957
 - extension functions and, 957
 - XPath expressions in, 1201–1202
 - saxon:deep-equal() function, 748**
 - saxon:doctype, 477**
 - saxon:evaluate() function, 135, 541, 546, 1210–1214**
 - saxon:group extension element, 111**
 - Saxonica, 10, 30**
 - saxon:while, 139**
 - SAXParser (JAXP class), 1165–1166**
 - SAXParserFactory (JAXP class), 1164–1165**
 - SAXResult (JAXP class), 1174–1175**
 - SAXSource (JAXP class), 1175–1176**
 - SAXTransformerFactory (JAXP class), 1176–1177**
 - Scalable Vector Graphics. See SVG**
 - schema(s), 1228. See also XML Schema(s)**
 - adding
 - transition from XSLT 1.0 to 2.0, 1145

schema(s) (continued)

schema(s) (continued)

- cached, 369
- compiled, 370
- compile time/runtime, 752
- components, 557, 1228. *See also* attribute
 - declarations; element declarations; type definitions
- consistency rules for, 370
- declarations in. *See* declarations
- definition/description, 1228
- definitions, 369
- different versions of same, 370
- importing, 180–181, 368–372. *See also* `xsl:import-schema`
 - with no target namespace, 482
 - user-defined derived types in, 664, 900
 - XQuery and, 369
- information, 557
- inline, 175, 368, 369
- locating, 314
- no-namespace, 372
- OpenGIS, 557
- stylesheets and, 151–183
- type definitions in, 653. *See also* schema types
- types imported from, 793
- XHTML, 172
- for XSLT 2.0 stylesheets, 352
 - examples, 674–677
 - global attribute declarations and, 676

schema namespace, 475

schema processor, 143, 297

- ID values and, 379
- validation of XML document with, 752

schema types, 653, 1228, 1230. *See also* atomic types; complex types; list types; simple types; union types

- annotated with*, 653, 742
- as complex types, 653, 742
 - with simple content, 742
- definition/description, 1228
- element content and, 742
- mixed content and, 742
- PatternStep and, 694
- sequence types v., 653–654
- as simple types, 653, 742
- in static context
 - `type-available()` and, 899–901

schema validation, 52, 65, 107, 165, 262, 303, 311

- atomization and, 742
- `doc()` and, 752
- `document()` and, 757
- namespace fixup and, 393
- `nilled()` and, 842
- strings converted to other types and, 663
- `xs:untypedAtomic` and, 670

schema-attribute (N), 371, 558, 622

SchemaAttributeTest construct, 674, 698

schema-aware XSLT processors, 151, 190, 352, 471

- as attribute and, 161
- built-in types, 190, 192, 557, 670
- complex element-only content and, 499
- extension of type hierarchy in, 163–164
 - built-in atomic types and, 163, 164
 - user-defined types in, 163, 164
- NameTest and, 696
- non-schema-aware and
 - stylesheets for, 370
- type/validation attributes and, 258
- validation/annotation
 - of attributes, 255, 289
 - of elements, 307, 312–315
- XPath, 54
- `xsl:import-schema` in, 368

schema-defined types, 219, 427, 601

- of attribute nodes, 622
- `attribute()` and, 617
- AttributeTest and, 616
- `element()` and, 617
- ElementTest and, 616
- KindTests and, 613, 698
- type checking based on, 353

schema-element

- global element declaration and, 674, 699

schema-element(N), 371, 558, 674

- KindTest, 700
- type annotation and, 674

SchemaElementTest construct, 674, 697

schema-location attribute (`xsl:import-schema`), 181, 369

Schematron, 396

Scheme (programming language), 28

schemes

- error-handling, 775
- numbering, 784

scientific notation. *See* exponential notation

scope

- of namespace declarations, 58
- namespace prefixes in, 808–809
- of parameter names, 429
- of variables, 81, 268
 - global, 501
 - local, 70, 501, 502

scoped variables, dynamically, 429

script

- in HTML output page, 137
- JavaScript and, 957
- in stylesheet, 137

script element, 936

secondary documents, 510, 692, 761

secondary weight, 461

- seconds component**
 - in `format-time()`, 782
 - fractional, 782, 783
 - from `xs:dateTime`, 873–874
 - from `xs:duration`, 874–875
 - from `xs:time`, 873–874
- seconds-from-dateTime() function, 873–874**
- seconds-from-duration() function, 874–875**
- seconds-from-time() function, 873–874**
- section headings, 78**
- section numbers, 405**
- select attribute, 230, 412**
 - cross-reference construction with, 412
 - sequence constructor *v.*, 457
 - string value of namespace node from, 391
 - value of variable and, 503
 - `xsl:apply-templates` and, 241
 - `xsl:for-each` and, 322–323
 - `xsl:param` and, 428
 - `xsl:value-of` and, 496
 - `xsl:variable` and, 501
- Select Case construct, 283**
- SELECT statement (SQL), 8, 568**
 - expressions *v.*, 568, 632
- selecting processing instructions, 616**
- self axis, 609, 612, 836, 1228**
 - «@» and, 622
 - definition/description, 1228
 - `name()` and, 836
- self predicate, 639**
- semantics**
 - existential, 590–592. *See also* general comparisons
 - shortcut, 595–596. *See also* boolean operators
- separator attribute, 254, 258, 495, 496**
- separator character «.» , 528**
- separators, 285**
 - between adjacent strings, 258, 879
 - default, 258, 497
 - spaces as, 266, 304, 309
 - `tokenize()` and, 894, 895
- sequence(s), 187–189, 351, 1228–1229. *See also***
 - atomic values; node(s)**
 - atomic values and, 187–188, 322, 351, 633
 - collating. *See* collations
 - concatenation, 261, 634. *See also* comma operator
 - definition/description, 1228–1229
 - empty. *See* empty sequence
 - «for» expression and, 640. *See also* «for» expression functions on, 72
 - (list), 711
 - identity and, 633
 - immutability of, 633
 - items in, 187–188, 189
 - iterate over, 324
 - locating items in, 807–808
 - manipulation of. *See* sequence expressions
 - mapping, 641–642
 - maps* the, 641
 - maximum/minimum values in, 830–832
 - multiple input
 - «for» expression and, 643–644
 - and navigating trees, 189
 - nesting and, 188, 633
 - nodes in, 187–188, 633
 - booleans compared to, 589
 - number of items in, 733
 - numbering. *See* numbering sequences
 - operands as, 588
 - reverse, of original order, 324, 458, 869
 - sets of nodes and, 629
 - sorting
 - into document order, 57
 - `xsl:perform-sort` and, 437, 438
 - of text nodes, type error and, 498
 - type labels and, 634
 - value as, 187
 - XPath operators on, 189
- sequence constructors, 69, 70–72, 89, 90, 108, 149, 228, 1229**
 - contained, 503
 - definition/description, 228, 675, 1229
 - element nodes within. *See* instructions
 - evaluation of, 309
 - instructions in, 766
 - nested, 73–74
 - result of function and, 346
 - scope of local variables and, 70
 - `select attribute v.`, 457
 - templates as, 90
 - text nodes in, 72
 - `xsl:document` in, 303
 - in `xsl:sort`, 457
 - `xsl:text` in, 492
 - `xsl:value-of` and, 495
 - in `xsl:variable`, 501
- sequence expressions, 633–651**
 - comma operator. *See* comma operator
 - «every» expression. *See* «every» expression
 - filter expressions. *See* filter expressions
 - «for» expression. *See* «for» expression
 - simple mapping expressions. *See* simple mapping expressions
 - «some» expression. *See* «some» expression
 - «to» operator. *See* «to» operator
- sequence numbers, 241. *See also* numbering sequences**
 - backward-compatibility mode and, 405
 - composite, 407, 419
 - hierarchic, 418
 - `xsl:number` and, 405–407

sequence type descriptors

sequence type descriptors, 221, 228

- as attribute and, 162, 164, 668
- defined, 654
- examples of, 162
- global element declarations and, 673–674
- local elements/attributes in schema and, 677
- matching atomic values, 669–670
- matching elements/attributes, 672–677
- matching nodes, 670–672
- within XPath expressions, 165, 668–677
- XQuery and, 668
- xs:anyType and, 670

sequence types, 653–654, 1229, 1230. See also item types

- cardinality in, 653
- definition/description, 1229
- instance of, 653
- «instance of» operator and, 668
- item types in, 653
- schema types v., 653–654
- «treat as» operator and, 668

sequence-constructor, 496

SequenceType construct, 228, 371, 426, 668. See also

- sequence type descriptors**
- atomic type and, 371
- occurrence indicators and, 505
- QNames in, 536
- syntax. *See also* sequence type descriptors
 - element names in, 482
 - type names in, 482
- type of variable and, 501
- union types in, 372
- xsl:template and, 484
- xsl:variable and, 501

sequential building of result trees, 257

sequential processing, 245

serialization, 43, 44, 929–949, 1229. See also

- xsl:output**
- attributes
 - also known as parameters, 932
 - as attribute value templates, 448
 - HTML output method, 938–939
 - as JAXP output properties, 1181–1182
 - on Saxon command line, 1199
 - text output method, 940
 - XML output method, 932–936
 - in xsl:output, 421–422
 - in xsl:result-document, 446–447
- backward-compatibility mode and, 1142
- character attribute replacement during, 425
- character maps and, 280, 282, 941. *See also* character maps
- controlling, 67, 450
- definition/description, 1229
- escape-uri-attributes and, 776, 936
- extension attributes and, 1225

- href attribute and, 446, 448–449
- method. *See* output methods
- namespace declarations from, 391, 473
- of newline characters, 939, 940
- parameters, 929
 - also known as attributes, 932
- property, 776
- W3C specification, 44, 929

server-side transformation, <?xml-stylesheet?>, 100, 319

servlets, 430

- (family tree case study)
- generating HTML from, 1087–1092
- installing/configuring, 1090–1092

set operators, 629. See also Venn operators

sets, attribute. See attribute sets

sets of atomic values, 631

sets of nodes

- combining, 599, 628–631
- sequences and, 629

sets, XDM data model and, 629

SGML (Standard Generalized Markup Language), 27, 903

- character maps and, 942

Shakespeare plays. See also Othello

- formatting of, 335, 336, 337, 338, 433
- grouping consecutive elements by name, 336–338
- handling repeating groups of adjacent elements, 338–340
- text of all, 335, 433

shallow copy, 287

shortcut semantics, boolean operators and, 595–596

siblings, 55. See also following siblings; preceding siblings

- numbering of, 406, 411–412

side effects, 34–35

- calling extension functions within loop, 967
- extension functions and, 138, 244, 549, 595, 955, 956
- in functional languages, 449
- functions with, 549–550
- immutability and, 633
- pure functions and, 34, 566, 987
- tunnel parameters and, 431
- without, 29, 34–35, 69–70, 350, 526
 - expression evaluation, 526
 - functions, 34
- of xsl:result-document, 449

sigma («Σ»), 827

signatures

- digital, 396
- function, 220
 - in static context, 547, 559

significant digits, 790

- rounding numbers with, 790

1290

- significant whitespace, 142**
- simple content, 154–155, 441**
 - complex types with, 154–155
 - schema types as, 742
- simple mapping expressions, 599, 601, 644–645**
 - «/» operator for, 599, 601, 644
 - atomic values and, 644–645
 - distinguish path expressions from, 645
 - path expressions v. «for» expressions v., 645
- simple type definitions, 52, 152–153, 163**
 - KindTest for attribute nodes and, 700
 - top-level, 371
- simple types, 52, 152–153, 163, 191–192, 371, 1229.**
 - See also atomic types; list types; union types**
 - as atomic types, 653
 - atomic types v., 191–192
 - definition/description, 1229
 - global, 259
 - as list types, 653, 670
 - schema types as, 653, 742
 - as union types, 653, 670
- simplified stylesheet modules, 89, 125–127, 263, 1229**
 - definition/description, 1229
 - syntax, 358, 374, 464, 466, 509
 - xsl:version and, 125
- single character escapes, 920, 921**
- single level. See level attribute**
- single quotes, 532**
 - string literals and, 532
- single-level grouping (example), 331–333**
- single-precision floating point, 190, 208, 576. See also xs:float**
- SingleType construct, 655**
- size**
 - of integer, 574
 - of table, calculation of, 724
- size, context. See context size**
- SMDL. See Standard Music Description Language (SMDL)**
- smileys, 529**
- SOAP envelope, 24, 120, 291, 297**
- soccer example, 294–296**
 - multiphase transformation, 511–513
- Software AG, 30**
- «some» expression, 39, 353, 540, 552, 646–650**
 - «=» operator and, 649
 - boolean() and, 722
 - context item and, 646
 - empty sequence and, 647
 - errors in, 649–650
 - dynamic, 649
 - focus within, 565
 - range variables in, 646
 - semantics of, 647–648
- sort key(s), 324, 328. See also xsl:sort**
 - conditional expression as, 462
 - dynamic, 462–463
 - equal, 324
 - major-to-minor order, 242, 324, 456
 - meanings for, 457
 - supplied as runtime parameter, 836
 - xsl:sort and, 455
- sort key components, 457, 462**
- sort key specification, 457**
- sort key values, 457**
- sorted sequence, 457, 459**
- sorting, 242. See also collations; xsl:perform-sort; xsl:sort**
 - collations and, 106, 459
 - distinct values, XQuery and, 750
 - groups, 330–331
 - grouping keys and, 330–331
 - items within each group, 331
 - language-dependent, 134
 - process, 457–458
 - initial sequence in, 457
 - rules for, 458
 - sorted sequence in, 457
 - on result of calculation, 463
 - sequences
 - into document order, 57, 241, 602
 - with xsl:perform-sort, 437, 438
 - strings, 84
 - XPath and, 562
 - xsl:apply-templates and, 199, 241
 - xsl:for-each and, 199, 323–324
- Source (JAXP interface), 1177**
- source documents, 41, 68, 1229**
 - annotation of, 165
 - copying to result, 293, 296
 - with few changes, 290
 - definition of, 1229
 - multiple, 44–45, 326, 482, 510
 - namespace nodes copied from, 474
 - position of node in, 406
 - principal, 246, 477, 506, 626
 - in transformation process, 41–42
 - type annotations on, 467
 - as untyped, 477
 - validating, 165–170
 - on AltovaXML processor, 169, 1218
 - example, 167–170
 - whitespace in, 141
- SourceLocator (JAXP interface), 1177–1179**
- space(s)**
 - ignorable
 - substring-after() and, 886
 - substring-before() and, 888
 - matching, in regex, 921, 924
 - as separators, 266, 304, 309
 - as whitespace characters, 440
 - «X» flag and, 926–927

space characters, nonbreaking

space characters, nonbreaking, 38, 142, 148, 281, 367, 732, 880, 938

« », 64, 281, 367, 938
character maps and, 943
normalization of, 880

space, stack, 487

Spanish collating rules, 732

Spanish language, 731, 732, 781

special characters, 425

converted
to character reference, 493
to entity reference, 493
escaping. *See* escaping special characters
jEdit and, 941
regex syntax and, 915

special-purpose rules, 239

special-purpose stylesheet module, 97

specification

HTML 4.0, 775, 776
JDK 1.1, 788, 790
ODF, 4
OOXML, 4
serialization (W3C Recommendation), 44, 929
stylesheet modules, 90
XHTML, 481
XML, 534
case study, 1001–1048
XML Schema, 369
XPath, 522

Sperberg-McQueen, Michael, 27

splitting

large XML file into smaller XML files, 445
lines into words, 279

SQL. *See also* relational databases; **SELECT statement**

extension (Saxon), 111, 475, 476
null value in, 182. *See also* nullability
empty sequence and, 586
XSLT vs., 8–9

sql:query (Saxon extension), 111, 475, 476

square brackets []

for Digit symbol, 530
doubling of, 781
hyphens allowed with, 920
in picture argument (format-date(), format-dateTime(), format-time()), 781
in predicates, 84, 638, 701, 856
in regex, 919–920

square root function, 346, 347

extension function example, 959–961
math:sqrt(), 346

stable attribute (xsl:sort), 242, 324, 456, 457, 458

stable functions, 726

stack frame, 487

stack space, 487

standalone attribute, 421, 422, 447, 448, 935

XML output method and, 935

standalone property, 67

standard conversion rules, 244, 346, 427, 504

backward-compatibility mode and, 505, 714
core function library and, 713–714

standard function library. *See* core function library

standard function namespace, 135, 482, 545, 556, 560, 655, 793

Standard Generalized Markup Language. *See* SGML

Standard Music Description Language (SMDL), 6

standard type conversions. *See* standard conversion rules

standards, 30. *See also* Recommendations

standoff markup, 995–996

start tags, 49, 1229

starting instants of dates

simultaneous, 583
xs:date minus xs:date and, 580

starting of numbering sequences

at one, 409
at value other than one, 412

starts-with() function, 204, 875–876. *See also*

contains(); ends-with(); matches(); string-length()
substring matching with collation, 731–732

states (U.S.) example, 284

static context, 84, 345, 545, 553, 554–563, 1229.

See also dynamic context

backward-compatibility mode, 544–545
base URI, 562, 868, 876
collations, 561–562
default namespaces, 556
definition/description, 1229
dynamic context v., 554
function-available() and, 793
in-scope functions, 559–560
in-scope namespaces, 555–556
in-scope schema definitions, 557–558
in-scope variables, 558–559
schema types in
type-available() and, 899–901
signatures of functions in, 547, 559
statically known documents/collections,
563–564

static errors, 80, 1229

static namespace context, 534

static type checking. *See* static typing

static typing, 165, 186, 187, 221–224, 559, 563, 1229

exactly-one() and, 777
one-or-more() and, 853
«treat as» operator and, 679
XPath and, 563, 680
zero-or-one() and, 912–913

statically known documents/collections, 563–564

- static-base-uri() function, 876–877. *See also* base-uri(); doc(); document(); resolve-uri()**
 - base URIs and, 562, 868, 876–877
 - resolve-uri() and, 868
- step expressions, 57**
- StepExpr construct, 602**
- steps, 83, 599, 1229. *See also* axis steps; path expressions**
 - RelativePathPattern and, 694
- sticky modes, 243**
 - not, 242
- StreamSource (JAXP class), 359, 1179–1180**
- strict (validation=), 173, 259, 297, 305, 371**
 - element declaration and, 314
 - unnamed type definition and, 314
- Strine types, 207**
- string(s), 1229. *See also* format string; xs:string**
 - as atomic values, 189, 190
 - comparing, 84, 724. *See also* codepoint-equal()
 - backward-compatibility mode and, 1141
 - default collation and, 588
 - incompatibility and, 588
 - in Java, 460
 - conversion of operands to, 586
 - conversion of, to booleans, 552
 - converting atomic values to, 878
 - definition/description, 1229
 - external file in form of, 904
 - first character of, 884–885
 - length, 880–881. *See also* zero-length strings
 - normalization, 849
 - numbers converted to, 531
 - ordering of, 584
 - replacement of. *See* replace()
 - returned from functions, 497–498
 - separator between adjacent, 258, 879
 - sorting, 84
 - starts with another string, 875. *See also* starts-with()
 - substrings in. *See* substring(s)
 - zero-length. *See* zero-length strings
- string functions**
 - empty sequence and, 711
 - list of, 710
- string literals, 203, 528, 532–534**
 - carriage return within, 532
 - double quotes and, 532
 - newline characters in, 532
 - single quotes and, 532
 - tab within, 532
- string mode (regex evaluation), 926**
- String Quartet examples, 488–491**
- string types, 549**
 - derived, 193, 213–215
- string value, 46, 47, 51–52, 1229**
 - attribute node, 51
 - characters in, 63, 884
 - comment node, 51, 285
 - definition/description, 1229
 - document node, 51
 - element node, 51, 742
 - namespace node, 51
 - of new attribute node, 258
 - of nodes, 46, 47, 51–52, 496
 - defined, 877–878
 - of processing instruction, 51, 443
 - text node, 51
 - typed value, type annotation, and, 54, 742
- string-delimiter character, 533**
- string() function, 52, 143, 200, 877–878. *See also* boolean(); number()**
 - backward-compatibility mode and, 877
- string-join() function, 204, 545, 879–880. *See also* concat()**
 - attribute value templates v., 880
 - concat() v., 880
 - normalize-unicode() and, 849
 - xsl:value-of v., 880
- string-length() function, 63, 204, 880–881. *See also* normalize-unicode(); substring()**
 - normalize-unicode() and, 880–881
 - zero-length strings and, 881
- StringLiteral symbol, 532**
 - whitespace within, 532
- string-to-codepoints() function, 642, 881–882. *See also* codepoints-to-string()**
- strip (validation=), 173, 178, 297, 305**
 - xs:untyped and, 314
- stripped spaces, MSXML processor and, 567**
- stripping. *See also* removal**
 - type annotations, 477
 - whitespace-only text nodes, 143, 146, 441–442, 465, 493, 757
- strong collations, 378, 461, 462, 562, 584**
- strong conversions, 81–82. *See also* conversions**
- strong typing, 187. *See also* static typing**
- style element, 936**
- stylesheet(s), 1229. *See also* portable stylesheets**
 - base URI of, 720, 756, 905
 - changes, in XSLT 2.0, 90
 - comments in, 287
 - computational, 985–1000, 1114. *See also* knight's tour (case study)
 - customization of, 429
 - Othello example, 433–437
 - debugging, 431
 - comments and, 286
 - declarations in. *See* declarations
 - definition/description, 228, 1229
 - design patterns, 973–1000
 - DocBook, 433

stylesheet(s) (continued)

stylesheet(s) (continued)

- document () and analysis of, 758–759
- document-formatting, 1047. *See also* rule-based stylesheets
- embedded. *See* embedded stylesheets
- examples
 - funcproto.xsl, 1046–1047
 - xmlspec.xsl, 1047
 - xsl-query.xsl, 1047
 - xsltdiff.xsl, 1046
 - xslt.xsl, 1045–1046
- for family tree case study, 1073–1085
- fill-in-the-blanks, 38, 973–976
 - as function, 986
- functions. *See* stylesheet functions; xsl:function
- generating
 - namespace aliasing and, 121
 - through other stylesheets, 393, 396
 - QNames and, 261
 - xsl:namespace-alias and, 394
- importing, 237, 357–368
- included conditionally at runtime, 374
- layer, 360
- level, 238
- look-up table in, 761–763
- meta-, 393, 396
- modular structure, 90–98
- modules, 45, 88, 89, 90, 228, 357, 1229
 - definition/description, 1229
 - general-purpose, 97
 - locating, 357–359
 - media-type of, 470
 - official specification, 90
 - overlay. *See* overlay stylesheet modules
 - principal, 90–91, 480
 - special-purpose, 97
 - xsl:include and, 372
- multiple xsl:import-schema in, 370
- navigational, 976–980, 1114. *See also* family tree (case study)
- parameters, 68, 82, 138, 425, 426, 427, 430–431.
See also xsl:param
 - environment variables and, 427
 - extension functions v., 430
 - in form of DOM Document object, 430
 - in form of XPath expression, 430
- program, 90, 94
- QNames in, 535
- rule-based, 74, 246, 980–985, 1047, 1113. *See also* XML specification (case study)
- running in browser, 430, 463, 957, 1154–1157
- for schema-aware and non-schema-aware processors, 370

- schemas and, 151–183. *See also* XML Schema(s)
- script in, 137
 - script in HTML output page v., 137
- simplified. *See* simplified stylesheet modules
- structure, 89–149
- terminating execution of, 386
- top-level elements in, 90, 104
- transformation process and, 41–42
- tree, 360, 361
- type declarations in, 161
- in WD-xsl, 467
- whitespace in, 146–147. *See also* whitespace XML Schemas and, 151–183
- xml:space in, 147

stylesheet functions, 1229

- context item and, 350, 736
- current group and, 739
- definition/description, 1229
- function-available() and, 792, 793
- named templates as wrapper for, 349
- named templates v., 347, 349–350
- as wrapper for named templates, 349
- with xsl:function, 106, 135, 136, 271, 344

Stylus Studio development environment, 10, 12, 13, 16–17, 1195

subexpressions (regex), 918

- parentheses and, 234, 918

sublanguage, XPath as. *See* embedded languages

subpicture, 789. *See also* picture string

- negative, 790

subroutine call, 271

subsequence() function, 882–883

substitutability, 664

substitution groups, 158–161, 683, 696

- content model restriction/extension and, 701
- global element declarations and, 674
- KindTests and, 701
- match by type in, 981

substitution, of characters, 897–898

substitution, textual, 541

substring(s)

- captured, 860
- comparing, 730–732. *See also* contains()
- matching with collation, 731–732, 876
- minimal matching, 886, 888
- replacing all occurrences of, 862
- tokenize() and, 894–895

substring-after() function, 204, 885–887

substring-before() function, 204, 887–888

substring() function, 63, 204, 883–885

- as conditional expression, 885
- «if» expression v., 885

subtract-dates-yielding-yearMonthDuration (dropped from final specification), 348

- subtracting**
 - dates, 348
 - durations, 577, 579
 - subtraction (character group), 919**
 - subtypes, 701**
 - sum() function, 200, 889–890. See also avg(); count()**
 - «+» operator and, 889
 - empty sequence and, 634, 889
 - NaN and, 889
 - xs:dayTimeDuration and, 889
 - xs:yearMonthDuration and, 889
 - super() function (object-oriented programming), 237, 400. See also xsl:apply-imports**
 - supplied value, 505**
 - surrogate pair, 63, 203, 460, 725, 880, 882, 884, 898**
 - SVG (Scalable Vector Graphics), 291, 531**
 - trigonometric functions and, 955
 - Sweden, collations in, 459**
 - Swedish language, 561, 848**
 - switch construct, 282, 283**
 - syntactic ambiguities, of «/», 623–624**
 - syntactic rules, XPath, 521, 524**
 - comments and, 524
 - lexical rules v., 524
 - whitespace and, 524
 - syntax**
 - arithmetic operators, 571–572
 - axis steps, 606–607
 - path expressions, 39, 83–84, 526
 - of patterns, 483, 689–707
 - regex, 232–234, 915–927
 - brief summary, 232–234
 - disallowed constructs, 927
 - matches() and, 915
 - replace() and, 915
 - special characters and, 915
 - tokenize() and, 915
 - XML Schema and, 829, 915
 - xsl:analyze-string and, 915
 - SequenceType
 - element names in, 482
 - type names in, 482
 - simplified stylesheet modules, 358, 374, 464, 466, 509
 - XML, XSLT and, 33–34
 - XPath expressions, 39–40, 82–83, 522, 1117–1122
 - conventional programming expressions, 39
 - notation, 522–523
 - predicate logic, 39
 - system clock, 738**
 - system identifier, 447, 902. See also base URIs; doctype-system; unparsed-entity-uri()**
 - in DOCTYPE declaration, 422, 447, 933, 938
 - system properties, 891–892**
 - xsl:is-schema-aware, 478, 891
 - xsl:product-name, 891
 - xsl:product-version, 891
 - xsl:supports-backwards-compatibility, 892
 - xsl:supports-namespace-axis, 892
 - xsl:supports-serialization, 892
 - XSLT 2.0
 - (list), 891–892
 - xsl:vendor, 478
 - xsl:vendor-url, 891
 - system-defined functions, 793–794**
 - system-property() function, 128, 131, 134, 321, 467, 890–893**
 - function-available() and, 892
 - portable stylesheets and, 970
 - system properties (list) and, 891–892
 - use-when and, 370, 478, 769, 892
 - xsl:fallback and, 892
 - xsl:import-schema and, 769
 - System.Xml namespace (.NET), 138**
 - framework classes, 567
 - XmlResolver and, 138, 567
 - System.Xml.Xsl processor (Microsoft .NET), 9, 957, 1158–1161**
 - binding in MSXML and, 957
 - binding, using external objects in, 961–962
 - extensibility in, 961
 - extension functions in .NET languages, 957
-
- T**
 - tab (whitespace character), 440, 528**
 - within string literal, 532
 - table(s)**
 - calculating size of, 724. *See also* ceiling() function
 - formatting. *See* XML specification (case study)
 - look-up, 107, 321, 510, 753
 - in stylesheet, 761–763
 - numbering sequences for, 415
 - table (HTML) examples, 116**
 - arranging data in, 343–344
 - attribute set for, 269
 - generate empty table cells in, 344
 - table of contents, 78, 247, 490**
 - tags, 49, 1229**
 - tail-call optimization, 487, 966, 993**
 - tail-recursive function, 351, 353, 862, 966**
 - Tamil digits, 408**
 - target namespace, 180, 370**
 - importing schemas with no, 482
 - tax rates. See look-up data**
 - technical documents, sequence numbers for, 418**
 - Technical Report #15 (Unicode), 848**
 - Technical Report #21 (Unicode), 827**

template(s)

template(s), 90, 1229. *See also* attribute value templates; named templates; sequence

constructors; xsl:template

current. *See* current template rule

definition/description, 1229

evaluating, 486–487

as functions, 988

identity, 290

initial, 68, 1225

definition/description, 1225

in knight's tour example, 1102–1104

local parameters to, 425, 427

recursion and, 274–280

term, 74

template call, recursive, 487

template parameters, 82, 425, 426, 427, 431

xsl:apply-imports and, 238, 431

xsl:apply-templates and, 243, 431

xsl:call-template and, 272, 431

xsl:next-match and, 400, 431

xsl:with-param and, 431, 517–518

template rules, 35–38, 73, 487–488, 1230

built-in. *See* built-in template rules

choosing, 242–245, 485–486

import precedence, 486

contents of, 69–70

current. *See* current template rule

default, for comment nodes, 700

definition of, 228, 1232

empty, 261

example, 488–490

import precedence of, 79, 94, 365–368, 484

last specified, 362

match attribute and, 487

modes and, 242, 402

overriding, 237

patterns and, 681, 686, 696

priority, 79

KindTests and, 701

sequence constructors, 70–72, 241

nested, 73–74

for xsl:apply-templates, 362

XSLT transformation process and, 68–74

Templates (JAXP interface), 1180

TemplatesHandler (JAXP interface), 1180–1181

templating languages, 973

temporary documents, 72, 85–88, 244, 274, 428, 503, 510

examples, 86

node selection in, 378

unnecessary, 514

validation of, 174–176

xsl:document and, 175

well-formed XML document and, 87

xpath-default-namespace and, 482

temporary trees, 428, 1230

copying data to/from, 293

definition/description, 1230

document() and, 756

example, 320–321

final result trees v., 1225

ID attributes in, 379

namespace fixup and, 932

validation on

document-level, 303

element-level, 305

xsl:value-of and, 497

xsl:variable and, 619

terminate attribute (xsl:message), 386, 387

terminating execution, of stylesheets, 386

terminator for processing instruction, in HTML, 444

terms/glossary, 1221–1231

tertiary weight, 461

test expression (xsl:when), 283

testing. *See also* KindTests; NameTests; NodeTests

name of nodes, 836

nodes before/after in document order, 57, 593, 594

for nonempty node sequence, 276

for vendor extensions, 769–770. *See also*

element-available() function

text(), 671, 698, 700

KindTest, 613

text declarations, 935

encoding attribute of, 933

omit-xml-declaration and, 935

XML declaration v., 931

«text» keyword, 616

text nodes, 49, 1230

adjacent, 143, 258, 304, 309, 496

construction of, 495. *See also* xsl:text; xsl:value-of

definition of, 1230

in sequence constructors, 72

sequence of, type error and, 498

string value, 51

stripped, 143

text() and, 671

whitespace-only. *See* whitespace-only text nodes

zero-length, 258, 304, 496

text output method, 44, 423, 940. *See also* serialization

encoding attribute and, 940

line endings and, 940

TextTest construct, 616, 670, 698

textual substitution, 541

text/xsl, 100

Thai digits, 408

third-party extensions. *See* vendor-specific extensions

third-party function libraries, 546, 955. *See also* EXSLT

library; functx; FXSL

thousands, grouping separator and, 791

«³» (superscript three), 849

- times.** *See also* **dates/times**; **xs:dateTime**; **xs:time**
 - arithmetic on, 574, 577–581
 - current. *See* **current-dateTime()**; **current-time()**
 - formatting, 738–739, 781. *See also*
 - format-dateTime()**; **format-time()**
 - ISO format, 785
- time-sequence data**, 335
- timezone(s).** *See also* **dates**; **xs:dateTime**
 - absence of, 567
 - dates with, 583
 - definition/description, 806
 - implicit, 567, 580, 583, 585, 718, 738, 806–807
 - current-date()** and, 583
 - names for, 787
 - part of value space, 195, 196, 197
 - xs:dayTimeDuration** added to **xs:dateTime** in same, 578
- timezone component**, 782, 893
 - from **xs:date**, 893
 - from **xs:dateTime**, 893
 - from **xs:time**, 893
- timezone-from-date() function**, 893
- timezone-from-dateTime() function**, 893
- timezone-from-time() function**, 893
- tiny tree (in Saxon)**, 1205
- title attribute (<?xml-stylesheet?>), 101**
- title case**, 783
- «to» operator**, 538, 633, 636–637
 - examples with, 637
 - lazy evaluation, 637
 - type error and, 636
- tokenize() function**, 21, 204, 234, 278, 894–895
 - regex and, 894–895
 - regex syntax and, 915–927
 - separators and, 894–895
 - substrings and, 894–895
 - xs:analyze-string** and, 895
- tokenizing**, 528–529
- tokens**, 528. *See also* **name(s)**; **number(s)**; **operators**
 - lexical, 524
 - operators as, 538
 - xs:number**
 - formatting, 407
 - punctuation, 407
- Tomcat**, 1090, 1091, 1092
- top-level attribute declarations.** *See* **global attribute declarations**
- top-level declarations.** *See* **declarations**
- top-level element declarations.** *See* **global element declarations**
- top-level elements**, 90, 1230
 - as declarations, 90, 104, 360. *See also* **declarations**
 - definition of, 1230
 - as user-defined elements, 104, 107–108
 - msxsl:script**, 134
 - xmlns:import** as, 357
 - xmlns:include** as, 372
- top-level schema declaration**, 314
- top-level type definitions**, 370, 557
 - complex, 557
 - simple, 371, 557
- totals, numeric values/durations**, 889–890. *See also* **sum()**
- trace() function**, 550, 896–897. *See also* **xs:message**
 - diagnostic output and, 896–897
 - optimizer and, 896
 - Saxon, 896
 - in XQuery, 897
 - xs:message v.**, 897
- traditional numbering**, 783
- trailing whitespace**, 142, 493
 - normalize-space()** and, 493, 845–846
- trailing zeros**, 300, 790
- transformation process, XSLT**, 67–80
 - error handling, 80
 - final result tree of, 74
 - invoking transformation, 67–68
 - overview, 41–45
 - result documents in, 41–42
 - source documents in, 41–42
 - stylesheets in, 41–42
 - template rules, 68–74
- transformations.** *See also* **multiphase transformations**
 - phases of, 86
 - in two passes, 991
 - XML-to-XML, 290, 396
- Transformer (JAXP class)**, 1181–1182
- TransformerConfigurationExpression (JAXP class)**, 1182
- TransformerException (JAXP class)**, 1182–1183
- TransformerFactory (JAXP class)**, 1183–1185
 - Saxon and, 1200–1201
- TransformerFactoryConfigurationError (JAXP class)**, 1185
- TransformerHandler (JAXP class)**, 1185–1186
- transformNode() (MSXML)**, 941
- transition aid, backward compatibility mode as**, 554. *See also* **backward-compatibility mode**
- transitivity**, 582, 586
 - distinct-values()** and, 587
 - «eq» operator and, 587
 - xs:for-each-group** and, 587
- translate() function**, 63, 410, 897–898. *See also* **contains()**; **matches()**; **replace()**; **substring()**
 - matches()** v., 898
 - replace()** v., 864, 898
- TrAX**, 68, 1163, 1164. *See also* **JAXP**
- «treat as» operator**, 165, 371, 538, 624, 654, 668, 678–680
 - as assertion, 678
 - dynamic typing and, 679

«treat as» operator (continued)

«treat as» operator (continued)

- sequence types and, 668
- static typing and, 679

TreatExpr construct, 526, 678

tree(s), 42, 1230. *See also* result trees; temporary trees

- definition of, 1230
- final result, 74, 448
- import, 238
- linked (Saxon), 1205
- with no document node, 624
- non-document, 626
- with root other than document node, 624, 625, 688
- sequences and navigation of, 189
- stylesheet, 360, 361
- tiny (Saxon), 1205

tree model (family tree case study), 1050. *See also* family tree

tree model, XDM. *See* XDM tree model

tree models (in Saxon), 1205

tree-valued variables, 244

tree-walking, 250

trigonometrical calculations, 531, 955

trivial documents, 514

true value, booleans and, 189, 191, 194, 195

true() function, 899

- boolean constant and, 779, 899

try/catch mechanism, 80, 752, 774

tunnel parameters, 244, 272, 425, 429–430, 431, 1230

- definition of, 1230
- example, 433–437
- global variables v., 431
- local variables v., 431
- modes and, 431
- optional, 430
- required, 430
- side effects and, 431
- «tunnel = »yes''', 272
- xsl:with-param and, 429, 518

tuples, 647

Turkish dotless I, 827, 910

24:00:00 time value, 743

«²» (superscript two), 849

type(s), 38–39, 185–224. *See also* type systems

- abstract, 163
- anonymous, 164, 313, 673
- atomic. *See* atomic types
- built-in. *See* built-in types
- checking. *See* dynamic typing; static typing
- complex. *See* complex types
- concrete, 163
- conversions. *See* type conversions
- default namespace for. *See* default namespace
- definition/description, 185–186
- derived. *See* derived types

- derived by restriction, 664

- duration. *See* xs:dayTimeDuration; xs:duration; xs:yearMonthDuration

- element-only content, 53, 142, 156–157

- error detection and, 53, 186, 221

- errors. *See* type errors

- expressions and. *See* type expressions

- globally defined, 315

- imported from schema, 793

- item types v. schema types, 219

- least common, 831

- list, 53, 54, 153, 163, 164

- mapping Java to XPath, 961

- mixed content. *See* mixed content

- names. *See* type names

- noncomparable, 329

- numeric. *See* numeric types

- optimization and, 186, 431

- parameter. *See* parameters

- primitive. *See* primitive types

- purpose of, 186, 219, 221

- required. *See* required type

- of root expression, 624

- schema. *See* schema types

- schema-defined. *See* schema-defined types

- sequence. *See* sequence types

- simple. *See* simple types

- simple content, 154–155, 441

- union. *See* union types

- usefulness of, 186

- user-defined. *See* user-defined types

- of variables, 501

- XML Schemas, 38–39

- XPath 1.0, 186–187

- XPath 2.0, 186–187

- XSLT 1.0, 39

- XSLT 2.0, 39

type annotations, 53–54, 165, 289, 293, 312, 1230

- of attribute as xs:QName, 393

- copying, 297–298

- deep-equal() and, 747

- definition of, 1230

- of element nodes, 307

- global attribute declarations and, 673, 699

- global element declarations and, 673, 699

- invented name as, 313, 314

- KindTest and, 697

- schema-element (S) and, 674

- on source documents, 467

- string value, typed value, and, 54

- stripping, 477

- typed value of node and, 741

- xs:untyped, 741

- xs:untypedAtomic, 741

typeswitch expression (XQuery)

- type attribute, 100, 176, 254, 293, 303, 305, 307, 312, 446**
 - as attribute v., 178–179
 - default-validation and, 470
 - QNames and, 536
 - schema-aware XSLT processor and, 258
 - type names in, 482
 - validation of result document and, 173–174
 - [xsl:], 467
 - of xsl:attribute, 179, 370
 - of xsl:copy, 370
 - of xsl:copy-of, 370
 - of xsl:document, 370
 - of xsl:element, 370
 - of xsl:result-document, 173, 370
- type checking, 273, 277, 504**
 - backward-compatibility mode and, 549
 - based on schema-defined types, 353
 - dynamic. *See* dynamic typing
 - of function arguments, 1143
 - production rules and, 523
 - static. *See* static typing
 - for value comparisons, 586–587
 - empty sequence and, 586
- type conversions, 81, 377. *See also* casts; conversions**
 - strong, 81–82
 - weak, 81–82, 220
- type declarations, 303**
 - as attribute and, 161, 514
 - benefits of, 514
 - document. *See* DOCTYPE declaration
 - of parameters, 161
 - purposes of, 161
 - in stylesheet, 161
- type definitions, 259, 312, 1230. *See also* DTDs**
 - anonymous, 164, 313
 - description of, 1230
 - global. *See* global type definition
 - local, 315
 - in schemas, 653. *See also* schema types
 - simple, 52, 152–153, 163
 - top-level. *See* top-level type definitions
 - unnamed, 259, 314
- type errors, 241, 346, 505, 1230**
 - atomization of operands and, 573, 586
 - binary «/» and, 603
 - comparison of operand types and, 582
 - at compile time, 506
 - definition of, 1230
 - one-or-more() and, 853
 - at runtime, 506
 - sequence of text nodes, 498
 - «to» operator and, 636
 - Venn operators and, 630
- type expressions, 653–680**
 - converting atomic values, 654–668
 - converting between derived types, 664–668
 - converting between primitive types, 656–664
 - «instance of» operator. *See* «instance of» operator
 - sequence type descriptors. *See* sequence type descriptors
 - «treat as» operator. *See* «treat as» operator
- type hierarchy, 157–158, 162**
 - derived numeric types, 211
 - derived string types, 213
 - extension of, in schema-aware processor, 163–164
 - KindTests and, 701
 - in XML Schema, 157–158, 162, 217–218, 219
 - in XPath, 218–219
- type information, at compile time, 590**
- type labels, 664, 666**
 - converting between derived types and, 664
 - empty sequence and, 634
 - sequences and, 634
- type matching rules, 219–221, 981**
- type names**
 - in cast expression, 482
 - QNames as, 536
 - in SequenceType syntax, 482
 - in type attribute, 482
- type operators, 538**
- type promotion. *See* numeric type promotion**
- type safety, 777**
- type systems, 185–186, 224. *See also* dynamic typing; static typing; type(s)**
 - item types v. schema types in, 219
 - purpose of, 186, 219
 - type expressions and, 653. *See also* schema types; sequence types
 - type matching rules in, 219–221, 981
 - XML Schema, 38–39, 161–165, 217–218, 219, 506
 - XPath 1.0, 186–187, 224, 506
 - XPath 2.0, 186–187, 218–219, 224
- type-available() function, 899–901**
 - xsl:import-schema and, 900
 - [xsl:]use-when and, 900
- typed values (of nodes), 52–53, 81, 165, 258, 496, 506, 548**
 - atomization of operands and, 573, 586
 - data() and, 741
 - schema and, 741–742
 - string value, type annotation, and, 54
 - type annotation and, 741
- TypeName construct, 672, 698**
- typeswitch expression (XQuery), 546, 669, 680**

typing

typing

- dynamic. *See* dynamic typing
- static. *See* static typing
- strong, 187
- weak, 39, 187, 961
 - XPath 1.0 and, 555

U

UCA. *See* Unicode Collation Algorithm

UK phonebook, collations and, 462

UltraEdit, 1196

UML (Unified Model Language) diagram, 50

- class hierarchy for nodes, 50, 56–57
- sequences, items, atomic values, nodes, 188

unary minus operators, 200, 530, 572. *See also* «-»

unary plus operator, 573. *See also* «+»

UnaryExpr construct, 526, 572

undeclarations, namespace, 24, 58, 60, 104, 120, 391, 935

- default, 58
- in-scope-prefixes() and, 809
- namespace-uri-for-prefix() and, 840–841
- undeclare-prefixes and, 447, 935

undeclare-prefixes, 120, 291, 312, 421, 422, 448, 935

- inherit-namespaces and, 935
- namespace undeclarations and, 447, 935
- XML 1.1 output and, 447
- XML output method and, 935

undefined context item, 350, 544, 736

undefined order

- of attributes, 316
- of result of distinct-values(), 640

underflow, 575, 576

underscore, 534

Unicode. *See also* character encoding

- 4.0, 827
- case mappings, 828, 910. *See also* lower-case(); upper-case()
- locale-sensitive, 828
- characters, 63, 203, 281, 534
 - categories, 407
 - expressions as, 527
 - normalization of, 422, 447, 561, 731, 847–849
 - in range x80 to x9F, 937
 - XML 1.0, 527
 - XML 1.1, 527
- codepoints. *See* codepoints
- consortium, 460
- database, 233, 828, 922
- digits, 408, 782
- names for character ranges in. *See* character blocks
- private use area, 287, 943
- surrogate pair, 63, 203, 460, 725, 880, 882, 884, 898

Technical Report #15, 848

Technical Report #21, 827

Unicode codepoint collation, 377, 378, 460, 470, 561, 724, 731, 773, 1222

Unicode Collation Algorithm (UCA), 460, 461

Unified Model Language. *See* UML

Uniform Resource Identifiers. *See* URIs

Uniform Resource Locators. *See* URLs

«union» operator («|»), 57, 523, 526, 538, 542, 573, 601, 628–631

- path expressions and, 599, 628. *See also* path expressions
- Pattern and, 690
- XQuery and, 602

union patterns, 384, 400, 419, 686

union types, 53, 54, 153, 163, 164, 1230

- as content of node, 164
- definition of, 1230
- example, 372
- in SequenceType, 372
- simple types as, 653, 670
- xsi:type and, 182

UnionExpr construct, 526, 629

- Pattern and, 690

uniquely identified functions, 546, 559, 712

uniqueness, of parameter names, 429

United States (state example), 284

units of measure, as complex types, 213

universal quantifiers (logical), 646. *See also* «every» expression

unknown instruction, 318

unnamed mode. *See* default mode

unnamed type definition, 259

- strict and, 314

unnecessary/unwanted

- digits, 660. *See also* round-half-to-even() function
- namespace declarations, 471, 473–474, 475
- temporary documents, 514
- whitespace-only text nodes, 143, 146, 441–442, 465, 493, 757

unordered() function, 901–902

- joins and, 902
- Saxon and, 901, 902
- XML database and, 902

unparsed entities, 57, 902–904, 1230

- definition of, 1230

unparsed-entity-public-id() function, 57, 902–904

unparsed-entity-uri() function, 57, 902–904

unparsed-text() function, 904–910. *See also*

- xsl:analyze-string**
- comma-separated-values file, 906, 907–908
- envelope/payload and, 909
- HTML boilerplate generation, 909–910
- media-type and, 905

1300

- up-conversion, 906
- xsl:analyze-string and, 906
- unparsed-text-available() function, 904–910**
 - doc() and document() v., 905
 - up-conversion, 906
- unprefixed function names, 482, 710. See also specific unprefixed function names**
- unprefixed names**
 - attribute, 482, 837
 - in expressions, 481–482
 - not available as function names, 546
 - in patterns, 481–482
 - xpath-default-namespace and, 481–482
- unprefixed QNames, 536, 766, 866**
- unrecognized elements, 318**
- unreserved characters, 772**
- untyped atomic values, 81–82. See also xs:untypedAtomic**
 - general comparisons and, 588
- untyped source documents, 477**
- unwanted/unnecessary. See unnecessary/unwanted**
- up-conversion, 864, 906**
- updateable variables, without, 274, 350**
- updating, variables and, 506**
- upper-case() function, 204, 910**
 - case variants and, 926
- upper-case/lower-case letters, 461. See also case-order attribute**
- URI Reference, 867**
- URI resolution, 867–869**
 - algorithm for, 867
- URI resolvers, 369, 755–756**
 - CollectionURIResolver (Saxon), 727, 1197
 - OutputURIResolver (Saxon), 449, 1197
 - URIResolver (JAXP), 138, 567, 751, 755, 965, 1186
 - XmlResolver (.NET), 138, 567, 751, 755, 965
- URIs (Uniform Resource Identifiers), 23, 755–756. See also base URIs; collation URIs; collection URIs; namespace URIs; xs:anyURI**
 - absolute, 58, 100, 193, 357, 373, 755
 - as atomic values, 760
 - collations identified by, 460, 561
 - comparing, 724
 - definition/description, 755
 - dereferenced, 357, 373, 567, 755, 757
 - document-uri() and, 764
 - escape-uri-attributes. *See* escape-uri-attributes
 - escaping special characters in. *See* percent encoding of external entity, 54
 - final result trees identified by, 449
 - fragment identifier and, 756, 905
 - functions on
 - (list), 711
 - held in nodes, 757
 - IRIs converted to, 811–812
 - IRIs v., 812
 - mappings from, to documents, 568
 - not known at compile time, 263
 - relative. *See* relative URIs
 - resolving, 867–869. *See also* resolve-uri(); URI resolvers
 - result trees with same, 450
 - special characters in. *See* percent encoding
 - URLs v., 755
- URI-valued attributes, %HH encoding and, 422, 771, 772, 776, 811, 936, 938, 940**
- URLs (Uniform Resource Locators), 755**
 - ampersands in, 937
- URNs, 23, 903**
- use attribute (xsl:key), 376**
 - global variables and, 378
 - values of key and, 377
- use-attribute-sets attribute, 289, 352**
 - of literal result elements, 268
 - of xsl:copy, 268
 - of xsl:elements, 268, 309
- use-character-maps attribute, 281, 421, 422, 448. See also xsl:character-map**
 - character maps and, 447, 935, 941
 - HTML output method and, 939
 - XML output method and, 935
 - xsl:character-map and, 422, 447
- user-defined atomic types**
 - constructor functions for, 371, 546
 - converting to, 371
- user-defined attribute declarations, 369**
- user-defined element declarations, 369**
- user-defined error codes, 775**
- user-defined extensions, 318**
- user-defined functions, 72, 346, 709. See also extension functions**
 - in C#, 546
 - for calling, 546
 - in-scope functions as, 560
 - in Java, 546
 - in JavaScript, 546
 - same name for
 - vendor-supplied function, 346
- user-defined top-level elements, 104, 107–108**
- user-defined types, 163, 164, 369**
 - constructor functions and, 482
 - defined in imported schema, 900
 - derived, 664, 900
- user-written extension functions, 135, 953**
 - Java/JavaScript
 - XSLT 1.1 working draft and, 135, 953
- use-when attribute, 90, 98, 112, 127, 140–141, 321, 357, 358, 467, 478–479**
 - at compile time, 478
 - context, 478

use-when attribute (continued)

use-when attribute (continued)

- effective boolean value in, 478
- effect/usage, 478–479
- element-available() and, 140–141, 478, 766
- function-available() and, 478
- portable extensions and, 970
- system-property() function and, 370, 478, 769, 892
- [xsl:], 130, 319, 468, 792, 900
- xsl:import-schema and, 769
- XSLT 1.0 processors and, 767

UTC, 717

UTF-8 encoding, 63, 64, 906

- ASCII and, 365

UTF-16 encoding, 63

V

valid HTML, HTML output method and, 936

validating XML parsers, 62, 752, 904

validation, 113, 1230

- of attributes, 179–180, 255, 258–260. *See also* validation attribute
 - xs:NOTATION, 180
 - xs:QName, 180
 - context, 298
 - controlling, 450
 - of copied nodes, 297
 - default. *See* default-validation
 - definition/description of, 165, 1230
 - document-level, 293, 303, 305
 - DTD, 752, 755, 757, 805
 - element-level, of temporary tree, 305
 - of elements, 176–179, 289, 312–315
 - failure, 166, 259, 1230
 - IDs and, 803, 805
 - XML, 379
 - of literal result elements, 176
 - request, 166
 - AltovaXML, 166, 1218
 - Saxon, 166, 671, 1197
 - of result documents, 170–174, 932
 - example, 171–174
 - type attribute and, 173–174
 - validation attribute and, 173
 - XHTML, 170
 - schema. *See* schema validation
 - of source documents, 165–170
 - on AltovaXML processor, 169, 1218
 - of temporary documents, 174–176
 - by XML parsers, 62, 752. *See also* DTDs
- validation = lax, 173, 259, 297, 305, 371**
- validation = preserve, 173, 178, 260, 289, 293, 297, 305**
- xs:anyType and, 313
- validation = strict, 173, 259, 297, 305, 371**

validation = strip, 173, 178, 297, 305

- xs:untyped and, 314
- xs:untypedAtomic and, 314

validation attribute, 173, 176, 254, 293, 297, 303, 305, 307, 313, 446

- default-validation and, 470
- schema-aware XSLT processor and, 258
- validation of result document and, 173
- values of, 173
 - [xsl:], 467

validation request

- AltovaXML processor, 166, 1218
- Saxon, 166, 671, 1197

value(s). *See also* atomic values; sequence(s)

- default. *See* default values
- of parameters. *See* parameters
- removal of duplicate, 749–750
- returning any, 453
- as sequences, 187
- supplied, 505
- term, 187
- of variables. *See* variables

value comparisons, 581–587. *See also* «eq»; «ge»; «gt»;

- «le»; «lt»; «ne»
- atomization and, 586
- backward-compatibility rules and, 586
- examples, 587
- general comparison operators v., 582, 588
- operand types for, 582–586
 - RangeExpr and, 582
 - type checking for, 586–587
 - empty sequence and, 586
- XQuery and, 582
- xs:untypedAtomic values and, 587

value space, 195

- of dates, 786
- timezone part of, 195, 196, 197

ValueComp construct, 582

van der Vlist, Eric, 151

variable binding (XSLT), 1230

variable references, 540–542, 1230

- \$ sign and, 540
- circularity and, 683
- definition of, 1230
- as first component in path expressions, 626
- id() and, 704
- key() and, 704
- xsl:param and, 540
- xsl:variable and, 540

VariableReference construct, 540

variables, 72, 81–82, 506–510, 1230. *See also*

- xsl:variable
 - categories of, 507
 - context, 510
 - convenience, 507

- deferred evaluation of, 388
 - definition of, 1230
 - dynamically scoped, 429
 - environment, 427
 - global. *See* global variables
 - in-scope. *See* in-scope variables
 - intermediate results in, 510
 - literal string value assigned to, 504
 - local. *See* local variables
 - mathematical use of term, 989
 - names of, 501–502
 - \$ sign and, 528
 - namespace prefixes in, 541
 - in namespaces, 559
 - no updateable, 274, 350
 - range. *See* range variables
 - references to. *See* variable references
 - scoping rules, 81, 268
 - tree-valued, 244
 - global, 294
 - type of, 501
 - updating and, 506
 - usage, 506–510
 - values of, 306, 503–505
 - as attribute and, 503, 504
 - dynamic context and, 566
 - expressions *v.*, 604
 - select attribute and, 503
 - xsl:apply-templates and, 510
 - xsl:call-template and, 510
- VarName construct, 640, 646**
- VBScript**
- as dynamically typed language, 959
 - extension functions in, 134, 957
 - MSXML and, 957
 - extension functions in, 957, 958–959
- vendor portability example, 319–320**
- vendor-defined APIs, 427**
- vendor-defined attributes, 134**
- vendor-defined numbering sequences, 134**
- vendor-defined output methods, 423. *See also* serialization**
- vendor-specific extensions, 107, 134, 318, 319, 954–955. *See also* extension functions**
- allowed, 954–955
 - context/environment and, 954
 - function-available() and, 794
 - implementation-defined XSLT language behavior and, 954–955
 - testing for, 769–770
 - variance in, 953
- vendor-supplied functions, 709**
- for calling, 546
 - user-defined function name and, 346
- Venn operators, 599, 629. *See also* «except»; «intersect»; «union»**
- combining sets of nodes with, 628–631
 - examples of, 630
 - type errors and, 630
- version = “1.0,” 480**
- version = “1.1”, 291**
- version = “2.0,” setting of, 1142–1144**
- arithmetic operators and, 1143
 - comparison operators and, 1143
 - empty sequence and, 1143–1144
 - error semantics for «and» and «or», 1144
 - first node rule, 1142–1143
 - type checking of function arguments, 1143
- version attribute, 98, 318, 358, 374, 421, 422, 468, 479–480, 496, 935–936**
- effect/usage, 479–480
 - forward/backward compatibility modes, 479, 496
 - XSLT 1.0 processor, 479
 - XSLT 2.0 processor, 479
 - HTML version and, 939
 - XML version and, 935–936
 - XMLSpy and, 479
 - [xsl:], 318
- version compatibility, 128–134. *See also* backward-compatibility mode; forward-compatibility mode**
- versions (XSLT)**
- browsers and different, 318
 - effective, 318
 - element-available() and, 766–769
 - future, 318
 - testing, 321
- versions, of same schema, 370**
- vertical bar symbol. *See* «|»; «union» operator**
- visitor pattern, 250. *See also* patterns**
- VoiceXML, 7**

W

- W3C (World Wide Web Consortium), 21, 25. *See also* Recommendations**
- Walmsley, Priscilla, 151, 546**
- third-party function library, 546
- WD-xsl, 467**
- weak collations, 378, 461, 562, 584**
- weak conversions, 81–82, 220**
- weak typing, 39, 187, 961. *See also* dynamic typing**
- XPath 1.0 and, 555
- Web documents, doc() and, 751**
- Web servers, XSLT processors running within, 430**
- week numbers, ISO, 787**
- weeks of month, 782**
- weeks of year, 782**
- numbering, 785

weights

weights

- in non-Latin scripts, 461
- primary, 461
- secondary, 461
- tertiary, 461

well balanced, 48

well-balanced XML fragment, 48, 1230–1231

well-formed documents, 48, 174, 304, 305, 309, 671, 700, 929–930, 1231

- definition of, 1231
- external parsed entities v., 930–931
- not required by XDM, 304, 309
- prerequisite for validity, 174, 305
- rules for, 930
- temporary document and, 87
- well-balanced fragment v., 48
- XML output method and, 929–932

while loop, 274

whitespace, 72, 141–149, 241, 493–494, 1231. *See*

also line ending; newline; nonbreaking space; spaces

- «-» and need for, 538
- AltovaXML and, 1216
- characters, 63, 440. *See also* carriage return; linefeed; spaces; tab
- collapse, 143, 214
- definition of, 142, 845, 1231
- facet, 143, 214, 663, 665, 846
- insignificant, 142, 465
- leading, 142, 493
- lexical rules (XPath) and, 524
 - between lexical tokens, 524
 - matching with \s (regex), 921
 - normalization, 528, 663, 665
 - normalize-space() and, 493, 845–846
 - operators and, 538
 - preserving, 143, 214, 439–442
- in regex, 926
- replace, 143, 214
- significant, 142
- in source document, 141
- within StringLiteral, 532
- in stylesheets, 146–147
- syntactic rules (XPath) and, 524
- too little, 148
- too much, 147–148
- trailing, 142, 493
- «x» flag and, 231, 926

whitespace-only text nodes, 66, 72, 141, 146–147, 439–440

- deep-equal() and, 747
- handling with MSXML, 567, 1158
- stripping unwanted, 143, 146, 441–442, 465, 493, 757
- xsl:preserve-space and, 757

- xsl:strip-space and, 439–440, 465, 493, 757
- xsl:text and, 492–493

whitespace-separated lists, 113

- of IDs, 804
- of values
 - normalize-space() and, 846

width modifier, 782, 783

Wildcard construct, 614

words

- matching in regex, 921
- as numbering sequences, 409, 783
- reserved, 40, 528, 616
- word count example, 19–21

World Wide Web Consortium. *See* W3C

wrapped Java object, 960

wrapper for named template, stylesheet function as, 349

wrapper for stylesheet function, named template as, 349

X

«x» flag, 926–927

- comments in regex and, 926
- spaces and, 926–927
- whitespace and, 231, 926

Xalan-J processor, 9, 319

- extension functions and, 957
- extension instructions and, 140
- Java/JavaScript and, 957
- JAXP and, 1163

XDM tree model, 8, 42–43, 45–67, 185, 471, 1223, 1231

- bottom-up process of construction, 73
- characters in, 62–63
- consistency rules, 310
- definition/description, 1223
- document node in, 304
- DOM v., 42, 963–965
- items left out of, 64–65, 66
- namespaces in, 59–60
- nodes in, 49–57
- sets and, 629
- well formed document not required, 304, 309
- XQuery support and, 66–67
- xs:QName and, 857

XForms, 544

- functions specific to, 553, 709
- instance() function, 544
- XPath within, 553, 709

XHTML, 170

- empty elements, 939
- flat documents example, 340–342
- implicit hierarchies in, 340
- meta element (charset attribute), 939, 940, 941

- namespace, NameTest and, 697
- output method. *See* XHTML output method
- poem example, 171–173
- result tree, 423–424
 - backward compatibility mode and, 424
- schema, 172
- serialization. *See* serialization; XHTML output method specification, 481
- XHTML output method, 43, 44, 421, 423, 776, 939–940. *See also* serialization**
 - empty elements and, 939
 - escape-uri-attributes and, 940
 - HTML/XML output methods v., 939
 - include-content-type and, 940
- XInclude, 66, 567, 720, 752, 755, 757**
- XML**
 - Canonical, 64
 - converted to canonical form, 64–65
 - data
 - non-, 209, 903, 906
 - data model. *See* XDM
 - databases, 568
 - «//» and, 627
 - collections and, 568, 726
 - unordered() and, 902
 - data-oriented, 53, 215, 977
 - declaration. *See* XML declaration
 - editing tools, 64
 - entities, 440, 1224
 - definition/description, 1224
 - external parsed. *See* external parsed entities
 - unparsed. *See* unparsed entities
 - XML document. *See* well-formed documents
 - entity references. *See* entity references
 - escaping conventions of, 527
 - family, XSLT and, 21–26
 - formats for word processor documents, 4
 - fragment interchange proposal, 48
 - Information Set, 46, 62, 64, 65
 - narrative, 53
 - NDATA and, 209, 903, 906
 - output method. *See* XML output method
 - parsers, 8, 752
 - validation by, 62
 - Schema. *See* XML Schema(s)
 - serialization. *See* serialization; XML output method specification, 534
 - specification (case study), 1001–1048
 - standards, 522
 - syntax usage, XSLT and, 33–34
 - tree, 45–46
 - example, 46–48
 - up-conversion to, from Excel spreadsheets, 343
 - uses/benefits, 3–4, 215
 - data handling, 53, 215
 - separating data from presentation, 4
 - transmitting data between applications, 4
 - version, 67
 - version attribute and, 935–936
 - well-formed document. *See* well-formed documents
- xml (prefix), 256**
 - mapped to namespace URI
 - http://www.w3.org/XML/1998/namespace, 311
- XML 1.0**
 - Unicode characters, 527
- XML 1.1, 291**
 - output
 - undeclare-prefixes and, 447
 - parser, 104, 440
 - Unicode characters, 527
- XML declaration, 931, 935**
 - encoding attribute of, 933
 - not a processing instruction, 49
 - omit-xml-declaration and, 421, 422, 447, 935, 940
 - standalone property, 67
 - text declaration v., 931
 - xsl:processing-instruction and, 443
- «xml» namespace, 472, 473**
- XML Namespaces 1.1. *See* Namespaces 1.1, XML**
- XML output method, 43, 423, 929–936. *See also* serialization**
 - entity references and, 930
 - external parsed entities and, 929–932
 - HTML output method v., 936–937
 - indent attribute and, 934
 - internal DTD subset and, 930
 - namespace fixup and, 932
 - well-formed XML documents and, 929–932
 - XHTML output method v., 939
 - XML Namespaces and, 930
- XML Query Working Group, 25, 31**
- XML Schema(s), 24–25, 151–161**
 - books on, 151
 - built-in types of, 557
 - complex type with simple content, 154–155
 - complications, 25
 - element-only content, 53, 142, 156–157, 506, 878
 - IDs and, 61
 - mixed content. *See* mixed content
 - namespace, 99, 135
 - numeric literals and, 531
 - overview, 151–161
 - pattern facet. *See* pattern facet
 - Perl and, 829, 915
 - regex syntax and, 829, 915. *See also* regular expressions
 - Relax NG vs., 25
 - simple type definitions, 52, 152–153, 163
 - specification, 369
 - stylesheets and, 151–183

XML Schema(s) (continued)

XML Schema(s) (continued)

- substitution groups, 158–161
- type hierarchy, 157–158, 162, 217–218, 219
- type system, 38–39, 161–165, 217–218, 219, 506
- XPath within, 553
- in XSLT 2.0, 24–25, 151, 183
 - optional feature, 151, 183
- XML Schema (van der Vlist), 151**
- XML Schema 1.1, 568**
 - expressions in, 682
- XML specification (case study), 1001–1048**
 - cross references, 1007, 1029–1032
 - document header, 1012–1019
 - formatting, 1002–1003
 - HTML outline, 1008–1012
 - hyperlinks
 - cross references and, 1029–1032
 - preface, 1007
 - section headers, 1023
 - table of contents, 1019, 1021, 1023
 - list production for, 1028–1029
 - overlay stylesheets, 1041–1044
 - diffspec.xml, 1003, 1041–1044
 - REC-xml.xml, 1003, 1044
 - preface, 1004–1008
 - production rules, 1033–1041
 - section headers, 1023–1024
 - stylesheets for other specifications, 1044–1047
 - funcproto.xml, 1046–1047
 - xmlspec.xml, 1047
 - xsl-query.xml, 1047
 - xsltdiff.xml, 1046
 - xslt.xml, 1045–1046
 - table of contents, 1019–1023
 - text formatting, 1024–1028
- xml:base attribute, 54, 357, 373, 374, 720, 755, 756, 868. See also base URIs**
 - attribute value templates and, 124
- xml:id, 62, 803**
- xml:lang, 358, 374, 409, 461, 781**
 - attribute value templates and, 124
 - lang() and, 819
- XmlNode (.NET class), 1159**
- xmlns, 17, 256**
 - attribute value templates and, 124
- xmlns="" , 311, 475. See also undeclarations, namespace**
- XmlResolver (.NET), 138, 567, 751, 755, 965**
- xml:space attribute, 72, 142, 146, 358, 374, 397, 441, 492**
 - attribute value templates and, 124
 - indent attribute and, 934
 - in stylesheet, 147

xmlspec.xml stylesheet module, 1047

- XMLSpy development environment, 10, 12, 13, 15–16, 91, 479, 1195, 1215–1216. See also AltovaXML processor**
 - version attribute and, 479
- <?xml-stylesheet?> processing instruction, 67, 68, 89, 90, 99–101, 149**
 - alternate attribute, 101
 - charset attribute, 101
 - client side transformation, 100
 - for controlling transformation, 100
 - embedded stylesheet in, 469
 - example, 444–445
 - Firefox and, 101
 - href attribute, 100
 - Firefox and, 101
 - Internet Explorer and, 101
 - id attribute, 98, 469–470
 - Internet Explorer and, 101
 - in JAXP, 1184, 1189–1190
 - pseudo-attributes, 100–101, 444
 - Recommendation, 99, 444
 - server-side transformation, 100, 319
 - title attribute, 101
- XML-to-XML transformation, 290, 396**
- XMLValidator (Altova), 1218–1219**
- XM_XPATH_FUNCTION_LIBRARY (Gestalt), 962**
- XM_XPATH_SYSTEM_FUNCTION (Gestalt), 962**
- XPath**
 - chevrons and, 522
 - collations in, 84
 - conditional expressions. *See* conditional expressions
 - context. *See* context
 - data model. *See* XDM
 - default namespaces. *See* default namespaces;
 - xpath-default-namespace
 - as embedded language. *See* embedded languages
 - evaluation context. *See* evaluation context
 - as expression language, 185, 553
 - expressions. *See* expressions
 - functions written in, 546
 - grammar, 521
 - lexical rules. *See* lexical rules
 - operators. *See* operators
 - parser, 527
 - processor, schema-aware, 54
 - production rules. *See* production rules
 - read-only language, 188, 526
 - sorting and, 562
 - specification document, 522
 - as sublanguage. *See* embedded languages
 - syntactic rules, 521, 524
 - syntax. *See* syntax
 - XSLT and, 22–23, 39–40

XPath 1.0

- «..» and, 623
- compatibility mode. *See* backward-compatibility mode
- «except» operator and, 631
- «intersect» operator and, 631
- type system, 186–187, 224, 506
- as weakly typed language, 555

XPath 2.0

- changes in arithmetic operators, 573
- changes in general comparison operators, 588–589, 1143
- changes in node comparisons, 594
- development, 32–33
- history, 22–23
- path expressions changes in, 601–602
- Recommendation, 82
- regular expressions. *See* regular expressions
- sequence manipulation, 633. *See also* sequence expressions
- syntax, 82–83, 1117–1122
- type system, 186–187, 218–219, 224
- types/typing, 186–187
 - multiple approaches to, 187
 - static/strong, 187

xpath-default-namespace attribute, 98–99, 349, 358, 374, 468, 480–482

- default namespace declarations and, 480–482, 615
- effect/usage, 480–482
- on literal result elements, 468
- QNames and, 536
- temporary documents and, 482
- unprefixed names and, 481–482
 - in expressions, 481–482
 - in patterns, 481–482
- [xsl:], 481, 556, 615

XPathDocument (.NET class), 1158**XPathNavigator (.NET class), 1159****XPointer, 22, 40**

- fragment identifier, 756
- namespace declarations, 556
- XPath expressions within, 553, 682

XProc, 67, 86, 448, 956

- patterns in, 682
- pipelines and, 992

XQJ (XQuery API for Java), 961**XQuery 1.0, 26, 541**

- attributes with same name in, 257
- collations and, 562
- data model. *See* XDM
- distinct-values() and, 750
- doc() and, 751
- FLWOR expression in, 602, 647–648
- and functions written in XPath, 546
- importing schemas and, 369
- namespace declarations, 556

- sequence type descriptors and, 668
- static typing and, 563, 680
- steps creating new nodes, 605
- trace() and, 897
- typeswitch expression, 546, 669, 680
- «union» operator and, 602
- value comparisons and, 582
- variables in namespace, 559
- Working Group, 25, 31
- workshop, 31
- XDM and, 66–67
- XSLT vs., 26, 31–32

XQuery API for Java (XQJ), 961**xs namespace prefix (namespace <http://www.w3.org/2001/XMLSchema>), 135, 190** **xsi namespace, 167**

- namespace alias for, 183

xs:any, 314**xs:anyAtomicType, 162, 192, 713**

- matching atomic values and, 670

xs:anyType, 313

- sequence type descriptors and, 670
- xs:untyped v., 314

xs:anyURI, 190, 193–194

- codepoint-equal() and, 724
- converting from string, 663
- converting to other types, 657
- «eq» and, 583
- «lt», «le», «ge», «gt», and, 584
- namespace-uri() and, 838
- percent encoding in, 194
- promotion of, to xs:string, 81, 427, 506, 548, 713, 838

xs:base64Binary, 190, 203, 208

- converting from hexBinary, 662
- converting from string, 663
- converting to other types, 657
- «eq» and, 584
- «lt», «le», «ge», «gt», and, 585

xs:boolean. *See* booleans**xs:boolean() constructor function, 552, 722**

- boolean() function v., 552

xs:byte, 211, 212**xs:date, 162, 190, 195–196**

- adjusting timezone, 297
- converting from, 207
- converting from dateTime, 658
- converting from string, 663
- converting to other types, 658
- current, 738
- day component from, 744
- day of the month from, 744
- «eq» and, 583
- formatting, 781

xs:date (continued)

xs:date (continued)

- «lt», «le», «ge», «gt», and, 585
 - minus xs:date, 580
 - starting instants of dates and, 580
 - month component from, 833
 - partially ordered dates and, 196, 585
 - timezone component from, 893
 - xs:dayTimeDuration added to, 578
 - xs:time added to, 743
 - xs:yearMonthDuration added to, 579
 - year component from, 911
- xs:dateTime, 190, 196–197. See also dates/times**
- adjusting timezone, 297
 - construction of, 743
 - dateTime(), 743
 - converting from, 207
 - converting from date, 658
 - converting from string, 663
 - converting to other types, 658–659
 - current, 738
 - day component from, 744
 - day of the month from, 744
 - «eq» and, 583
 - formatting, 781
 - hour component, 800
 - «lt», «le», «ge», «gt», and, 585
 - minus xs:dateTime, 580
 - instants in time and, 580
 - minutes component from, 832
 - month component from, 833–834
 - seconds component from, 873–874
 - timezone component from, 893
 - xs:dayTimeDuration added to, 578
 - xs:yearMonthDuration added to, 579
 - year component from, 911
- xs:dayTimeDuration, 192, 205–206**
- added to xs:date, 578
 - added to xs:dateTime, 578
 - added to xs:time, 578
 - converting to, 668
 - divided by xs:dayTimeDuration, 580–581
 - «lt», «le», «ge», «gt», and, 585
 - plus(add/subtract) xs:dayTimeDuration, 579
 - sum() and, 889
- xs:decimal, 190, 197–198. See also numeric functions**
- arithmetic operators and, 571
 - comparison of numeric values and, 585
 - converting from boolean, 657
 - converting from double, 660
 - converting from float, 661
 - numeric overflow in, 661
 - converting from string, 663
 - converting to other types, 659
 - formatting. *See* format-number()

- literal constants, 191, 530
- overflow, 575
- precision in, 575
- type promotion, 572, 585
- underflow, 575

xs:double, 198–201. See also numeric functions

- arithmetic operators and, 554–555, 571
- comparison of numeric values and, 585
- conversion to, 454, 852
 - number() and, 852
 - xs:untypedAtomic and, 574
- converting from boolean, 658
- converting from decimal, 659
 - numeric overflow in, 659
- converting from float, 661
- converting from string, 663
- converting to other types, 660
- formatting. *See* format-number()
- literal constants, 191, 530
- NaN value, 586, 852
- number() and, 851
- operators/functions and, 200
- ordered, 199
- type promotion, 572, 585
- unordered, 199
- XPath expressions for, 199
- on xsl:variable, 455

xs:duration, 190, 205, 208–209. See also

- xs:dayTimeDuration; xs:yearMonthDuration**
- arithmetic and, 578
- converting from string, 663
- converting to other types, 660
- day component from, 745
- «eq» and, 584
- hour component, 801
- «lt», «le», «ge», «gt», and, 585
- minutes component from, 832–833
- month component from, 833
- seconds component from, 874–875
- year component from, 911–912

xs:ENTITIES, 670

xs:ENTITY, 213, 215, 217

xs:float, 81, 190, 193, 198, 208. See also numeric functions

- arithmetic operators and, 571
- comparison of numeric values and, 585
- converting from boolean, 658
- converting from decimal, 659
- converting from double, 660
- converting from string, 663
- converting to other types, 661
- formatting. *See* format-number() function
- NaN value, 586
- single-precision, 576
- type promotion, 572, 585

- xs:gDay, 190, 207**
 - converting from date, 658
 - converting from dateTime, 658
 - converting from string, 663
 - converting to other types, 661
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
- xs:gMonth, 190, 207**
 - converting from date, 658
 - converting from dateTime, 659
 - converting from string, 663
 - converting to other types, 661
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
- xs:gMonthDay, 190, 207**
 - converting from date, 658
 - converting from dateTime, 659
 - converting from string, 663
 - converting to other types, 662
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
- xs:gYear, 190, 207**
 - converting from date, 658
 - converting from dateTime, 659
 - converting from string, 663
 - converting to other types, 662
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
- xs:gYearMonth, 190, 207**
 - converting from date, 658
 - converting from dateTime, 659
 - converting from string, 663
 - converting to other types, 662
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
- xs:hexBinary, 190, 203, 208**
 - converting from base64Binary, 657
 - converting from string, 663
 - converting to other types, 662
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
- xs:ID, 61, 213, 215, 305, 802**
- xs:IDREF, 213, 215, 305**
 - id() and, 802
 - idref() and, 805
 - meaning, 215
- xs:IDREFS, 217, 305, 670**
 - id() and, 802
 - idref() and, 805
- xs:import, 180**
 - xsl:import-schema and, 180
- xs:int, 211, 212**
- xs:integer, 162, 201**
 - arithmetic operators and, 571
 - atomic values, 191
 - comparison of numeric values and, 585
 - converting from string, 667
 - converting to, 667–668
 - literal constants, 191, 530
 - special status, 191, 201
 - type promotion, 572, 585
- xs:integer() constructor function, 780. See also floor() function**
 - «idiv» and, 780
 - round() and, 872
- xs:key, 179**
- xs:keyref, 179**
- xs:language, 213, 215**
- xs:long, 211, 212**
- xs:Name, 213, 215**
- xs:NCName, 213, 215**
 - local-name-from-QName() and, 826
 - prefix-from-QName and, 857
- xs:negativeInteger, 211, 212**
- xs:NMTOKEN, 189, 213, 215, 217**
- xs:NMTOKENS, 670**
- xs:nonNegativeInteger, 211, 212**
- xs:normalizedString, 144, 213, 214**
 - meaning, 214
 - normalize-space() function v., 144, 215, 846
- xs:NOTATION, 180, 190, 209–210. See also unparsed entities**
 - converting from string (not allowed), 657, 663
 - converting from xs:NOTATION (not allowed), 663
 - converting to other types, 663
 - DTDs and, 209
 - «eq» and, 584
 - «lt», «le», «ge», «gt», and, 585
 - namespace-sensitive, 260
 - validation of, 180
- xs:pattern, 214**
- xs:positiveInteger, 211, 212**
- xs:QName, 51, 60, 190, 201–202, 261, 393. See also QNames**
 - attribute creation and, 261–263
 - converting from string, 664
 - converting to other types, 663
 - «eq» and, 584
 - local-name-from-QName() and, 826
 - «lt», «le», «ge», «gt», and, 585
 - namespace prefix in, 857
 - namespace URI of, 50–51, 840–841
 - namespace-sensitive, 260
 - node-name() and, 843–844
 - parts of, 857
 - type annotation of attribute as, 393
 - validation, 180
 - XDM and, 857
- xs:redefine, 370**
- xs:schema, 368**

xs:short

xs:short, 211, 212

xs:simpleType, 191–192, 654

xs:string, 52, 190, 202–204, 213, 214, 215. *See also* **string()**

- converting from anyURI, 657
- converting from base64binary, 657
- converting from date, 658
- converting from dateTime, 659
- converting from decimal, 659
- converting from double, 660
 - NaN and, 660
- converting from duration, 660
 - zero-length duration in, 660
- converting from float, 661
- converting from gDay, 661
- converting from gMonth, 661
- converting from gMonthDay, 662
- converting from gYear, 662
- converting from gYearMonth, 662
- converting from hexBinary, 662
- converting from NOTATION, 663
- converting from QName, 663
- converting from time, 664
- converting from xs:integer, 667
- converting to derived types, 665–666
- converting to other types, 663–664
 - schema validation rules and, 663
- «eq» and, 583
- literal constants, 191
- «lt», «le», «ge», «gt», and, 584
- meaning, 214
- promotion of xs:anyURI to, 81, 427, 506, 548, 713

xs:string() function, 52

xs:time, 190, 204–205

- converting from dateTime, 659
- converting from string, 663
- converting to other types, 664
- cyclic values and, 585
- «eq» and, 583
- hour component, 800
- «lt», «le», «ge», «gt», and, 585
- minus xs:time, 580
- seconds component from, 873–874
- timezone component from, 893
- xs:date added to, 743
- xs:dayTimeDuration added to, 578

xs:token, 213, 215

xs:unique, 179

xs:unsignedByte, 211, 212

xs:unsignedInt, 211, 212

xs:unsignedLong, 211, 212

xs:unsignedShort, 211, 212

xs:untyped, 54, 297, 305, 312, 477

- strip and, 314

type annotation, 741

xs:anyType v., 314

xs:untypedAtomic, 52, 54, 259, 297, 305, 477, 506, 548

atomic values, 191

matching, 670

atomization of operands and, 574, 586, 589

conversions of, 427, 714

converting from anyURI, 657

converting from base64binary, 657

converting from boolean, 658

converting from date, 658

converting from dateTime, 659

converting from decimal, 659

converting from double, 660

converting from duration, 660

converting from float, 661

converting from gDay, 661

converting from gMonth, 661

converting from gMonthDay, 662

converting from gYear, 662

converting from gYearMonth, 662

converting from hexBinary, 662

converting from NOTATION, 663

converting from QName, 663

converting from string, 664

converting from time, 664

converting to boolean, 552

converting to other types, 664

converting to xs:double, 574, 589

schema validation and, 670

strip and, 314

type annotation, 741

value comparisons and, 587

xs:whiteSpace facet, 143, 214, 663, 665. *See also* **whitespace**

normalize-space() and, 846

xs:yearMonthDuration, 192, 205–206

added to xs:date, 579

added to xs:dateTime, 579

converting to, 668

divided by xs:yearMonthDuration, 580–581

«lt», «le», «ge», «gt», and, 585

multiplied/divided by number, 579

plus(add/subtract) xs:yearMonthDuration, 579

sum() and, 889

xsi namespace

xsi:nil, 167, 182–183

xsi:noNamespaceSchemaLocation,
167, 174

xsi:schemaLocation, 167, 174

xsi:type, 167, 181–182

xsi:nil, 167, 182–183, 673, 699

nilled() and, 843

question mark and, 673

- xsl:noNamespaceSchemaLocation, 167**
 - in result document, 174
- xsl:schemaLocation, 167, 314, 478**
 - AltovaXML and, 671, 1217
 - doc() function and, 558, 671
 - locating schemas and, 314
 - in result document, 174
- xsl:type, 167, 181–182, 261, 263, 314, 392**
 - union type and, 182
- [xsl:]default-collation, 112, 470**
- [xsl:]exclude-result-prefixes, 112, 118, 119, 315, 358, 392**
 - redundant namespace nodes and, 394
- [xsl:]extension-element-prefixes, 111, 113, 118, 138, 317**
 - element-available() and, 766
- [xsl:]inherit-namespaces, 113, 120, 310, 393**
- [xsl:]type attribute, 113, 176, 467. See also xsl:type**
 - attribute value templates and, 124
 - of literal result elements, 370
- [xsl:]use-attribute-sets, 113, 117**
- [xsl:]use-when, 112, 127, 130, 319, 468, 478. See also use-when attribute**
 - function-available() and, 792
 - type-available() and, 900
- [xsl:]validation, 113, 176, 467**
- [xsl:]version, 113, 496. See also version attribute**
 - simplified stylesheet and, 125
- [xsl:]version attribute, 318**
- [xsl:]xpath-default-namespace, 481, 556, 615. See also xpath-default-namespace attribute**
- xsl:instruction, 675–676**
- xsl:is-schema-aware (system property), 478, 891**
- xsl:product-name (system property), 891**
- xsl:product-version (system property), 891**
- xsl:schema-aware, 370**
- xsl:supports-backwards-compatibility (system property), 892**
- xsl:supports-namespace-axis (system property), 892**
- xsl:supports-serialization (system property), 892**
- xsl:vendor (system property), 478**
- xsl:vendor-url (system property), 891**
- xsl:version (system property), 891**
 - simplified stylesheets and, 125
- xsl:analyze-string, 109, 230–237, 278**
 - context item and, 232, 736
 - “context item is not a node” error, 624
 - regex attribute of, 123
 - regex in, 123, 232–234
 - regex syntax and, 915–927
 - regex-group() and, 860–861
 - replace() v., 864
 - tokenize() and, 895
 - unparsed-text() and, 906
 - xsl:fallback and, 231, 317
 - xsl:matching-substring within, 386
 - xsl:non-matching-substring within, 402
- xsl:apply-imports, 109, 237–240, 360**
 - current template rule and, 399
 - import precedence and, 239
 - parameters and, 238
 - template parameters and, 238, 431
 - xsl:next-match v., 399, 1043
 - xsl:with-param and, 238, 517
- xsl:apply-templates, 71, 73, 82, 85, 109, 146, 238, 240–253, 274, 332**
 - context items and, 543, 735
 - context position and, 564
 - context size and, 564, 821
 - current template rule and, 399
 - in global variables, 245
 - last() and, 734, 821
 - match attribute and, 484
 - modes and, 247–251, 484, 486
 - result, 245–246
 - sorting and, 199, 241
 - template parameters and, 243, 431
 - template rule for, 362
 - variables and, 510
 - xsl:for-each v., 246–247
 - xsl:preserve-space and, 441
 - xsl:sort and, 240, 455
 - xsl:sort in, 455
 - xsl:with-param and, 240, 517
- xsl:attribute, 109, 254–266**
 - attribute node created with, 255
 - attribute value templates v., 122
 - literal result elements and, 117
 - namespace attribute, 255
 - namespace declarations and, 391
 - type attribute of, 179, 370
 - validation attribute, 179
- xsl:attribute-set, 97, 105, 117, 256, 257, 260, 266–271, 309**
 - import precedence and, 267, 361
 - top-level declaration, 468
- xsl:call-template, 82, 109, 238, 239, 271–280**
 - context item and, 273
 - current template rule and, 400
 - import precedence and, 272
 - name attribute and, 124, 272, 485
 - recursive use of, 275–280
 - result of, 273–274
 - template parameters and, 272, 431
 - variables and, 510
 - xsl:with-param and, 272, 517
- xsl:character-map, 105, 280–282**
 - import precedence and, 281, 361
 - top-level declaration, 468

xsl:character-map (continued)

xsl:character-map (continued)

- use-character-maps and, 422, 447
- xsl:output-character in, 424

xsl:choose, 109, 282–284

- conditional expressions v., 552–553
- evaluation of, 552
- xsl:if v., 355
- xsl:otherwise within, 420
- xsl:when in, 515

xsl:comment, 109, 285–287

xsl:copy, 110, 287–291

- acquiring namespace nodes with, 392
- attribute creation and, 260–261
- default-validation and, 471
- identity template. *See* identity template
- type attribute of, 370
- use-attribute-sets attribute of, 268
- xsl:copy-of v., 179

xsl:copy-of, 109, 110, 166, 261, 292–298, 332, 351

- attribute creation and, 260–261
- default-validation and, 471
- instruction, 292
- type attribute of, 370
- xsl:copy v., 179
- xsl:value-of v., 499

xsl:decimal-format, 106, 298–302

- countries/languages and, 301
- digits in, 302
- format-number() and, 789
- formatting and, 403
- grouping-separator in, 300, 790
- groups of digits in, 300, 790
- import precedence and, 299, 361
- infinity in, 300, 302, 790
- minus-sign, 300, 789, 790
- NaN and, 300, 790
- pattern-separator, 300, 789
- percent symbol, 300, 790
- picture string, 300
- per-mille, 300, 790
- picture string, 300
- picture string, 300, 789–791
- symbols in, 300
- top-level declaration, 298, 468
- zero-digit, 300, 789

xsl:document, 110, 303–306

- type attribute of, 370
- validation of temporary document and, 175
- xsl:result-document v., 448

xsl:element, 110, 176, 306–316

- converting attributes to child elements with, 315–316
- default-validation and, 471
- element created with, 307–310
- content of, 307–310
- name of, 307
- literal result element v., 306, 308, 315

- namespace attribute, 308
- sequence constructor evaluation, 309
- type attribute of, 370
- use-attribute-sets attribute of, 268, 309

xsl:fallback, 110, 111, 140, 231, 316–321

- different versions of XSLT and, 318
- element-available() and, 766
- extension elements and, 317
- extension instructions and, 769
- system-property() and, 892
- xsl:analyze string and, 317
- xsl:next-match and, 317, 399
- in xsl:perform-sort, 438
- xsl:result-document and, 317, 447
- in xsl:sequence, 453
- XSLT 1.0 processor and, 317

xsl:for-each, 85, 110, 238, 274, 322–326

- context item and, 543, 735
- “context item is not a node” error, 624
- context position and, 564
- context size and, 564, 821
- current template rule and, 400
- last() and, 734, 821
- select attribute and, 322–323
- sorting and, 199, 323–324
- xsl:apply-templates v., 246–247
- xsl:sort in, 323, 455

xsl:for-each-group, 110, 116, 249, 326–344

- concat() and, 730
- context item and, 736
- current-group() and, 739
- default-collation and, 470
- distinct-values() v., 750
- grouping keys and, 384, 816
- grouping keys v., 816
- «is» operator and, 593
- patterns in, 681
- transitivity and, 587
- unparsed-text() and, 906
- xsl:sort in, 330, 455

xsl:function, 82, 106, 135, 161, 252, 344–353

- as attribute of, 346, 371
- context item/position/size and, 346
- conversions and, 346
- declaration, 344
- function-available() and, 793
- import precedence and, 346, 361
- override attribute, 346–347
- stylesheet functions with, 106, 135, 136, 271, 344
- top-level declaration, 345, 469
- writing functions with, 136
- xsl:param in, 425
- xsl:sequence in, 453

xsl:if, 110, 353–356

- boolean() and, 722
- effective boolean value in, 354

1312

- evaluation of, 552
- xsl:choose v., 355
- xsl:import, 45, 91, 94, 106, 357–368. *See also* import precedence**
 - class hierarchy, 95
 - customizing presentation with, 368
 - in example, 95–98
 - href attribute of, 357
 - import precedence and, 94, 148, 357, 358, 362
 - top-level declaration, 469
 - as top-level element, 357
 - xsl:include v., 94
- xsl:import-schema, 106, 163, 180, 192, 259, 313, 368–372, 669**
 - declarations
 - multiple, 370
 - non-schema-aware processor and, 370
 - import precedence and, 362, 370
 - in-scope schema definitions and, 557
 - namespace attribute, 180
 - not necessary to import every schema with, 370
 - schema-location attribute, 181, 369
 - system-property() and, 769
 - top-level declaration, 368, 469
 - type-available() and, 900
 - use-when and, 769
 - xs:import and, 180
- xsl:include, 45, 91, 94, 106, 358, 372–375**
 - example with, 91–95
 - href attribute of, 373
 - import precedence and, 148, 362
 - modes and, 374
 - with named attribute sets, 375–376
 - stylesheet modules and, 372
 - top-level declaration, 469
 - as top-level element, 372
 - xsl:import v., 94
- xsl:key, 85, 106, 376–385**
 - concat() and, 730
 - declarations with same name, 383–384
 - default-collation and, 470
 - faster access with, 378
 - import precedence and, 362
 - index/hash table and, 379
 - key definitions in, 812
 - key() and, 812
 - match attribute of, 377
 - optimization of joins and, 644
 - patterns in, 681
 - top-level declaration, 376, 469
 - use attribute, 376
 - global variables and, 378
 - values of key and, 377
- xsl:matching-substring, 231, 386**
 - regex and, 386
 - within xsl:analyze-string, 386
- xsl:message, 69–70, 110, 386–390**
 - diagnostic output from, 388, 897
 - error conditions and, 388
 - error() and, 775
 - languages and, 389
 - MSXML parser and, 388
 - terminate attribute, 386, 387
 - trace() v., 897
 - xsl:result-document v., 387
- xsl:namespace, 60, 110, 256, 262, 311, 390–393**
 - literal result elements and, 121
 - namespace node construction with, 390
- xsl:namespace-alias, 106, 121, 394–399**
 - import precedence and, 362
 - stylesheet production and, 394
 - top-level declaration, 394, 469
- xsl:next-match, 110, 238, 399–402**
 - current template rule and, 399–400
 - instruction, 399
 - subtypes and, 701
 - template parameters and, 400, 431
 - template rules of lower precedence and, 1043
 - xsl:apply-imports v., 399, 1043
 - xsl:fallback and, 317, 399
 - xsl:with-param and, 400, 517
- xsl:non-matching-substring, 231, 402–403**
 - regex and, 402
 - within xsl:analyze-string, 402
- xsl:number, 110, 146, 403–419. *See also* level attribute**
 - from attribute and, 412, 417
 - count attribute, 411–412
 - count() v., 734
 - digits in, 408–409
 - format attribute in, 407
 - format string and, 405, 407–408
 - format-number() v., 403, 406, 791
 - formatting numbers with, 403, 408–410
 - grouping-separator in, 409
 - groups of digits in, 409
 - lang attribute of, 409
 - languages and, 409
 - patterns in, 681
 - position() v., 856
 - punctuation tokens, 407
 - sequence number and, 405–407
 - tokens
 - formatting, 407
 - punctuation, 407
 - xsl:sort and, 856
- xsl:otherwise, 109, 283, 420**
 - within xsl:choose, 420
 - xsl:when and, 420, 516

xsl:output

xsl:output, **43–44**, **106**, **280**, **420–424**. *See also* **serialization**; **xsl:character-map**;

xsl:result-document

character encoding in, 422
encoding attribute of, 421–422, 941
escape-uri-attributes in, 422
import precedence and, 362, 423
include-content-type in, 421, 422
media-type in, 421, 422
result document and, 420
Saxon extensions and, 1208
serialization and, 940–941
serialization attributes in, 421–422
top-level declaration, 469
XHTML and, 421
xsl:result-document and, 421

xsl:output-character, **281**, **424–425**

character attribute, 425
character maps and, 425
in xsl:character-map, 424

xsl:param, **68**, **82**, **106**, **109**, **161**, **244**, **272**, **346**, **362**, **400**, **425–437**

as attribute of, 371, 425, 427–428
with default value (example), 432–433
global parameters and, 425, 427
import precedence and, 362
local parameters to templates/functions and, 425, 427
overriding, 431
required attribute, 238, 244, 273, 428, 487
select attribute and, 428
top-level declaration, 469
variable references and, 540
in xsl:function, 425
in xsl:template, 425, 484, 487
xsl:variable v., 429, 502

xsl:perform-sort, **71**, **110**, **331**, **343**, **437–438**

global variables and, 438
sorting sequences with, 437, 438
whitespace-only text nodes and, 439–440
xsl:fallback in, 438
xsl:sort in, 438, 455

xsl:preserve-space, **106**, **142**, **439–442**

element names in, 440, 482
import precedence and, 362
position() function and, 441
QNames and, 536
top-level declaration, 439, 469
whitespace-only text nodes and, 757
xsl:apply-templates and, 441
xsl:strip-space and, 439, 465

xsl:processing-instruction, **110**, **442–445**

XML declaration and, 443

xsl:result-document, **45**, **68**, **110**, **445–452**

attribute value templates in, 448

character encoding in, 447
default-validation and, 471
encoding attribute of, 447–448
escape-uri-attributes in, 447
format attribute in, 446, 448
href attribute of, 446
include-content-type in, 447, 448
instruction, 446
media-type in, 447, 448
result trees and, 445
creation of, 445
serialization of, 107, 445
Saxon extensions and, 1208
serialization attributes in, 446–447
side effects, 449
type attribute of, 173, 370
validation attribute of, 173
xsl:document v., 448
xsl:fallback and, 317, 447
xsl:message v., 387
xsl:output and, 421

xsl:sequence, **71**, **110**, **271**, **351**, **353**, **452–455**

xsl:fallback in, 453
in xsl:function, 453
xsl:value-of v., 454

xsl:sort, **109**, **199**, **240**, **241**, **242**, **323**, **328**, **455–464**

attributes as attribute value templates in, 330
in backward-compatibility mode, 459
concat() and, 730
constructor functions and, 459
context item, 457
context position, 457
context size, 457
conversions and, 459
current-group() and, 739
data-type and, 456, 458
default collation, 461
default-collation and, 470
lang attribute of, 456, 460
languages and, 456, 460
last() within select expression of, 823
sequence constructor in, 457
sort keys and, 455. *See also* sort key(s)
stable attribute, 242, 324, 456, 457, 458
xsl:apply-templates and, 240, 455
in xsl:for-each, 323, 455
in xsl:for-each-group, 330, 455
xsl:number and, 856
in xsl:perform-sort, 438, 455

xsl:strip-space, **46**, **106**, **142**, **143**, **148**, **465**

element names in, 465, 482
import precedence and, 362
normalize-space() v., 846
QNames and, 536
top-level declaration, 465, 469

XSLT (Extensible Stylesheet Language: Transformations)

- whitespace-only text nodes and, 439–440, 465, 493, 757
- xsl:preserve-space and, 439, 465
- xsl:stylesheet, 89, 94, 98–99, 148, 318, 374, 465–482. See also xsl:transform**
 - attributes on, 98–99
 - namespace declarations on, 99, 466–467
 - top-level declarations in, 468–469
 - xsl:transform v., 465, 495
- xsl:template, 68, 70, 106, 161, 242, 272, 483–491**
 - as attribute of, 371, 487
 - evaluating templates, 486–487
 - import precedence and, 243, 362
 - match attribute of, 68, 484–485
 - errors and, 485
 - global parameters and, 485
 - global variables and, 378, 485
 - pattern, 484
 - template rules and, 487
 - xsl:apply-templates and, 484
 - patterns and, 483, 484, 681
 - priority attribute of, 79
 - SequenceType and, 484
 - top-level declaration, 469
 - xsl:param in, 425, 484, 487
- xsl:text, 69, 110, 146, 148, 492–494**
 - disable-output-escaping and, 492, 493, 945
 - literal text and, 492
 - in sequence constructor, 492
 - whitespace-only text nodes and, 492–493
- xsl:transform, 89, 94, 98, 495. See also xsl:stylesheet**
 - xsl:stylesheet v., 465, 495
- xsl:value-of, 110, 351, 495–500**
 - concat() v., 730
 - disable-output-escaping and, 496, 497, 945
 - instruction, 495
 - select attribute and, 496
 - sequence constructors and, 495
 - string-join() v., 880
 - strings returned from function and, 497–498
 - temporary trees and, 497
 - unexpected results from, 498–500
 - unthinking usage of, 497
 - value of node as string and, 496
 - xsl:copy-of v., 499
 - xsl:sequence v., 454
- xsl:variable, 106, 110, 161, 500–515. See also variables; xsl:param**
 - as attribute of, 274, 371
 - global variables and, 501
 - import precedence and, 362
 - local variables and, 501
 - select attribute in, 501
 - sequence constructor in, 501
 - SequenceType and, 501
 - temporary tree and, 619
 - top-level declaration, 469, 500
 - variable references and, 540
 - xs:double on, 455
 - xsl:param v., 429, 502
- xsl:when, 283, 515–516**
 - boolean() and, 722
 - effective boolean value in, 516
 - test expression, 283
 - in xsl:choose, 515
 - xsl:otherwise and, 420, 516
- xsl:with-param, 82, 109, 238, 240, 272, 400, 487, 517–518**
 - as attribute of, 371, 518
 - template parameters and, 431, 517–518
 - tunnel parameters and, 429, 518
 - value of parameter and, 517–518
 - xsl:apply-imports and, 238, 517
 - xsl:apply-templates and, 240, 517
 - xsl:call-template and, 272, 517
 - xsl:next-match and, 400, 517
- XSL (Extensible Stylesheet Language), 22**
 - ambiguity of term, 22
 - history, 26–33
 - first proposal, 28–30
 - prehistory, 26–28
 - “A Proposal for XSL,” 28–29
 - purpose of, 28–29
 - WD-xsl and, 467
 - Working Group, 26, 30, 31
- XSL Formatting Objects (XSL-FO), 22, 268, 410**
- XSL-FO. See XSL Formatting Objects**
- xsl-query.xsl stylesheet module, 1045, 1046, 1047**
- XSLT (Extensible Stylesheet Language: Transformations)**
 - CSS vs., 24, 982
 - family tree case study, 1080, 1081
 - as declarative language, 8, 40, 979
 - defined, 3–5
 - design themes, 5
 - elements, 227–519
 - extensibility. *See* extensibility, XSLT
 - formatting stage, 7
 - functions, 546
 - (list), 712
 - instructions, 108–110, 764
 - (full list), 109–110, 765
 - element-available() and, 764–770
 - language characteristics, 33–40
 - computational power, 1099, 1112
 - general-purpose programming, 1113
 - implementation-defined, 954–955
 - manipulation/reporting tool for complex structured data, 1049, 1050, 1098
 - no side effects principle, 29, 34–35, 69–70, 350

XSLT (Extensible Stylesheet Language: Transformations) (continued)

XSLT (Extensible Stylesheet Language: Transformations) (continued)

rule-based structure, 35–38, 41
types based on XML Schema, 38–39
XML syntax usage, 33–34
XPath as embedded language, 22, 39–40, 527, 553. *See also* embedded languages
for loop and, 274
namespace, 466, 471, 475
patterns. *See* patterns
processing model. *See* processing model, XSLT
processors. *See* processors, XSLT
SQL vs., 8–9
structural transformation, 7
transformation task, 3
versions. *See* versions
while loop and, 274
without updateable variables, 274, 350
XML family and, 21–26
XML namespaces and, 23–24
XPath and, 22–23, 39–40
XQuery vs., 26, 31–32
XSL-FO and, 22

XSLT 1.0
‘first item’ rule, 132, 133, 480, 498, 1142–1143
forward compatibility in, 130–131, 479
processors, 9–10
 element-available() and, 766
 multiple result documents and, 44–45, 319–320
 use-when attribute and, 767
 version attribute and, 479
 xsl:fallback and, 317
Recommendation, 28, 30, 479
transition stages to XSLT 2.0, 1139–1145
type system, 39
XSLT 2.0 vs., 5

XSLT 1.1 working draft, 30–31
extension functions in Java/JavaScript and, 135, 953
xsl:document in, 303. *See also* xsl:document

XSLT 2.0
changes in patterns, 682–683
development, 5, 32–33
forward/backward compatibility modes, 479
minimum set of atomic types, 669–670
Recommendation, 5, 32–33, 479
requirements, 32
schema for stylesheets, 352
 examples, 674–677
schema processing in, 151, 183
 optional, 151, 183
stylesheet changes in, 90
system properties (list), 891–892
transition stages from XSLT 1.0 to, 1139–1145

type system based on XML Schema. *See* type systems
XML Schemas in, 24–25, 151, 183
 optional feature, 151, 183
XSLT 1.0 vs., 5
XSLT 1.1 vs., 30–31

XSLT namespace, 765

instructions in, 317, 765
namespace declarations, 466, 765

XSLT2 object (Altova), 1219

XSLTC, 10

xslt:diff.xsl stylesheet module, 1046

XslTransform (.NET class), 1159–1160

xslt.xsl stylesheet module, 1045–1046

xt processor, 319, 320, 550. *See also* Clark, James
forward-compatible processing and, 320

Y

year component

from xs:date, 911
from xs:dateTime, 911
from xs:duration, 911–912

year, negative, 196

year-from-date() function, 911

year-from-dateTime() function, 911

years-from-duration() function, 911–912

Z

zero(s)

division by, 574, 575
finite nonzero values, 198
formatting, 409
leading, 300, 783, 790
or more predicates, 620
negative, 199, 576, 723
positive, 198, 576
trailing, 300, 790

zero sign, Greek, 460

zero-digit (xsl:decimal-format), 300, 789

zero-length duration, 660

zero-length strings, 203, 243, 308

default value as, 503
empty sequences and, 203
empty() and, 770
namespace URIs and, 391
regex and, 231, 863
relative reference of, 868
string-length() and, 881

zero-length text nodes, 258, 304, 496

zero-or-one() function, 912–913. *See also* exactly-one();

one-or-more()

static type checking and, 912–913