

# Index

- `+=` operator 47
- 2D graphics 92–5
- 3D graphics 99–100
- access point
  - selection dialog 169
  - setting default 172
- `address()` function 68, 75
- animal sounds, recording 82–3
- APIs (Application Programming Interfaces) 199–201
- `append()` function 39
- AppleScript, controlling
  - applications with 146–8
- application body 59
- application building
  - application structure 52–6
    - application body 59, 60
    - application menu 56–9
    - content handler 60
    - tabs 59
  - functions 49–52
  - SMS game server 70–6
- application keys, Web API 199, 207, 217
- application menu 56, 58–9
- `appuifw` module 27–8, 31–2
  - `app` object 53–5
  - `multi_query()` function 36–8
- `multi_selection_list()` function 43–4
- `note()` function 35–6
- `popup_menu()` function 40–1
- `query()` function 33–5
- `selection_list()` function 41–3
- `uid()` function 283
- ArduinoBT micro-controller board 261–6
- asynchronous communication 188–92
- audio module
  - functions 83–4
  - `open()` function 78–9
  - `play()` function 78–9, 81–2, 83
  - `record()` function 81, 82, 83
  - `say()` function 78
  - `stop()` function 80, 81, 82, 83
- automatic updating 236–8
- binding
  - `bind()` function 54, 69, 88, 139
  - keycodes to callback functions 86–8
- `blit()` function 92, 95, 96, 104, 206
- Bluetooth
  - client–server chat application 138–41
  - connecting to external GPS reader 148–50
  - connecting to other devices 150–1
  - creating Bluetooth servers 144–6
  - Max/MSP connection using RFCOMM 266–71
  - serial port, setting up 289–92
  - using the Bluetooth console 292–3
- `break` statement 45
- bugs, finding 298–9
- callback functions 54–5, 69, 79
  - binding to keycodes 86–8
  - capturing key events 91–2
  - `event_callback()` function 88–91
  - `redraw_callback()` function 92–5
- camera
  - functions 100–1
  - taking a photo 102–4
  - viewfinder 101–2

- Canvas object 85–6
  - bind() function 54, 69, 88, 139
  - blit() function 92, 95, 96, 104, 206
  - double buffering 106
  - event\_callback parameter 88–9
  - redraw\_callback parameter 92, 94, 95
  - size() function 94–5
- capabilities, Platform Security 280–2
- certificates *see* devcerts (developer certificates)
- C++ extensions 236
- chat application 138–41
- client–server applications
  - MobiLenin 245–52
  - voting service 179–82
- code blocks, writing 28–9
- coding styles 241–3
- color
  - hexadecimal constants 94
  - specified as Unicode strings 42
- communication protocols 166–7
  - default access point, setting 172
  - HTTP client 169–70
  - JSON client 170–1
  - TCP client 167–9
  - TCP/IP 159
- concurrent programs 241
- connect() function 139, 168, 190, 269
- contacts module 151–2
- Content\_handler object 60, 157, 213
- content() function 65–6
- continuation lines 29
- conversions
  - plain to Unicode text 121
  - type 36, 60, 63, 64, 72, 140, 144, 203
- custom modules 234–6
- databases
  - contacts database 151–2
  - Eventfu application 207–15
    - local database 121–3
    - MySQL database 261
- data encoding using JSON 166–7, 170–1, 174
- data handling 111–12
  - basic file operations 114–16
  - dictionary data structure 118–20
  - error handling 113–14
  - file organization 112–13
  - finding sound, photo and video files 116–17
  - local database 121–3
  - log files 116
  - reading and writing text 117–18
  - reading and writing Unicode text 120–1
- debugging 295
  - interpreting tracebacks 295–7
  - logging output to file 116
  - procedure for 298–9
- decode("utf-8") function 121
- default access point, setting 172
- def keyword 51
- deployment of PyS60 programs 287
- devcerts (developer certificates)
  - capabilities 280, 281–2, 283
  - obtaining 285–6
  - signing with 284, 285
- dial() function 151, 152
- dialog functions 27–8
  - multi-query dialog 36–8
  - note dialog 35–6
  - single-field dialog, query 33–5
- dictionary object 118–19
  - constructing 230–1
  - event object 88–9
  - JSON client 170–1, 174
  - JSON gateway 185, 186
  - and local databases 121–3
  - reading contents from a file 120
  - writing contents to a file 119–20
- directories 112–13
  - creating 113
  - protected 282–3
- dot notation 56
- double buffering 92, 106
- download\_plugin() function 238–9
- drawing functions 95
- drive letters 113
- dynamic time 105–6
- e32dbm module 122–3
- e32db module 121
  - lock object 55, 56, 69, 79, 105
  - sleep() function 105–6, 110, 182
  - timer object 97, 125–6, 211–13, 214–15, 240
  - yield() function 99, 106
- empty value 36
- emulator, using 30, 301–2
- encoding
  - Unicode strings 34–5, 121, 129, 211, 213
  - urllib.urlencode() function 204
  - using JSON 166–7, 170–1, 174
- errors
  - see also* debugging
  - during installation 29–30
  - exception handling 113–14, 296–7
- event\_callback() function 88
- event dictionary 88–9
- Eventfu application 207–9
  - access point dialog 214
  - constants, setting up 209–10
  - description of event 214
  - event form 208, 213
  - preferences form 208, 209–10
  - storing preferences 210–11
  - UI functions 214–15
  - updating events 211–13
- event handling functions 91

- event loops 105
  - events database *see* Eventfu application
  - exception handling 113–14, 296–7
  
  - File object 114–16
  - file organization 112–13
  - file system protection 282–3
  - find() function 61, 62, 66
  - for loop 44–5
  - flickering, reducing by double buffering 106
  - Flickr *see* InstaFlickr application
  - functions, creating own 49–52
  
  - games
    - controlling with event loops 105
    - double buffering 106
    - dynamic time 105–6
    - guess my number 145–6
    - Hangman server 70–6
    - Manhattan Story Mashup 252–6
    - random number generation 106
    - structure of 104–6
    - UFO Zapper 104, 106–10
  - glcanvas and gles modules, 3D graphics 99
  - global variables 72–3, 83, 231, 235–6
  - GPS positioning
    - external GPS over Bluetooth 148–50
    - using position module 127
  - graphical user interface
    - customizing, Max/MSP connection 268–9
    - native elements of PyS60 31–45
  - graphics 92
    - 3-dimensional 99–100
    - drawing graphics primitives 92–5
  - interactive 97–9
  - GSM positioning 123–6
  
  - hang\_up() function 151
  - Hangman server game 70–6
  - “Hello World” script, writing
    - 2nd Edition devices 25–7
    - 3rd Edition devices
      - Linux Users 20–1
      - MAC OS X Users 19
      - Windows Users 15–18
  - httplib module 157
  - HTTP server 174–7
  - HyperTerminal, Windows 292–3
  
  - if statement 38–9
  - image composition tool 256–61
  - image files
    - see also* photos
    - reading 117
    - uploading to Web 223, 259–61
  - image masks 96
  - Image object 92, 106
    - blit() function 92, 95, 96, 104, 206
  - drawing graphics primitives 92–5
  - loading and saving images 96
  - map images 203–5
  - MobileArtBlog 257–60
  - taking screenshots 96–7
  - viewfinder images 101–2
- importing modules
    - \_import\_() function 239
    - import statement 32
  - Inbox object 64–5
    - accessing messages 65
    - bind() function 69
    - forwarding messages 69–70
    - receiving messages 68–9
    - searching for messages 66
    - sms\_messages() function 65, 66, 67, 228, 229
    - sorting messages 66–8
  - indentation of code 28–9
  - innovation 3
  - democratizing 8–11
    - sharing of 9–10
    - user-centered 9
  - input verification 62–3
  - InstaFlickr application 215–16
    - constants 216–17
    - data uploading 221–2
    - progress bar 223–4
    - result parsing 217
    - signed calls 219–21
    - taking photos 222–3
    - token handling 217–19
    - UI functions 224
  - installation of PyS60 13–14
    - 2nd Edition devices 21
      - downloading install files 21–2, 301–2
      - installing files to phone 25
      - sending files to phone 22–5
      - writing and running a script 25–7
    - 3rd Edition devices
      - downloading install files 14–15, 301–2
      - Linux users 19–21
      - Mac OS X users 18–19
      - Windows Users 15–18
  - instant messaging 188–92
  - interactive graphics 97–9
  - Internet 159
    - see also* web services
    - automatic updating 236–8
    - communication protocols 166–72
    - connecting to 158–65
    - downloading data from 156–7, 238–9
    - plug-in mechanisms 238–9
    - uploading data to 157–8, 221–2
  - interpreter for Python 7–8
    - downloading and installing 301–2
    - signing with a devcert 285
  - introspection 231–4
  - IP addresses 159
    - local IP address 162
    - phone’s IP address 178
    - finding server 163

- join() function 72
- JSON (JavaScript Object Notation)
  - 166–7
  - installing JSON module 159–60
  - JSON client 170–1
  - JSON gateway 184–8
  - JSON server 173–4
- keyboard keys 84–6
  - binding keycodes to callback functions 86–8
  - capturing key events 91–2
  - event\_callback() function 88–9
  - key pressed or held down 89–91
- keycapture module 91–2
- keycodes 86–91
- key\_code module 86–91
- key events 50, 84–5
  - capturing 50, 88–90, 91–2
  - handling 85–6, 88–9
- key–value pairs 88–9, 118–20
- language tool application 127–30
- lead users 5
  - development of products 10
  - innovation by 8–9
  - motivation of 9
- len() function 61
- Linux users
  - installing PyS60 files 19–20, 24–5
  - IP addresses, finding 162
  - RFCOMM setup 291–2
  - writing first script 20–1
- list comprehensions 228–30
- lists 39
  - list comprehensions 228–30
  - multi-selection list 43–4
  - selection list 41–3
  - tuples 56–7
- local database 121–3
- local variables 72–3
- location application 124–6
- lock object 55, 56, 69, 79, 105
- loops
  - for loop 44–5
  - while loop 45
- lower() function 62
- MAC OS X users
  - AppleScript, controlling applications with 146–8
  - installing PyS60 files 18, 23–4
  - IP addresses, finding 162
  - RFCOMM setup 290–1
  - writing first script 19
- makedirs() function 113
- makefile() function 140, 144, 168–9
- Manhattan Story Mashup 252–6
- map explorer application *see* MopyMaps! application
- masks, image 96
- Max/MSP, controlling with a phone 266
  - Bluetooth RFCOMM connection 266–71
  - WiFi connection 271–3
- menus
  - application menu 56, 58–9
  - popup menus 40–1
- messages
  - accessing 65
  - receiving 68–9
  - searching 66, 228–30
  - sending 45–7
  - sorting 66–8
- messaging module 45–7, 64, 70, 71–2, 74
- micro-controller board, connecting phone to 261–6
- MIDI files, playing 79–80
- missing values, denoting 36
- MobileArtBlog 256–7
  - client code 257–60
  - inserting data into MySQL database 261
  - server-side PHP script 260–1
- mobile networking 155–6
  - communication protocols 166–7
  - HTTP client 169–70
  - JSON client 170–1
    - setting default access point 172
  - TCP client 167–9
- development environment, setting up 158–62
- downloading from the Web 156–7
- networking environments 160–2
- peer-to-peer networking 183–8
- testing network connection 162–5
  - uploading to the Web 157–8
- MobiLenin system 245–6
  - mobile client code 249–52
  - system architecture 246–9
- Mobile Web Server (MWS) 193
- modules
  - creating custom 234–6
  - importing on the fly 238–9
  - using built-in 31–2
- MopyMaps! application 201–2
  - constants 202–3
  - fetching map images 203–5
  - result parsing 203
  - UI functions 205–7
- MP3 files, playing 78–9
- multi-query dialog 36–8
- multi-selection list 43–4
- multi-user applications 252, 273–4
- music video voting application 245–52
- MySQL database, inserting data into 261
- National Marine Electronics Association (NMEA) 148–9
- native UI elements 31
  - multi-query dialog 36–8
  - multi-selection list 43–4
  - note 35–6
  - popup menu 40–1
  - query 33–5
  - selection list 41–3

- networking environments 158, 160–1
  - finding local and server IP addresses 162–3
  - local wireless network 161
  - phone Internet access
    - and external test server 161–2, 163
    - and external web server 162, 163
    - and PC as a server 161, 163
  - setting default access point 172
  - testing connection using a test server 163–5
- Nokia phone models 302
- None, empty value 36
- note dialog 27–8, 35–6
  
- Object EXchange (OBEX) 134–8
- objects 55–6
- OpenGL graphics API 99–100
- OpenSound Control (OSC) 273–4
- open-source 5, 6, 7
- operating systems 6
- OSC *see* OpenSound Control
- OScmobile module 273–4
- os module
  - makedirs() function
  - path.exists() function 113
  - remove() function 81
  - system() function 147
- output formatting 63–4
  
- packages
  - creating standalone 286–7
  - running under Platform Security 287–8
  - signing 284–6
- parameters, function 28, 51
- pass statement 130
- pausing execution *see* sleep() function
- PC, controlling remotely 146–8
- PC to phone communication 141–8
  - peer-to-peer networking 183–4
    - instant messaging 199–91
    - JSON gateway 184–8
  - phone calls, recording 83
  - phone to PC communication 141–2
    - AppleScript, controlling applications with 146–8
    - communicating with the PC 142–4
    - PySerial, creating Bluetooth Servers with 144–6
  - phone to phone communication 136
    - using OBEX 136–8
    - using RFCOMM 138–41
  - phone providing a web service 193–7
  - phone-as-server, drawbacks of 177–9
- photos
  - InstaFlickr application 215–24
  - Manhattan Story Mashup 252–6
  - MobileArtBlog 256–61
  - sending 136–8
  - taking 102–4
- PHP script 158, 249, 252, 259–61
- placeholders 63
- plain text, conversion to Unicode 121
- Platform Security 279–80
  - capabilities 280–2
  - file system protection 282–3
  - running PyS60 under 287–8
  - SIS package signing 284–7
- play() function 78–9, 81–2, 83
- plug-in mechanism, automatic updating 238–9
- popup menus 40–1
- popup notes 27–8, 35–6
- positioning 123
  - GPS positioning 127, 148–50
  - GSM cell ID mapper 123–6
- position module 127
- preferences, storing 210–11
- print statement 43
  - for writing to a file 115, 116
- private directory 116, 283
- program patterns 239–41
- prototyping with PyS60 1, 5, 7, 10–11
- proxy server, phone as 194–5
- py2sis tool 286–7
- PySerial module, creating Bluetooth servers 133–6
- Python programming language 7, 8
- Python for S60 (PyS60) 1–2, 7–8
  - see also* installation of PyS60
  - automatic updating 236–8
  - coding styles 241–3
  - deployment of 288
  - distribution of 288
  - extending using Symbian C++ 236
  - potential users 5–6
  - rapid prototyping tool 10–11
  - reasons for using 3–4
  - toolkit 10
  - writing first program 27–8
- query() function 27–8, 33–5
- quit() function 54
  
- random number functions 106, 108, 145
- range() function 44–5
- read() function 116, 117
- recording sounds 80–3
- redraw\_callback() function 86, 92, 206
- relational databases *see* databases
- remote control of PC 146–8
- replace() function 62
- resource directory 283
- return values, functions 51
- RFCOMM
  - phone-to-phone communication 138–41
  - serial port, setting up 289–92
- RGB colors 94

- robotics 274–7
- Roomba robotic vacuum cleaner 275–7
- S60 software platform 6–7, 302
- scancodes 86–7, 89
- scope of variables 72–3
- screenshots, taking 96–7
- screen size 59, 60
- security *see* Platform Security
- selection list 41–2
- self-signed certificates, creating 284
- serial communication, setting up 289–92
- Serial object 145, 146
- server software 172–3
  - HTTP server 174–7
  - JSON server 173–4
  - running on a phone 177–9
- shopping list assistant program 46–7
- signal() function 55
- signing *see* SIS package signing
- single-field dialog 33–5
- SIS package signing 284
  - developer certificates (devcerts) 285–6
  - keys and certificates, creating 284
  - signing PyS60 interpreter with a devcert 285
  - standalone packages, creating 286–7
- sleep() function 105–6, 110, 182
- smartphones 3
- sms\_messages() function 65, 66, 67, 228, 229
- SMS game server application 70–6
- SMS messages
  - Hangman game application 70–6
  - sending messages 45–7
  - SMS inbox 64–5
    - accessing 65
    - forwarding messages 69–70
    - receiving messages 68–9
    - searching 66
    - sorting 66–8
- socket module 172, 177–8
- SocketServer module 163–5, 174, 185–6
- sort() function 66–8
- sound
  - playing MIDI files 79–80
  - playing MP3 files 78–9
  - reading files 117
  - recording 80–3, 129, 151
  - text-to-speech functionality 77–8
- source code, sharing of 9–10
- split() function 63, 64
- standalone programs, creating 286–7
- start\_viewfinder() function 102
- startswith() function 62
- state() function, audio module 84
- str() function 36
- string handling
  - accessing parts of a string 61
  - cleaning up input strings 62–3
  - decision-making functions 61–2
  - defining strings 60–1
  - formatting output 63–4
- strip() function 62
- substrings 61–2
- Symbian C++, extending Python 236
- Symbian OS 6–7
  - Platform Security 279–80
  - SIS file signing 284–6
- symbol tables 231–4
- synchronous communication 138
- sysinfo module 152–3
- system directory 282–3
- system information 152–3
- tabs, defining 59
- take\_photo() function 102–3, 128, 137, 157, 196, 222–3
- TCP/IP 159, 166–7
  - Max/MSP 271–3
  - TCP client 167–9
- telephone module 151–2
- terminal emulator software, using 144, 301–2
- terminology 7–8
- text editors 15, 19, 20, 25, 29
- text reading/writing
  - dictionary key–value pairs 118–20
  - list items 117–18
  - Unicode strings 120–1
- text-to-speech functionality 77–8
- threading 185–92, 212, 241
- time
  - current time 126, 171, 213
  - dynamic time 105–6
  - time() function 68
- timer object 97, 126, 211–13, 214–15, 240
- toolkit for PyS60 10
- tracebacks, interpreting 295–7
- translation application 127–30
- troubleshooting
  - see also* debugging
  - installation problems 29–30
- try–except block 113–14
- tuples 56–9
- type conversions 36, 60, 63, 64, 72, 140, 144
- type parameters 33–4
- UFO Zapper game 106–10
- Unicode strings
  - encoding and decoding 34–5, 121, 129, 211, 213
  - reading and writing 120–1
- unread() function 68
- upper() function 62
- urban storytelling game 252–6
- urllib module 156–7, 170, 171, 200, 204
- user interface (UI)
  - see also* appuifw module
  - custom elements 254–6, 268–9

- native elements 31–45
  - structure of 52–3
- UTF-8 encoding/decoding 121, 129, 211, 213
  
- vacuum cleaner robot 275–7
- variables 36–3, 72–3
- video files, reading 117
- viewfinder 101–2, 103
- Vocabulector language-learning tool 127
  - adding new entries 127–9
  - boilerplate text 130–1
  - displaying entries 129–30
- von Hippel, Eric 8–10
- voter application 63–4
- voting service, client–server 179–82
  
- `wait()` function 55
- WAV files, recording 80–3
- web server, using phone as 193–7
- web services (Web APIs) 199
  - see also* Internet
  - application keys 199, 207, 217
  - EventFu event finder 207–15
  - InstaFlickr photo uploader 215–24
  - MopyMaps! mobile map explorer 201–7
  - Representational State Transfer (REST) 200–1
  - using Web APIs 200
- `while` loop 45
- white space 28–9
- WiFi
  - connecting phone to Max/MSP 271–3
  
- instant messenger application 188–92
  - security risks 220
  - testing connection to wireless network 162
- Windows users
  - installing PyS60 files 15, 22–3
  - IP addresses, finding 162
  - RFCOMM serial port, setting up 290
  - writing first script 15–18
- `write()` function 115
  
- XML parsers 203, 216–17
  
- Yahoo! Maps 201–7
- `yield()` function 99, 106