

- += 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)
 - Web APIs 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
- bind() function 54, 69, 88, 139
- binding 54
 - 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 100
 - 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
 - re-draw_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 118–19
 - reading from and writing to 119–20
 - error handling 113–14
 - file organization 112–13
 - finding sound, photo and video files 116–17
 - local database 121–3
 - log files 116
 - reading/writing text 117–18
 - Unicode text 120–1
- debugging 295
 - interpreting tracebacks 295–7

- logging output to file 116
 - procedure for 298–9
- decode UTF-8 statement 121
- default access point, setting 172
- def keyword 51
- deployment of PyS60 programs 288
- devcerts (developer certificates)
 - capabilities 280, 281–2, 283
 - obtaining 285–6
 - signing with 284, 286
- 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
- e32 module 54
 - 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
- gcanvas and gles modules, £D 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
 - customising, 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