

Index

Symbols

: (colon) 12
' (single quotes in strings) 21
" (double quotes in strings) 21, 275
() (function arguments) 30, 126
_ (underscore in variables) 20
* (all) 170, 172
""" (triple-quoted string) 91
/ (forward slash) 294
\ (backslash) 109, 275, 277, 294
\n (newline) 297
& (before hot keys) 267
(with comments) 90, 92
% (percentage sign) 32, 277, 281
%d (integer in format string) 279
%E, %e (E-notation in format string) 281
%F, %f (Float in format string) 278, 280
 negative numbers 280
 positive numbers 280
%G, %g (choice of float or E-notation in format string) 281
%i (integer in format string) 278–279
%s (string in format string) 278
>> (redirecting output) 303
>>> (Python prompt) 3, 16

Math and comparison operators

- (minus/subtraction) 27
-= (decrement) 33
< (less than) 66, 71
<= (less than or equal to) 66, 71
<> (not equal to) 72
!= (not equal to) 66, 72
* (multiplication) 5
** (exponentiation) 31, 71
/ (division) 27
% (modulus) 71
+ (plus/addition) 27, 280, 282, 284
+= (increment) 33
= (assignment/equal) 16, 28, 65, 204
== (testing if equal) 65, 71
> (greater than) 66, 71
>= (greater than or equal to) 66, 71

Numerics

24-bit color 181
3-D graphics 360

A

A conditional or while loop 85
A loop using range() 78
A program for moving balls around using sprites 207
A program to print squares and cubes 276
A trickier version of blocks of stars 103
A variable nested loop 102
A very simple for loop 75
A well-connected sine wave 188
acceleration 337
access values in a table 128
activate a menu, ALT key 268
active window 177
addends 28
Adding an event handler for the Hello button 259
Adding an _init_() method 151
adding comments in Python 89
Adding menu event handlers 269
addition 5, 27
ALT key 268
and 69
animation 193
.app 291
append() 114, 118
appending to a file 296, 301
arguments 82, 134, 137
arithmetic 5, 26
arithmetic expression 17
assigning a name 16
assignment operator 28
asterisk, * 5
attributes
 change data 159
 view data 159
attributes of an object 147

B

background music 244
 repeating 245
backslash (line continuation character), \ 109, 294
backslash character, \ (for special printing codes) 275
BASIC 362
BBC BASIC 362
binary xiv, 40

- binary files 299–300
 - opening 300
 - rb 300
 - binary format 291
 - bits 181, 291
 - blit 191, 204, 220, 339
 - blitting 191
 - block of code 64
 - Blocks of stars with double-nested loops 103
 - body of the loop 76
 - Boolean
 - False 121
 - True 121
 - Boolean expression 121
 - bottom-up 326
 - Bouncing a beach ball 197
 - in 2-D 197
 - Bouncing ball program, with sprites and
 - Clock.tick() 219
 - Bouncing ball with up and down arrow keys 221
 - break 86
 - bugs 28
 - button 256
 - Cancel 57
 - changing properties 256
 - button property position 260
 - buttonbox() 55
 - byte 181, 291
- C**
- C 359
 - C++ 359
 - Jcalling a function 131, 133
 - Cancel 57
 - card game, Crazy Eights 323
 - carriage return 274
 - case-sensitive 20, 24
 - chaining operators 66
 - chevrons, >> 303
 - choice() 314, 320
 - choicebox() 56
 - class 149, 153
 - Clock 212
 - date 344
 - derived 161
 - families 160
 - group 208
 - inheritance 160
 - naming convention 156
 - polymorphism 160
 - Sprite 203
 - subclass 161
 - time 344
 - timedelta 345
 - class keyword 258
 - Cleaning up our code 270
 - Cleaning up our code even more 271
 - closing a file, close() 298, 302
 - code 4
 - code stubs 162
 - coin flipping 318
 - collection 112
 - collision detection 208, 230
 - pixel-perfect 211
 - rect 211
 - colon 117
 - colon symbol, : 12, 64, 132
 - colors
 - 255 181
 - numbers 180
 - primary 180
 - RGB 180
 - combination 105
 - combine() 345
 - comma 46, 113, 138, 274
 - command shell 177
 - command-line 3
 - commenting out 92
 - comments 89
 - end of line 90
 - multiline 90
 - single line 90
 - common bug 24
 - comparison operator 66
 - Complete temperature-conversion program 264
 - component 256
 - controls 261
 - widgets 261
 - component property command 270
 - components, StaticText 262
 - computer address 181
 - computer animation 192
 - computer card games 323
 - computer generated
 - decks of cards 313
 - dice 313
 - random events 313
 - computer memory 15
 - read from 15
 - write to 15
 - computer model 336
 - computer punch card 300
 - computer simulation 336
 - concatenation 21, 275, 284
 - condition 62–63
 - more than one 69
 - conditional loop 74, 84
 - Connect-the-dots mystery picture 189
 - console 3
 - constants 101
 - continue 85
 - controls 261

- converting data 38
- Converting temperatures 47
- copy a list 125
- copying pixels 191
- counter 318
- counting loop 75, 77
- CR 274
- creating a function 131
- Creating a module 165
- Creating a simple Ball class 149
- Creating and using a function 132
- Creating and using a function that returns a value 140
- Ctrl-Break 177
- CTRL-C 76, 177
- CTRL-SHIFT-R 54, 216
- current time 343
- timestamp 343
- current working directory 294

D

- data structure 127
- date class 344
- datetime object 343
 - parameters 344
- datetime.combine() 345
- debugging 265
- debugging a program 92
- decimal numbers 27
- decision point 106
- decision tree 106
 - decision point 106
- decisions 62
- deck of cards
 - drawing a card 319
 - memory 319
 - picking a card 320
- decrement a variable, -= 23, 33
- def 131
- default 58
- define a function 131
- del 120
- delta 343, 345
- dialog box 55, 254
 - input 55
- dice 314
 - different combinations 317
 - rolling two 315
- directories 292
 - subdirectories 293
- directory structure 292
- display surface 180
 - double-buffered 180
 - flip 179–180
- Displaying a beach ball image in the Pygame window 190

- Displaying what's in the player's hand 329
- division 27
 - modulus 32
 - quotient 31
 - remainder 31
- documentation 89
- Doing something different each time through the for loop 76
- dot notation 149
- double backslash, \\ 294
- double equal sign, == 65
- double star sign, ** 31
- double-buffered 180
- double-nested loop 103
 - down arrow, K_DOWN 221
 - dragging 224
- Drawing a circle 178
- Drawing curves using a lot of small rectangles 187
- dump() 304

E

- EasyGui 53, 175
 - buttonbox() 55
 - choicebox() 56
 - enterbox() 57
 - integerbox() 59
 - learning more 60
 - making the choice box smaller 57
 - message box 53
 - text input 57
- easygui.py 53
- elif 67, 99
- else 68
- endless loops 76
- end-of-line comments 90
- endswith() 285
- E-notation 33–34, 36, 305
- Enter key 3
- enterbox() 57
- entry box
 - spin box 262
 - Spinner 262
 - TextField 262
- equal sign, = 28, 65
- erase, paint over 193
- error 28
 - invalid literal 42
 - roundoff 40
 - syntax 9
- error handling 352
- error message 4, 10
- errors 9
 - handling 352
- evaluate an expression 17
- event 351
 - random 314

- event driven programs 217
- event handlers 218, 259, 351
 - on_ 259
 - parameters 259
- event loop 175, 218
- events 175
- event queue 218, 221
- events 175, 217–218, 259–260
 - event handlers 218
 - event loop 218
 - event queue 218, 221
 - key events 220
 - keyboard 219
 - KEYDOWN 219, 221
 - mouse 223
 - MOUSEBUTTONDOWN 223–224
 - MOUSEBUTTONUP 223–224
 - MOUSEMOTION 224, 229
 - pygame.event.get() 221
 - QUIT 220
 - timer 225
 - timer events 225
 - user event 225
- executable file, .exe 291
- execute 5
- exponents 31, 36
- extend() 118–119
- extension 291
 - .py 8
- F**
- False value 121, 215
- file 290–291
 - .app 291
 - .exe 291
 - .mp3 291
 - .py 291, 295
 - .txt 291
 - adding to 296
 - appending 296, 301
 - binary 299
 - closing 298, 302
 - creating 296
 - extension 291
 - filename 296
 - folders 292
 - load() 304
 - location 291
 - name 291
 - opening 296
 - path 293
 - pathname 293
 - pickle 303
 - print 303
 - properties 291
 - reading 297
 - readline() 298
 - readlines() 297
 - replacing 296
 - seek() 299
 - size 291
 - storing 291–292
 - text files 299
 - type 291
 - unpickling 304
 - writing 296, 300–301
- file locations 292
- file mode
 - a 301
 - r 301
 - w 301
- filenames 291, 296
- Final PyPong code 235
- First version of PyPong 230
- flip 179–180
- flipping a coin 314
- float 346
- float variable, %f 278
- float() 38, 47, 346
- float(raw_input()) 47
- floating-point formatting 279
- floating-point numbers 27
- floats 27, 59
- folder structure 292
- folders 292
 - branches 293
 - multi-level 292
 - root 293
 - subfolders 293
 - tree 293
- font object 232
- for 75
- for loop 75, 321
- force
 - gravity 337
 - net 338
- force of gravity 337
- format strings 277
 - %d 279
 - %E, %e 281
 - %F, %f 278, 280
 - %G, %g 281
 - %i 278–279
 - %s 278
- forward slash, / 27, 294
- frame 212
- frame rate 212–213
 - adjusting 214
 - scaling 214
- frames per second 212
 - limit 213

- function
 - buttonbox() 55
 - choice() 314, 320
 - choicebox() 56
 - Displaying a beach ball image in the Pygame window 190
 - enterbox() 57
 - float() 346
 - help() 60
 - image 190
 - integerbox() 59
 - msgbox() 54
 - pygame.draw.circle() 179
 - pygame.draw.lines() 187
 - pygame.draw.rect() 183
 - pygame.image.load() 191
 - randint() 314
 - range() 78
 - range() shortcut 80
 - remove() 320
 - sorted() 125
 - Surface.get at() 190
 - Surface.set at() 190
 - sys.exit() 221
- functions 38, 131, 146
 - arguments 82, 134, 137–138
 - calling 131, 133
 - code stubs 162
 - creating 131
 - defining 131
 - float() 38, 47
 - input() 48
 - int() 38, 47
 - list 138
 - methods 148
 - more than one argument 137
 - multiple arguments 138
 - multiple parameters 138
 - parameters 82, 137
 - passing arguments 134
 - raw_input() 44
 - result 139
 - return value 139
 - returning a value 139
 - str() 38
 - type() 41
- Functions with two arguments 138
- future versions of Python 48
- G**
- games, random 313
- Getting a string using raw_input() 45
- Getting input from a file on the Web 49
- Getting input using a choice box 56
- Getting input using an enter box 58
- Getting input using buttons 55
- Getting the new suit when the player plays an 8 331
- Getting the player's choice 330
- global 143
- global namespace 169
- global variables 141, 143
- Graphical User Interface 3, 52, 254, 327
- graphics 174
 - erasing 193, 208, 220
 - frame 212
 - frame rate 212
 - frames per second 212
 - painting over 193, 208, 220
 - render 232
- gravity simulation 337
- greater than sign, > 66
- group class 208
- GUI 3, 52, 254, 327
 - components 255
 - controls 255
 - Hangman 305
 - Virtual Pet 350
 - widgets 255
- GUI elements
 - dialog boxes 55, 254
- H**
- handling errors 352
- Hangman 305
- Hello World 3
- help() 60
- horizontal spacing 275
- Hot dog program with calorie counter 109
- hot key 267
- HotDog class with cook(), add_condiments(), and _str_() 157
- How to make default arguments 58
- HyperCard 254
- hyphen 27
- I**
- lines 286
- identifiers 363
- IDLE 2, 53, 335
 - colors 4, 91
 - text editor 7
- if 64, 69, 99
- images 190
- immutable variable 126
- import 166
- import a namespace 169
- importing a module 166
- importing EasyGui 53
- in 121, 286
- in the scope variables 141
- increment a variable, += 23, 33
- indenting 64–65
- indenting convention 65

- index 115
 - start from 0 115
- index number 115
- index() 122, 286
- indices 115
- infinite loops 76
- inheritance 160
- _init_() 151
- initialize 151, 339
- initializing the object 151
- input 14–15, 44, 52, 290, 296
 - events 217
- input from other sources 49
- input() 48
- insert() 118–119
- installing EasyGui 53
- installing Python 1–2
- instance 149
- instance reference 154
- instruction 3–4
- int() 38, 40, 47
- integer 27, 346
- integer formatting 279
- integer variable, %i 278
- integerbox() 59
- interactive 5
- interactive mode 19, 39
- invalid literal error 42
- isfile() 348
- iteration 76, 79, 100

J

- Java 359
- join() 284

K

- Keeping the Pygame window open 176
- key repeat 222
- keyboard event 219
- KEYDOWN 221
 - delay value 222
 - interval value 222
 - key repeat 222
 - multiple events 222
- keyword 5, 8
 - and 69
 - class 258
 - def 131
 - elif 67, 99
 - else 68
 - for 75
 - global 143
 - green 8
 - if 64, 69, 99
 - in 121, 286
 - not 70
 - or 70

- orange 8
- pass 162, 176
- print 17
- return 139
- sort() 123
- while 85
- Kids Programming Language 358
- KPL 358

L

- learning more about EasyGui 60
- less than sign, < 66
- LF 274
- line continuation character 109
- line feed 274
- list 56, 77, 112, 126, 138, 146, 320
 - adding to 113
 - append() 113, 118
 - contents 114
 - copy 125
 - creating 113
 - del 120
 - empty 113
 - extend() 119
 - in 121
 - index 115
 - index() 122
 - insert() 119
 - items 113
 - modifying 118
 - mutable 126
 - ordered 123
 - pop() 121
 - remove() 120
 - reverse() 123
 - searching 121
 - slice 117
 - slice notation 117
 - slice shortcut 117
 - slicing 116
 - sorted copy 125
 - sorting 123
 - sorting in reverse order 123
 - tuple 126
- list of operators 71
- lists of lists 126
- load() 304
- local namespace 169
- local variables 141
- logical expression 121
- logical operators 71
- Logo 360
- Looking for ten heads in a row 319
- loop 99
 - conditional 84
 - double-nested 103
 - event 218

- loop (*continued*)
 - for 75
 - nested 107
 - nested-nested 103
 - variable 101
 - variable name 79
 - variable nested 102
 - while 84
- looping 74
 - body 76
 - break 86
 - breaking out 86
 - condition 74, 84
 - continue 85
 - counting 75
 - endless 76
 - infinite 76
 - iteration 76, 79, 100
 - list of strings 84
 - naming convention 80
 - nested 99
 - runaway 76
 - skipping ahead 85
 - through a list 122
 - using i, j, and k 80
- lower() 288
- lowercase 288
- Lunar Lander 339
- M**
- Mac OS X, .app 291
- mailing list 359
 - Pygame 360
 - PythonCard 359
- Making a deck of cards 322
- making a list 77
- Making a Pygame window XE 175
- making the choice box smaller 57
- Making the Pygame window closeable 178
- math 26
- memory 15, 314
 - read from 15
 - write to 15
- memory management 140, 142
- menu 266
 - activate 268
- Menu Editor
 - New Menu 267
 - New Menu Item 268
- menu event, select 269
- methods
 - code stubs 162
 - combine() 345
 - inheritance 160
 - passing arguments 148
 - polymorphism 160
 - returning values 148
- methods of an object 147
- minus 27
- model 336
- Modern art with color 185
- modifying a list 118
- module 53, 165, 254, 322
 - import 166
 - random 172, 313
 - sleep() 171
 - sprite 202
 - time 171
- modules 131, 164
 - creating 165
 - importing 166, 169
 - namespace 167
 - Python Standard Library 170
 - using 166
- modulus, % 32, 279
- more than one argument 137
- mouse dragging 224
- mouse events 223
- move() parameter, speed 206
- Moving a beach ball image smoothly 195
- Moving a beach ball image with wrapping 198
- .mp3 291
- MP3 files 240
- msgbox() 54
- multi-level folders 292
- multiline comments 90
- multiple arguments 138
- multiple parameters 138
- multiplication 5
- music 241
 - background 244
 - fadeout() 250
 - get busy() 244
 - PyPong 250
 - repeat forever 245
 - repeating 245
 - set volume() 243
- Music and sound with volume adjustment 243
- mutable variable 126
- N**
- name 15–16
 - assigning 16
- namespaces 167, 233
 - global 169
 - local 169
- naming a program 15
- naming convention for loop variables 80
- naming variables 144
- nested loops 99, 107, 321
- nested-nested loop 103
- net force 338
- newline character, \n 274–275, 297, 299

- not 70
- not equal to sign, <>, != 66
- now() 345–346
- number sign, # 90, 92
- Number-guessing game 11
- Number-guessing game using EasyGui 59
- numbers 21
- O**
- object oriented 146
- object syntax 114
- objects 114, 131, 146
 - attributes 147
 - change attributes 159
 - class 149, 153
 - collect functions and data together 146
 - creating 149
 - creating an instance of 150
 - data hiding 159
 - derived class 161
 - __init__() 151
 - initializing 151
 - instance 149
 - instance reference 154
 - methods 147–148
 - properties 147
 - self 153
 - __str__() 152
 - subclass 161
 - variables 148
- Ogg Vorbis files 240
- on_ 259
- Open Graphics Language 359
- open source xvi
- open() 296
 - a argument 301
 - r argument 297, 301
 - w argument 301
- OpenGL 359
- Opening and reading from a file 297
- operands 28
- operating system, os 348
- operators 28, 66
 - chaining 66
 - equal 63
 - greater than 66
 - less than 66
 - list 71
 - logical 71
 - not equal 66
 - relational 66
- or 70
- order of operations 29
- os 348
- out-of-scope variables 142
- output 14, 52, 290
 - redirecting 303
- P**
- parameter 82, 137
 - reverse 123
- parentheses 30, 45, 48, 109, 121, 135
- part 297
- pass 162, 176
- Passing an argument to a function 135
- passing arguments 134
- path 293
 - pathname 293
- pathname 293
- percent 32
- percent sign, % 277, 281
- permutation 105
- Phrogram 358
- pickle 303, 348
 - dump() 304
 - load() 304
- picture element 179
- pixel 179
 - copying 191
- Playing music 242
- plus 27
- plus sign, + 282, 284
- polymorphism 160
- Pong 227
- pop() 121
- print 4, 17, 19, 40, 273
- print a \ 277
- print formatting 264, 273
 - automatic float 281
 - comma 274
 - concatenation 275
 - E-notation 281
 - floating point 280
 - integers 279
 - number formatting 278
 - pad 280
 - storing formatted numbers 282
- printing prompt messages 46
- Printing the 8 times table 77
- Printing the 8 times table up to 10 using range() 79
- Printing the loop variables in nested loops 104
- Printing three multiplication tables at once 100
- probability 313, 315, 317
 - memory 314, 319
- process 14, 52
- programming xiii
 - block of code 64
 - bottom-up 326
 - commenting out 92
 - comments 89
 - condition 62–63
 - conditional test 66
 - debugging 92, 265
 - decision tree 106

- programming (*continued*)
 - decisions 62
 - documentation 89
 - dot notation 149
 - functions 131
 - importance of comments 91
 - indenting 64–65
 - logical 66
 - looping 74
 - modules 131
 - objects 131, 146
 - question 65
 - statement 65
 - style 80, 91
 - test 62
 - top-down 326
- programming language
 - BASIC 362
 - BBC BASIC 362
 - C 359
 - C++ 359
 - Java 359
 - Kids Programming Language 358
 - KPL 358
 - Logo 360
 - Phrogram 358
 - QBASIC 362
 - Smalltalk 358
 - Squeak Etoys 358
- programs xiv, 7, 14
 - A conditional or while loop 85
 - A loop using range() 78
 - A program for moving balls around using sprites 207
 - A program to print squares and cubes 276
 - A trickier version of blocks of stars 103
 - A variable nested loop 102
 - A very simple for loop 75
 - A well-connected sine wave 188
 - Adding an event handler for the Hello button 259
 - Adding an `_init_()` method 151
 - Adding menu event handlers 269
 - Blocks of stars with double-nested loops 103
 - Bouncing a beach ball 197
 - Bouncing a beach ball in 2-D 197
 - Bouncing ball program, with sprites and `Clock.tick()` 219
 - Bouncing ball with up and down arrow keys 221
 - Cleaning up our code 270
 - Cleaning up our code even more 271
 - command-line 3
 - Complete temperature-conversion program 264
 - Connect-the-dots mystery picture 189
 - console 3
 - Converting temperatures 47
 - Creating a module 165
 - Creating a simple Ball class 149
 - Creating and using a function 132
 - Creating and using a function that returns a value 140
 - Displaying what's in the player's hand 329
 - Doing something different each time through the for loop 76
 - Drawing a circle 178
 - Drawing curves using a lot of small rectangles 187
 - event-driven 217
 - Final PyPong code 235
 - First version of PyPong 230
 - Functions with two arguments 138
 - Getting a string using `raw_input()` 45
 - Getting input from a file on the Web 49
 - Getting input using a choice box 56
 - Getting input using an enter box 58
 - Getting input using buttons 55
 - Getting the new suit when the player plays an 8 331
 - Getting the player's choice 330
 - GUI 3
 - Hello World 3
 - Hot dog program with calorie counter 109
 - HotDog class with `cook()`, `add_condiments()`, and `_str_()` 157
 - How to make default arguments 58
 - input 44, 52
 - Keeping the Pygame window open 176
 - Looking for ten heads in a row 319
 - Lunar Lander 339
 - Making a deck of cards 322
 - Making the Pygame window closeable 178
 - Modern art with color 185
 - Moving a beach ball image smoothly 195
 - Moving a beach ball image with wrapping 198
 - Music and sound with volume adjustment 243
 - naming 15
 - Number-guessing game 11
 - Number-guessing game using EasyGui 59
 - Opening and reading from a file 297
 - output 14, 52
 - Passing an argument to a function 135
 - Playing music 242
 - Printing the 8 times table 77
 - Printing the 8 times table up to 10 using `range()` 79
 - Printing the loop variables in nested loops 104
 - Printing three multiplication tables at once 100
 - process 14, 52
 - Putting the circle in the middle of the window 182
 - Putting your program to sleep 171
 - PyPong with sound and music 250
 - Ready for lift-off? 83

programs (*continued*)

- Rolling a single 11-sided die 1,000 times 315
 - Rolling two 6-sided dice 1,000 times 316
 - Sample run of Crazy Eights 328
 - Saving time to a file using pickle 348
 - Skier 95
 - Solving the hot dog program 107
 - Start of our hot dog program 107
 - text-mode 3
 - The Card class 321
 - The computer's turn 332
 - The main loop of Crazy Eights 325
 - The main loop with scoring added 333
 - The whole hangman.py program 309
 - Trying out sounds in Pygame 242
 - Trying to modify a global variable inside a function 143
 - Trying to move a beach ball 192
 - Trying to move a beach ball again 194
 - Trying to print a local variable 141
 - Unpickling using load() 305
 - Using a module 166
 - Using a sprite group instead of a list 209
 - Using a timer event to move the ball up and down 226
 - Using append mode 301
 - Using Clock and get_fps() in the beach ball program 214
 - Using draw.rect to make art 185
 - Using pickle to store a list to a file 304
 - Using readline() more than once 298
 - Using sprites to put multiple ball images on the screen 204
 - Using `_str_()` to change how the object prints 153
 - Using the Ball class 150
 - Using the comparison operators 67
 - Using write mode on a new file 302
 - Using write mode on an existing file 302
 - VirtualPet.py 352
 - Waiting for the end of the song 244
 - Who's the coolest of them all? 84
- prompt, >>> 3, 16
- messages, printing 46
- properties 147
- Property Editor (PythonCard) 255, 266
 - Property Editor properties
 - max 266
 - min 266
- public license 359
- Putting the circle in the middle of the window 182
 - Putting your program to sleep 171
 - .py 8, 291, 295
 - Pygame 95, 174, 338, 360
 - animation 193
 - arguments 184
 - blit 191
 - clock.get_fps() 213
 - clock.tick() 213
 - collision detection 208, 230
 - color names 181
 - colordict.py 181
 - colormixer.py 181
 - colors 180
 - display 178
 - displaying text 231
 - down arrow 221
 - draw 183
 - draw.rect() 339
 - drawing shapes 178
 - event loop 175, 218
 - events list 223
 - flip() 178
 - font 231
 - font object 232, 339
 - frame rate 213
 - grouping sprites 208
 - images 190
 - key names list 223
 - line width 184
 - list of events 223
 - list of key names 223
 - location 182
 - mailing list 360
 - pixel-perfect 211
 - pygame.event 220
 - pygame.font 231
 - pygame.mixer 240
 - pygame.time.Clock() 213
 - PyPong 227
 - rect 183
 - size 183
 - spritecollide() 210–211
 - sprites 202
 - surfaces 180
 - sys.exit() 178
 - time 212
 - time.delay() 212
 - up arrow 221
 - user-defined events 225
 - pygame window 177
 - pygame.draw.circle() 179
 - color 179
 - location 179
 - size 179
 - surface 179
 - width of the line 179
 - pygame.draw.lines() 187
 - closed 188
 - color 187
 - list 188
 - surface 187
 - width 188

- pygame.draw.rect() 183
- pygame.font 231
- pygame.image.load() 191
- pygame.mixer 240
 - initialize 241
 - music 241
 - Sound 241
- PyLogo 360
- PyPong 227
 - adding music 250
 - adding sounds 245
- PyPong with sound and music 250
- Python 1, 359
 - .py extension 8
 - 3-D graphics 360
 - adding comments 89
 - adding newlines 275
 - assigning a name 16
 - block of code 64
 - built-in functions 38, 45
 - class-naming convention 156
 - datetime 343
 - double star (asterisk) 31
 - EasyGui 53
 - error handling 352
 - file object 296
 - first program 7
 - future versions of 48
 - Global Module Index 359
 - help() 60
 - immutable 126
 - indenting 65
 - indenting convention 65
 - installing 1–2
 - integer-division behavior 28
 - Language Reference 359
 - Library Reference 359
 - list 112
 - making a list 77
 - memory management 140, 142
 - module 53, 95, 164–165
 - multiplication 5
 - name 16
 - naming a program 15
 - newline character 274
 - Numeric module 174
 - object oriented 146
 - objects 114, 146, 282
 - online documentation 359
 - order of operations 29
 - os 348
 - pad 280
 - pickle 303
 - print 273
 - prompt 3
 - PyLogo 360
 - PyOpenGL 359
 - Python Standard Library 170
 - PythonCard 254
 - Pywinauto 361
 - randint() 314
 - retrieving information 17
 - running your program 8
 - RUR-PLE 360
 - shell 3
 - spacing 46
 - SPE 54
 - special methods 152
 - standard modules 170
 - Stani's Python Editor 54
 - start using 2
 - stopping a program 76
 - strings 282
 - syntax 205
 - testing 62
 - try-except 351
 - turtle 360
 - Tutorial 359
 - using IDLE 2
 - using quotation marks 16, 21
 - using variables 22
 - van Rossum, Guido 359
 - variable names 20
 - version 2
 - VPython 360
 - win32com 361
 - Windows 361
 - Python file, .py 291, 295
 - Python Standard Library 170
 - PythonCard 254, 305
 - Background 259
 - button 256
 - command 270
 - component 256, 350
 - dialog box 356
 - event handlers 259
 - events 260
 - Gauge 350
 - GUI 350
 - hot key 267
 - Image 350
 - ImageButton 350
 - list 260
 - mailing list 359
 - menu 266
 - Menu Editor 266
 - menu event handlers 269
 - NewButton 256
 - object 258
 - Property Editor 255, 266
 - Resource Editor 255, 262, 266, 311
 - resource file 256

PythonCard (*continued*)

- self 259
- StaticText 350
- timer 351
- wxPython 255

Pywinauto 361

Q

- QBASIC 362
- queue 218
- QUIT 220
- quotation marks, ' ' 16, 21
- quotes 21, 41
- quotient 31

R

- randint() 314
- random 172, 313
 - randint() 172
- random event 314
- random games 313
- random.choice() 307
- range() 78, 82
 - argument 82
 - shortcut 80
 - third parameter 82
- raw_input() 44, 56
 - getting a number 47
 - getting a string 45
- read 290
- read from memory 15
- reading a file 296
- reading in binary mode 300
- readline() 298
- readlines() 297
- Ready for lift-off? 83
- real numbers 27
- real time 342
- rect method, move() 206
- redirecting output 303
- relational operators 66
- remainder 31
- remove() 120, 320
- render 232
- reserved word 5
- resolution 180
- Resource Editor 311
- Resource Editor (PythonCard) 255, 262, 266
- resource file 256
- RESTART 9
- retrieving information 17
- return 139
- Return key 3
- returning a value 139
- reverse parameter 124
- reverse() 123
- RGB 180

- robotics 358
- Rolling a single 11-sided die 1,000 times 315
- Rolling two 6-sided dice 1,000 times 316
- rolling two dice 315
- round brackets 30, 45, 48, 119, 126
- rounded down 48
- rounding 279
- roundoff error 40, 305
- run 8
- Run in Terminal 175
- Run Module 8
- Run without arguments 216
- runaway loops 76
- running your program 8
- runtime error 10
- RUR-PLE 360

S

- Sample run of Crazy Eights 328
- Saving time to a file using pickle 348
- scientific notation 34
- scope of a variable 140
- screen coordinates 182
 - horizontal 182
 - vertical 182
 - x-axis 182
 - y-axis 182
- seek() 299
- self 153, 259
- shell 3
 - IDLE 3
- sign
 - 27
 - != 66
 - * 5
 - ** 31
 - / 27
 - # 90, 92
 - + 27
 - < 66
 - = 28, 65, 204
 - == 65
 - > 66
- simulation 336
 - gravity 337
 - time 342
- sine wave 186
- single-line comments 90
- Skier 95
- SkiFree 94
- sleep() 171
- slice notation 117, 124
- slice, shortcut 117
- slicing a list 116
- Smalltalk 358
- software xiv

- Solving the hot dog problem 107
 - sort() 123
 - sorted() 125
 - sound 239
 - .mp3 241
 - .ogg 241
 - .wav 241
 - .wma 241
 - generate 240
 - input 239
 - looping 245
 - MP3 files 240
 - music 241
 - Ogg Vorbis files 240
 - output 239
 - play back 240
 - playing 240
 - pygame.mixer 241
 - synthesize 240
 - volume 243
 - Wave files 240
 - WMA files 240
 - sound clips 241
 - sound effects 241
 - sound file, .mp3 291
 - sounds, PyPong 245
 - spacing in Python 46
 - spacing, horizontal 275
 - SPE 54, 175, 177, 335
 - speed 337
 - split marker 283
 - whitespace 283
 - split() 283 348
 - split marker 283
 - sprite 202
 - spritecollide() 210–211
 - sprite groups 208
 - sprite properties
 - image 203, 229
 - rect 203
 - spritecollide() 210–211
 - sprites 202, 338
 - square brackets 77, 113, 128
 - Squeak Etoys 358
 - standard modules 170
 - Stani's Python Editor 54
 - star, * 170, 172
 - Start of our hot dog program 156
 - startswith() 285
 - stopping a program 76
 - store a value 20
 - storing information 290
 - _str_() 153
 - str() 38
 - string 21, 42, 45, 82, 296
 - concatenation 21
 - substring 286
 - triple-quoted 22
 - string methods 282
 - endswith() 285
 - index() 286
 - join() 284
 - lower() 288
 - searching 284
 - split() 283
 - startswith() 285
 - strip() 287
 - upper() 288
 - string variable, %s 278
 - style 80, 91
 - subdirectories 293
 - subfolders 293
 - subscript 34
 - substring 286
 - subtraction 27
 - superscript 34
 - Surface.get at() 190
 - Surface.set at() 190
 - symbol 64
 - : 12, 132
 - & 267
 - % 32
 - syntax 9, 205
 - syntax error 9
 - sys.exit() 178
- T
- tab 108
 - tab stop 276
 - tab stop 276
 - table 126
 - accessing values 128
 - tabs 275
 - test 62
 - text editor 7
 - text file 299
 - text file, .txt 291
 - text-mode 3
 - The Card class 321
 - The computer's turn 332
 - The main loop of Crazy Eights 325
 - The main loop with scoring added 333
 - The whole hangman.py program 309
 - time 171
 - current 343
 - delta 343, 345
 - real 342
 - sleep() 171
 - time class 344
 - time simulation 342
 - time.delay() 212
 - timedelta 345–346

timer 225, 351
 timestamp 343
 top-down 326
 triple-quoted string 22, 91
 True value 121, 215
 try-except 351
 except block 352
 try block 352
 Trying out sounds in Pygame 242
 Trying to modify a global variable inside a
 function 143
 Trying to move a beach ball 192
 Trying to move a beach ball again 194
 Trying to print a local variable 141
 tuple 126
 turtle 360
 .txt 291
 type conversion 38
 float() 38
 int() 38
 str() 38
 type, tuple 126
 type() 41
 types of data 38

U

underscore character, `_` 20
 Unpickling using `load()` 305
 up arrow, `K_UP` 221
 upper() 288
 uppercase 288
 user event 225
 Using a module 166
 Using a sprite group instead of a list 209
 Using a timer event to move the ball up and
 down 226
 Using append mode 301
 Using `Clock` and `get_fps()` in the beach ball
 program 214
 Using `draw.rect` to make art 185
 using `i`, `j`, and `k` 80
 Using pickle to store a list to a file 304
 Using `readline()` more than once 298
 Using sprites to put multiple ball images on the
 screen 204
 Using `_str_()` to change how the object
 prints 153
 Using the Ball class 150
 Using the comparison operators 67
 Using write mode on a new file 302
 Using write mode on an existing file 302

V

validate 330
 validation code 330
 van Rossum, Guido 359

variable 20, 22, 24, 126
 count 108
 counter 318
 decrement 23, 33
 global 141, 143
 immutable 126
 in the scope 141
 increment 23, 33
 local 141
 mutable 126
 naming 144
 out of scope 142
 scope 140
 variable loops 101
 variable name
 case-sensitive 24
 rules 20, 24
 variable name, loop 79
 variable nested loop 102
 variables 113
 velocity 337
 Virtual Pet GUI 350
 VirtualPet.py 352
 volume control 243
 VPython 360

W

Waiting for the end of the song 244
 Wave files 240
 web browser 53
 while 85, 176
 while 1 215
 while loop 84, 99, 176
 event loop 218
 test 84
 whitespace 283, 287
 Who's the coolest of them all? 84
 whole number 39
 widgets 261
 win32com 361
 Windows Media Audio files 240
 WMA files 240
 working directory 294–295
 write 290
 write to memory 15
 writing to a file 296, 301, 304
 using print 303
 wxPython 255, 356

X

x-axis 182
 xturtle 360

Y

y-axis 182