Index

• Symbols and Numerics •

- (minus sign), 97, 98, 103
!= (not equal) operator, 98, 103
# (number sign), 74, 76
% operator, 98, 103
%= operator, 101, 103
& operator, 100, 103
( ) parentheses, 103, 106, 246
* (asterisk) multiplication operator, 98, 103, 236
variable argument lists, 111, 281
** operator, 98, 103
**= operator, 101, 103
*= operator, 101, 103
/ (forward slash), 98, 103, 295
// operator, 98
//= operator, 101, 103
: (colon), 106, 118, 125, 219
[ ] square brackets, 211, 226, 229
\ (backslash), 209–210, 295
^ operator, 100, 103
} curly brackets, 219
| operator, 100
~ operator, 97, 100, 103
+ (plus sign) addition operator, 98
concatenation using, 212, 236
operator precedence, 103
overloading, 283–284
as unary operator, 97
using indentation with, 72
using with tuples, 247
+= operator, 101, 103
< (less-than) operator, 99, 103
<< operator, 100, 103
<=(less-than or equal) operator, 99, 103
= (assignment) operator, 85, 101, 103
-= operator, 101, 103
== (equality) operator, 98, 103, 118
> (greater-than) operator, 98, 103
>= (greater-than or equal) operator, 99, 103
>> operator, 100, 103
" (double quotes), 74, 207, 209
' (single quote), 207, 209
3ds Max, 342

• A •

\a escape sequence, 210
Abaqus, 342
absolute paths, 295
accented characters, 209
accessors, 285
action warning level, 43
Add to Path option, 26
__add__() function, 246, 247, 283
Additional Help Sources feature, IDLE, 66
aggdraw library, 333
AIX (Advanced IBM Unix), 21
Alice Educational Software, 17
alignment, string, 220
American Standard Code for Information Interchange (ASCII), 206, 210
Amiga Research OS (AROS), 21
and operator, 99, 103
append() function, 193, 232, 257, 264, 305
appendleft() function, 264
Apple Siri, 7
Application System 400 (AS/400), 22
commands in, 68–69
commercial, written in Python, 18
compile time errors, 152
creating in Edit window, 67–68
CRUD and, 39
debugging, 353
decision-making and, 117
defined, 9
designing, 13–14
installing using PyInstaller, 350–351
loading in Edit window, 79
multithreaded, 261
overview, 12
procedures and, 10
purpose of, 13
quitting, 237
README files, 40
running from command line, 78
running from IDLE, 71–72, 79–80
runtime errors, 152
saving files for, 69–70
usage types, 16–17
apt-get command, 30–31
ArcGIS, 344
*args argument list, 281
arguments, command-line, 42
arguments, exception
listing, 163–164
overview, 161–163
arguments, function
accessing using keywords, 110
default values for, 110–111
overview, 108
position, 110
required, 108–110
variable number of, 111–112
arguments, method, 281–282
arithmetic operators
listing of, 97–98
precedence, 103

ArithmeticError exception, 167, 169
AROS (Amiga Research OS), 21
as clause, 162
as_string() function, 320
AS/400 (Application System 400), 22
ASCII (American Standard Code for Information Interchange), 206, 210
ASP.NET, 336
assignment operators
assigning value to variable, 85
listing of, 101
precedence, 103
asterisk (*)
multiplication operator, 98, 103, 236
variable argument lists, 111, 281
attributes, module, 184, 193–197
audio, 360–361

• B •
backslash (\), 209–210, 295
backspace character, 210
Base 2, 86
Base 8, 86
Base 10, 86
Base 16, 86
b command, 199
\b escape sequence, 210
-b option, 42
-\b option, 42
-\b option, 42, 44
BeOS, 22
bin() function, 86
binary codes, 12
binary operators, 96, 103
Binary to Decimal to Hexadecimal Converter, 100
BitBucket, 348
bitwise operators, 99–100, 103
Blender, 342
blue text in IDLE, 61
bool() function, 306
Boolean type, 89–90
break statements
  overview, 136–138
  for while statements, 144
bugs
  defined, 150
  tracking sites for, 348
  using virtual environments, 349
__builtins__ attribute, 194
byte code, 353
byte type, 42, 45
bytearray type, 42
-c option, 43

• C •
C#
  job opportunities and, 15
  Python versus, 19
  user interfaces, 17
__cached__ attribute, 194
caller, 105, 175–176
capitalization, 154, 171
capitalize() function, 213
car security systems, 345
Carnegie Mellon University, 17
carriage return character, 210
Cascading Style Sheets (CSS), 336
CASE (Computer Aided Software Engineering), 17
case sensitivity, 154
catching exceptions. See exceptions, handling
category warning level, 43
C/C++, 15, 340
clementtree library, 333
center() function, 213, 216
CentOS, 29
CGI (Common Gateway Interface), 332
characters
  ASCII, 206–207
  creating strings from, 207–208
escape sequences, 209–210
selecting individual in string, 211–213
sets of, 207
  special, 208–211
child classes, 287
Cinema 4D, 342
__class__ attribute, 271
classes
  built-in attributes, 271–272
  class suite, 270
  constructors, 275–277
  creating, 269–271
  creating external, 284–285
  explained, 268–269
  extending, 287–290
  importance of application organization, 267
  importing module for, 286
  inheritance, 287
  method arguments, 281–282
  methods, 273–275
  overloading operators, 282–284
  using external, 285–287
  variables, 277–280
clear() function, 232, 234, 241, 253
client (web) applications, 332
CMS (Content Management System), 332
code
  blocks of, 120–121
  cleaning using Isort, 355
  color coding, 61, 63–64
  commenting out, 75–77
  comments in, 74–75
  common mistakes, 336–337
  grouping into collections, 184–185
  highlighting, 63–64
  indentation, 72–73
  inspecting, 43
  introspection, 331
  optimizing, 43
  readability, 1, 15
  reusability, 104–105
code (continued)
runnable, 184
spaghetti code, 267
understandable, 93
using Edit window, 67–68
version control, 355–356
collections, 241, 243–244
colon ( : ), 106, 118, 125, 219
color coding, 61, 63–64
Comma Separated Values (CSV), 297
command-line Python. See also IDLE
accessing from command prompt, 34–35
advantages of, 40
arguments, 42
close button of terminal, 55
commands in, 46
Enter key in, 46
environment variables and, 44–45
exiting, 54–56
help mode, 48–49
IDLE versus, 58
options for, 42–44
running applications, 78
starting, 41
viewing result in, 46–47
comments
commenting out code, 75–77
multiline, 74–75
single-line, 74
uses for, 75
Common Gateway Interface (CGI), 332
communication
applications and, 9, 13
computers and, 7–8
exceptions and, 150–151
comparisons
function output, 114
if statements, 121–123
overview, 94–95
precedence, 103
compile time errors, 152, 154
complex numbers, 88

Computer Aided Software Engineering (CASE), 17
computers
characters and, 206–207
communication with, 7–8
comparisons and, 95
CRUD, 39
data storage, 84
exceptions, 150–151
lists and, 225–226
preciseness of, 11
procedures, 10–11
programming languages, 12
purpose of applications, 9
strings and, 205
concatenation
creating lists using, 236
defined, 72
using + operator, 212
using with tuples, 247
conditions for if statements, 118
configuration
environment variables, 44–45
IDLE, 63–66
console library, 333
constants, 193
constructors, 275–277
Content Management System (CMS), 332
Content-Type header, 320
Context-Transfer-Encoding header, 320
continue statements
overview, 138–139
pass clause versus, 140
for while statements, 144
control characters, 208–209, 296
control statements
if statements, 118–123
if...elif statements, 125–128
if...else statements, 124–125
nesting, 129–132
switch statement and, 128
copy() function, 232, 234
copyright() function, 48
copyright messages, 43
count() function, 217, 218, 244
Counter object, 240–242
Create, Read, Update, Delete.
See CRUD
credits() command, 48–49
cross-platform support, 19, 21–22
CRUD (Create, Read, Update, Delete) applications and, 83
defined, 39
file storage, 293–294
for lists, 232
CSS (Cascading Style Sheets), 336
CSV (Comma Separated Values), 297
curly brackets { }, 219
current directory, 191
-d option, 43, 44

D

data analysis, real-time, 346
data integrity, 294
data mining, 344–345
data storage
assigning values, 85
creating files, 298–301
deleting files, 308
file storage, 294–295
purpose of, 83
reading files, 301–303
structure of content, 295–298
variables, 84
writing data to files, 303–307
data types
Boolean, 89–90
complex numbers, 88
dates and times, 91–92
defined, 85
determining for variable, 90
floating-point values, 87–88
integers, 86–87
numeric types, 89
strings, 90–91
Database Administrator (DBA), 358
Database Management Systems (DBMSs), 332, 349, 358
databases, 16, 358–359
Datalist argument, 299
DataReader class, 302
DataWriter class, 299–300
dates and times, 45, 91–92
day value, 92
DBA (Database Administrator), 358
DBMSs (Database Management Systems), 332, 349, 358
debugging
defined, 150
starting debugger, 43
using pydbgr, 353
decryption, 358
default values for arguments, 110–111
del command, 253
deleting files, 308
delimiters, 214, 296
deque type
defined, 244
sequence types, 224
using, 263–265
development tools, 18
dictionaries
creating, 249
defined, 244
overview, 248–249
sequence types, 224
as switch statement, 253–256
using, 250–253
dir() function, 164, 193, 228, 271
directories, 294
division operator (/), 98, 103
doc() function, 198
__doc__ attribute, 194
documentation
- accessing from IDLE, 62–63
- in comments, 75
- creating using pdoc, 351–352
- online, 330
- opening pydoc application, 198–200
- quick-access links, 200–201
- searching, 202–204
  - .docx files, 296
- double quotes ("), 74, 207, 209
- downloading Python, 22–23
- drawing characters, 209
- dynamic systems, 88
- -E option, 43

- E -

  Edit window, IDLE, 67–68, 79
  effbot library, 333
  elementsoap library, 333
  elementtidy library, 333
  elementtree library, 333
  elif clause, 125–126, 237
  else clause
    - for if statements, 124–125
    - for loops, 141–142
    - try block and, 157
    - for while statements, 144
  email
    - creating HTML message, 324–325
    - creating text message, 323–324
    - envelope analogy, 311–313
    - host address, 313–314
    - hostname, 317–318
    - HowStuffWorks article, 310
    - letter analogy, 311–312, 318–319
    - MIME types, 319–321
    - ports, 312, 314–316
    - sending transmission, 321–322
    - SMTP, 309–310
    - subtypes, 322
    - viewing output, 325–326
  email.mime module, 319
  Embedded Python, 345
  empty() function, 261
  encryption, 358
  endless loops, 143
  endswith() function, 217
  engineering applications, 16, 88
  Enter key, 46
  enumerate() function, 281
  envelope analogy, 311, 312–313
  environment variables
    - ERRORLEVEL environment variable, 54
    - ignoring, 43
    - PATH environment variable, 26, 34–35
    - Python configuration, 35, 44–45, 191
  equality (==) operator, 98, 103, 118
  errno argument, 162
  ERRORLEVEL environment variable, 54
  errors. See also exceptions; exceptions,
    - handling
      - compile time, 152
      - handling, 149
      - logical, 154–155
      - runtime, 152–153
      - semantic, 154
      - syntactical, 154
      - types of, 153
  escape sequences, 209–210
  ETags, 364
  eval command, 353
  except clause
    - combining specific clauses with
      - generic, 167–170
    - defined, 157
    - listing exception arguments, 164
    - multiple clauses, 165–167
    - single clause, 164–165
    - using, 158–161
  Exception exception, 155
  exceptions. See also errors
    - arguments for, 161–163
    - built-in, 155
custom, 176–178
defined, 122, 149
listing arguments, 163–164
online resources, 331
raising, 174–175
exceptions, handling
except clause, 158–161
finally clause, 178–180
length checking, 137
multiple exceptions, 164–167
nesting, 170–173
passing error information to caller, 175–176
raising exceptions, 174–175
range checking, 123
single exception, 156–158
specific and unknown exceptions, 167–170
exec() command, 79
exemake library, 333
exit() command, 54–56
expandtabs() function, 213
exponents, 87, 89
expressions, 95, 113
extend() function, 232, 264
extending classes, 287–290
extendleft() function, 264
Extensible Markup Language (XML), 16, 296, 335–336
extensions, file, 294

\f escape sequence, 210
features, 13, 58
Fedora Core, 29
Fermilab, 17
fields, database, 359
FIFO (first in/first out), 244
file storage
creating files, 298–301
deleting files, 308
overview, 294–295
reading files, 301–303
structure of content, 295–298
supported file types, 299
writing data to files, 303–307
__file__ attribute, 194
FileNotFoundError exception, 155
fill character, 220
finally clause
exceptions and, 150
overview, 178–180
find() function, 217, 218
first in/first out (FIFO), 244
float() function, 90
float type, 87
floating-point values
formatting strings, 221
overview, 87–88
reasons for multiple numeric types, 89
flow control. See control statements
fluid dynamics, 88
flushing data, 300
folders, 294
Fonts/Tabs tab, IDLE, 63–64
for loops
break statements, 136–138
continue statements, 138–139
creating, 135
deque type and, 264
derange clause, 141–142
nesting, 145–147
pass clause, 140–141
for statement, 134
using with lists, 231
while statement versus, 144
format() function, 219–221
formfeed character, 210
forward slash (/), 98, 103, 295
freezing applications, 152, 261
from...import statements, 188–191
ftpparse library, 333
full() function, 261
function arguments
  default values, 110–111
  overview, 108
  required, 108–110
  using keywords, 110
  variable number of, 111–112
functions
calling, 107–108
code reusability and, 104–105
comparing output from, 114
defined, 104
defining, 105–107
overloading, 268
partial, 331
purpose of, 104
returning data from, 112–113
user input, 114–116
FUNCTIONS topic, 50

• G •
GCC (GNU Compiler Collection), 28
General tab, IDLE, 65–66
generators, 331
geocoding, 359
Geographic Information System (GIS), 344, 359
get() function, 241, 261
getaddrinfo() function, 315
__getattribute__() function, 164
gethostbyaddr() function, 313, 317
gethostbyname() function, 313, 317
gethostbyname() function, 317
gets() function, 316
getters/setters, 285, 297
GIMP, 342
GIS (Geographic Information System), 344, 359
Github, 348
GNU Compiler Collection (GCC), 28
Go.com, 17
Google, 17
Google App Engine, 342
Google Code, 348
Google Maps, 359
grabscreen library, 333
Graphic User Interface (GUI), 17, 359–360
graphs, 361–362
greater-than (>) operator, 98, 103
greater-than or equal (>=) operator, 99, 103
green text in IDLE, 61
GUI (Graphic User Interface), 17, 359–360
-h option, 43

• H •
handling exceptions. See exceptions, handling
headers, email, 311
help
  Additional Help Sources feature, 66
  command for, 48–49, 53–54
displaying, 43
help mode, 48–49, 50–52
in IDLE, 62–63
for specific commands or topics, 52–54
Hewlett-Packard Unix (HP-UX), 22
hex() function, 86
hexadecimal values, 86, 210
hierarchy of tuples, 247–248
highlighting code, 63–64, 334–335
horizontal tab character, 210
host address, 313–314
hostname, 317–318
Houdini, 342
hour value, 92
HP-UX (Hewlett-Packard Unix), 22
HTML (Hypertext Markup Language), 324–325, 336
-i option, 43, 45
IDE (Integrated Development Environment), 45, 334, 352
identity operators, 102, 103
IDLE (Interactive Development Environment). See also command-line Python
accessing on Mac, 36
accessing on Windows, 32–33
color coding in, 37, 61, 63–64
command-line Python versus, 58
commands in, 60
comments in, 74–77
configuration, 63–66
Edit window, 67–68
exiting, 80
feature overview, 58
help in, 62–63
indentation in, 72–73
overview, 58
Python versions and, 29
running applications from, 71–72, 79–80
saving files, 69–70
shortcut keys, 65
starting, 59
testing installation, 36–37
IETF (Internet Engineering Task Force), 349
if statements
code blocks for, 120–121
if...elif statements, 125–128
if...else statements, 124–125
multiple comparisons for, 121–123
nesting, 129–132
overview, 118
using relational operators, 119–120
IIS (Internet Information Server), 342
imaging library, 333
immutable types, 245
import statements
ignoring case in, 44
importing entire module, 187–188
importing only needed attributes, 188–191
overview, 183, 185–186
using, 162
in operator, 102, 103, 236
indentation, 63, 72–73
index
for dictionaries, 250
for lists, 229
for lists, negative, 230
for tuples, 247
index() function, 217, 244, 306
Industrial Light & Magic, 17
inheritance, 268, 287
__init__() constructor function,
275–276
initializing values, 275
__initializing__ attribute, 194
Inkscape, 342
input() function, 114–115
insert() function, 232, 233
insertion pointer, 208–209
inspecting code, 43
installing applications, 350–351
installing Python
on Linux, 30–32
on Mac, 27–29
testing installation, 36–38
on Windows, 25–27
instances
creating, 270
defined, 268
methods, 274–275
variables, 279–280
instantiation, 269
int() function, 90, 306
integers, 86–87, 220
Integrated Development Environment (IDE), 45, 334, 352
Interactive Development Environment. See IDLE
interactive environment, 354
Internet Engineering Task Force (IETF), 349
Internet Information Server (IIS), 342
IOError exception, 162
IPv4 (Internet Protocol version 4), 315
IPv6 (Internet Protocol version 6), 315
IPython, 354
IRLib library, 362–363
is not operator, 102, 103
is operator, 102, 103
isalnum() function, 213
isalpha() function, 213
isdecimal() function, 213
isdigit() function, 214
islower() function, 214
isnumeric() function, 214
Isort, 355
isspace() function, 214
istitle() function, 214
keywords topic, 50
Komodo Edit, 58, 352
**kwargs argument list, 281

• J •

j identifier, 88
Java
  development time, 15
  Python versus, 19
  using libraries in Python, 363
JavaScript, 16, 336
job opportunities
  data mining, 344–345
  embedded systems interaction, 345
  GIS, 344
  IT departments, 341–342
  network administration, 343
  programming languages and, 15
  QA, 340
  real-time data analysis, 346
  scientific tasks, 345–346
  specialty scripting, 342–343
  teaching, 343
join() function, 214
JPype library, 363
jQuery, 336
Jython, 363

• K •

KeyboardInterrupt exception, 155, 172
keys() function, 250–251
key/value pairs. See dictionaries
keywords topic, 50
Komodo Edit, 58, 352

• L •

Language Integrated Query (LINQ), 16
last in/first out (LIFO), 244
Launchpad, 348
Lawrence Livermore National Library, 17
learning curve, 15
LearnPython.org tutorial, 331
len() function, 214, 232, 253
length checking, 137
less-than ( < ) operator, 99, 103
less-than or equal ( <= ) operator, 99, 103
letter analogy, 311–312, 318–319
libraries
  defined, 183
  finding online, 357
  Google Maps, 359
  httplib2, 364
  IRLib, 362–363
  JPype, 363
  NumPy, 16
  PrettyTable, 360
  PyAudio, 360–361
  PyCrypto, 358
  PyQtGraph, 361–362
SciPy, 16
socket, 313, 315, 316, 317
SQLAlchemy, 358–359
third-party libraries, 332–333
TkInter, 359–360
Twisted Matrix, 364
license() command, 49
LIPO (last in/first out), 244
Lightwave, 342
linefeed character, 210
lineno warning level, 43
LINQ (Language Integrated Query), 16
Linux
accessing Python on, 36
installing Python, 30–32
Python support, 22
lists
accessing items in, 228–230
counter object for, 225–226
counter object for, 240–242
creating, 226–227
creating stacks using, 256–260
functions for, 228
looping through, 231
modifying items in, 232–235
mutable types, 245
negative indexes, 230
overview, 223–225
range of values in, 229
searching in, 236–238
sorting, 238–240
using operators with, 236
zero-based indexes, 229
ljust() function, 214
__loader__ attribute, 194
local hostname, 317–318
logical errors, 154–155
logical operators
listing of, 99
multiple comparisons for if statements, 121–123
preference, 103
loops
break statements, 136–138
continue statements, 138–139
deque type and, 264
defined, 269
defined, 135
else clause, 141–142
ends, 143
for loops, 134–135
nesting, 145–147
overview, 133–134
pass clause, 140–141
using with lists, 231
while statements, 143–145
lower() function, 214
lstrip() function, 214
-m option, 43

M

Mac OS X
accessing Python, 35–36
installing Python, 27–29
Python support, 22
mantissa, 89
mathematic applications, 16
max() function, 214, 216
Maya, 342
members, class, 268
membership operators, 102, 103
memory, and floating-point values, 89
MemoryError exception, 155
Mercurial version control, 355–356
message warning level, 43
methods
class, 273–274
defined, 269
instance, 274–275
instance variables and, 279
variable argument list for, 281–282
microsecond value, 92
Microsoft Disk Operating System (MS-DOS), 22
Microsoft Windows
accessing IDLE, 32–33
accessing Python from command prompt, 34–35
ignoring case in import statements, 44
installing Python, 25–27
opening pydoc application, 198
platform support, 22
MIME (Multipurpose Internet Mail Extensions), 319–321
min() function, 214
minus sign ( - ), 97, 98, 103
minute value, 92
Modo, 342
module warning level, 43
modules
defined, 183
finding on disk, 191–193
finding online, 357
from...import statements, 188–191
grouping code and, 184–185
ignoring paths for, 43
importing, 92, 185–188
numeric processing, 345
opening pydoc application, 198–200
quick-access documentation links, 200–201
running, 43
scientific, 345
searching documentation, 202–204
viewing attributes in, 193–197
modules topic, 50
month value, 92
MorphOS, 22
MotionBuilder, 342
MS-DOS (Microsoft Disk Operating System), 22
multiline comments, 74–75
multiplatform support, 19, 21–22
multiple processors, 245
multiplication operator ( * ), 98, 103, 236
Multipurpose Internet Mail Extensions (MIME), 319–321
multithreaded applications, 261
mutable types, 245, 248
\n escape sequence, 210, 215, 299
__name__ attribute, 194
NASA (National Space and Aeronautics Administration), 17
negation operator ( - ), 97
nesting
defined, 129
exception handling, 170–173
if statements, 129–132
loops, 145–147
network administration, 343
New York Stock Exchange, 17
newline attribute, 299
not equal ( != ) operator, 98, 103
not in operator, 102, 103
not operator, 99, 103
now() function, 92
Nuke, 342
number sign ( # ), 74, 76
numeric types
complex numbers, 88
floating-point values, 87–88
integers, 86–87
reasons for multiple, 89
NumPy library, 16, 362
-0 option, 43

ObjectDomain, 17
objects, 269
oct() function, 86
octal numeric values, 210
-oo option, 43, 45
open() function, 79, 162, 299
open source, 19
operands, 96
Operating System 2 (OS/2), 22
Operating System 390 (OS/390), 22
operators
  arithmetic, 97–98
  assignment, 101
  binary, 96
  bitwise, 99–100
  comparisons and, 95
  identity, 102
  logical, 99
  membership, 102
  overloading, 269, 282–284
  overview, 95–97
  precedence, 103
  relational, 98–99
  ternary, 96
  unary, 96, 97
  using with lists, 236
optimizing code, 43
or operator, 99
ord() function, 90
orphaned projects, 351
os._exit() command, 56
OS/2 (Operating System 2), 22
OS/390 (Operating System 390), 22
os.environ[ ] attributes, 192–193
os.pathsep constant, 193
os.remove() function, 308
os.rmdir() function, 308
overloading
  functions, 268
  operators, 269, 282–284

● p ●

__package__ attribute, 194
padding strings with zeroes, 215
Paint Shop Pro, 342
PalmOS, 22
parent classes, 287
parentheses ( ), 103, 106, 246
partial functions, 331
pass clause
  overview, 140–141
  for while statements, 144
PATH environment variable, 26, 34–35
paths, directory, 295
pdoc, 351–352
performance
  resources for, 338
  using virtual environments, 349
Perl, 20
PHP, 336
PIL (Python Imaging Library), 333
platform support, 21–22
Playstation, 22
plus sign ( + )
  addition operator, 98
  concatenation using, 212, 236
  operator precedence, 103
  overloading, 283–284
  as unary operator, 97
  using indentation with, 72
  using with tuples, 247
Pocket PC, 22
pop() function, 232, 234, 257, 264
POP3 (Post Office Protocol 3), 312
popleft() function, 264
ports, 314–316
positional arguments, 110
Post Office Protocol 3 (POP3), 312
precedence, operator, 103
precision of decimal number, 220
PrettyTable library, 360
print() function
  testing installation, 36–37
  typing commands, 46
  using in application, 68–69
  viewing command result, 46–47
procedures
  commands and, 46
  computers and, 10–11
procedures (continued)
defined, 9
separating from user interface, 304
tasks as, 9–10
processors, multiple, 245
production servers, 350
production-grade classes, 287
Program Files directory, 26
programming
application usage types, 16–17
code reusability, 104–105
common mistakes, 336–337
communication with computer, 94
exceptions and, 150–151
knowing multiple languages, 341
languages, 12, 14, 19–20
Python advantages, 15
protocol, defined, 310
prototypes, 16
Psion, 22
purple text in IDLE, 61
put() function, 261
.py files, 42
PyAudio library, 360–361
.pyco files, 42
PyCrypto library, 358
pydbgr, 353
pydoc application
opening, 198–200
quick-access links, 200–201
searching, 202–204
PyGame library, 361
PyInstaller, 350–351
PyOpenGL, 362
PyQtGraph library, 361–362
Python
advantages of, 15
applications written in, 18
C# versus, 19
documentation, 62–63
downloading, 22–23
Embedded Python, 345
environment variables for, 35
installing on Linux, 30–32
installing on Mac, 27–29
installing on Windows, 25–27
Java versus, 19
language comparisons online, 19
online documentation, 330
online tutorial, 331
organizations using, 17–18
Perl versus, 20
platform support, 21–22
popularity of, 15
reporting problems, 330
uses for, 16–17
using Java libraries in, 363
web programming using, 332
Python and XML Processing site, 336
python command, 78
Python GUI. See IDLE
Python Imaging Library (PIL), 333
PYTHONCASEOK environment variable, 35, 44
PYTHONDEBUG environment variable, 44
PYTHONDEFaulTHANDLER environment variable, 45
pythondoc library, 333
PythonEditors wiki, 334
PYTHONFAULTHANDLER environment variable, 35
PYTHONHASHSEED environment variable, 35, 45
PYTHONHOME environment variable, 35, 45
PYTHONINSPECT environment variable, 45
PYTHONIOENCODING environment variable, 35, 45
PYTHONNOUSERSITE environment variable, 35, 45
PYTHONOPTIMIZE environment variable, 45
**Index**

**PYTHONPATH** environment variable, 35, 45, 191
**PYTHONSTARTUP** environment variable, 35, 45
**PYTHONUNBUFFERED** environment variable, 45
**PYTHONVERBOSE** environment variable, 45
**PYTHONWARNINGS** environment variable, 45
**PYTHONWRITEBYTECODE** environment variable, 44
PyUnit, 354–355

• **Q**

q command, 199
-q option, 43
QA (Quality Assurance), 340
QNX, 22
quantum mechanics, 88
queue type
  defined, 244
  sequence types, 224
  using, 260–262
quit() command, 38, 54–55

• **R**

\r escape sequence, 210
raising exceptions. See also exceptions, handling
  defined, 150
  overview, 174–175
  passing error information to caller, 175–176
range checking, 121, 123
range of values in list, 229
Raspberry Pi, 343, 345
RDBMS (Relational Database Management System), 359
read() function, 79
readability of code, 1, 15
reading files, 301–303
README files, 40
real-time data analysis, 346
records, database, 358
Red Hat, 17, 29
Red Hat Package Manager (RPM), 29
regular expressions, 331
Relational Database Management System (RDBMS), 359
relational operators
  listing of, 98–99
  precedence, 103
  using with if statements, 119–120
relative paths, 295
remove() function, 193, 232, 234, 264, 306
repetition, 212
repetitive tasks. See loops
replace() function, 217
reporting problems, 330
required arguments, 108–110
resources
  common mistakes, 336–337
  IDEs, 334
  LearnPython.org tutorial, 331
  online documentation, 330
  performance, 338
  third-party libraries, 332–333
  Unicode characters, 337
  web programming, 332
  XML, 335–336
ResourceWarning exception, 155
returning data from functions, 112–113
reusable code, 104–105
reverse() function, 240
rfind() function, 217, 218
rindex() function, 217
RISC OS, 22
rjust() function, 214
rmtree() function, 308
Roundup Issue Tracker, 348
RPM (Red Hat Package Manager), 29
rstrip() function, 214
runnable code, 184
running applications
  from command line, 78
defined, 68
  from Edit window, 79
in IDLE, 71–72, 79–80
runtime errors, 152–153

- - S - -

-s option, 43, 45
-s option, 43
scientific applications, 16, 345–346
scientific notation, 87
SciPy library, 16, 362
screenshots in book, 32
Scribus, 342
SD (Secure Digital), 83
searching
  IRLib library, 362–363
  in lists, 236–238
  module documentation, 202–204
  in strings, 217–219
second value, 92
Secure Digital (SD), 83
seeding with random values, 45
selection tree, 129
self object, 274, 283
semantic errors, 154
sequences, 224–225, 243–244. See also lists
serialization, 331
Series 60, 22
server applications, 332
sets, 331
setters, 285, 297
shell, 54
shortcut keys for IDLE, 65
shutil.rmtree() function, 308
Simple Mail Transfer Protocol (SMTP), 309–310, 321–322, 324
Simple Object Access Protocol (SOAP), 333
single quote (‘), 207, 209
single-line comments, 74–75
__sizeof__ attribute, 194, 196
SMTP (Simple Mail Transfer Protocol), 309–310, 321–322, 324
smtplib module, 321
SOAP (Simple Object Access Protocol), 333
socket library, 313, 315, 316, 317
Softimage, 342
Solaris, 22
Solid State Drive (SSD), 293
sort() function, 239
sorting lists, 238–240
sound technologies, 361
spaghetti code, 267
special characters, 208–211
split() function, 193, 214, 216, 306
splitlines() function, 215
SQL (Structured Query Language), 16, 336, 359
SQLAlchemy library, 358–359
square brackets [], 211, 226, 229
squeeze library, 333
SSD (Solid State Drive), 293
stacks
  defined, 244
  sequence types, 224
  using, 256–260
startswith() function, 217
str() function, 42, 91
str type, 45
__str__() function, 284, 285, 297
strerror attribute, 162, 175–176, 177
strings
  creating from characters, 207–208
  as dictionary keys, 250
  formatting, 219–222
functions for, 213–216, 217
overview, 90–91
searching in, 217–219
selecting individual characters in,
  211–213
upper() function, 171
using special characters, 208–211
as viewed by computers, 206
strip() function, 215, 216
structured data, 295
Structured Query Language (SQL), 16,
  336, 359
subtraction operator (−), 98, 103
sudo command, 31
SUSE Linux, 29
swapcase() function, 215
switch statements, 128, 253–256
switches, command-line, 42–44
syntax
  concise, 1
  errors in, 154
  highlighting, 334–335
sys.exit() command, 56
sys.path variable, 43, 45, 192

T
\t (tab character), 210
Tcl (Tool Command Language), 360
ternary operator, 96
testing
  C++ applications, 340
  installation, 36–38
  production servers and, 350
  using PyUnit, 354–355
third-party libraries, 332–333
throwing exceptions, 150, 174–175. See
  also exceptions, handling
time() function, 92
TIOBE web site, 15
title() function, 215
TkInter library, 333, 359–360
TODO list management, 348
Tool Command Language (Tcl), 360
tools
  bug-tracking sites, 348
  IPython, 354
  Isort, 355
  Komodo Edit, 352
  Mercurial version control, 355–356
  pdoc, 351–352
  pydbgr, 353
  PyInstaller, 350–351
  PyUnit, 354–355
  Roundup Issue Tracker, 348–349
  VirtualEnv, 349–350
topics keyword, 50
traceback, 45
Trigger, 343
try block, 156, 164
tuples
  defined, 244
  hierarchy of, 247–248
  sequence types, 224
  using, 245–248
Twisted Matrix, 364
type() method, 90
typographical characters, 209

U
\u escape sequence, 210
-u option, 43, 45
UAC (User Access Control), 26
Ubuntu, 31
unary operators
  defined, 96
  listing of, 97
  precedence, 103
uncommenting lines, 77
Unicode characters, 210, 337
unit testing, 354–355
Universal Serial Bus (USB), 83, 293
unstructured data, 295
update() function, 241, 252
upper() function, 171, 215
USB (Universal Serial Bus), 83, 293
UsefulModules site, 357
User Access Control (UAC), 26
user input, 114–116
user interfaces, 17, 304

• V •
\v escape sequence, 210
-v option, 43
-v option, 43, 45
ValueError exception, 168, 177
variables
assigning values, 85
class, 268, 278–279
defined, 84
determining type of, 90
instance, 269, 279–280
returning data from functions, 113
verbose mode, 43
version control, 355–356
--version option, 43
vertical tab character, 210
VirtualEnv, 349–350
Visual Basic, 15
VMS (Virtual Memory System), 22
-w option, 43, 45

• W •
W3Schools site, 335–336
warning level, 43
web programming, 16, 332

while statements
nesting, 145–147
overview, 143–144
using, 144–145
whitespace, removing, 215
widgets library, 333
winsound module, 361
with statement, 299
writerow() function, 300
writing data to files, 303–307

• X •
\x escape sequence, 210
-x option, 44
-x option, 44
XML (Extensible Markup Language), 16,
296, 335–336

• Y •
Yahoo!, 18
year value, 92
Yellow Dog Linux, 29
YouTube, 18

• Z •
zero-based indexes, 229
ZeroDivisionError exception, 167, 169
zeroes, padding with, 215
zfill() function, 215
.zip files, 296
Zope, 18
z/OS, 22