

Index

- \ (backslash) for line continuation, 40, 149, 154, 159, 219, 324, 465
 - \ (backslash) for \n, \t, and other escape sequences, 49–51, 57, 206, 358, 392, 396, 404, 421
 - +, -, *, /, **, and % (in math operations), 38, 100, 105, 211, 258, 276, 424, 426, 487
 - + (plus sign) in concatenation, 43–45, 52–53, 60
 - * (asterisk) for import statements, 109
 - * (asterisk) for wild cards or placeholders, 191–193, 196, 219, 313–314, 516
 - / and \ (forward slash and double backslash) for file paths, 49–51
 - " or ' (double quotation mark or single quotation mark) for string literals, 39, 124, 151–152
 - " and ' (double quotation mark and single quotation mark) for sql queries, 150–154, 329
 - ! (exclamation point) for ArcGIS Python expressions, 100, 101, 207, 316, 477, 497
 - = (equals sign) for assignment statements, 26–27, 148
 - == (double equal signs), checking for equality in Python, 55, 146–148
 - != (exclamation point next to equals sign), checking for inequality in Python, 55, 146–148
 - # (hash sign) for comments, 23, 173
 - # (hash sign) as a default value, 465
 - '# (hash sign string) for tool parameter placeholder, 102–103
 - <> (less than sign next to greater than sign) for checking inequality in SQL, 150–151, 329
 - """ or ''' (triple double quotation marks or triple single quotation marks) for multi-line string literals, 39, 54
 - """ or ''' (triple quotation marks) for docstrings, 265, 267, 418
 - " vs. ' (double quotation makes versus single quotation marks) for script arguments, 124, 130
 - : (colon) to start code blocks, 143, 174, 175, 177, 243, 267, 359, 418
 - : (colon) in slicing, 42–43, 60, 112, 127–128
 - : (colon) in dictionaries, 336–337
 - { } (curly braces) for dictionaries, 336, 337
 - () (parentheses) for tuples, 64
 - () (parentheses) for function calls, 24, 262, 267
 - [] (square braces) for dictionaries, 337–341, 350
 - [] (square braces) for indexing, 41, 60
 - [] (square braces) for lists, 59, 60, 145, 337, 340
 - [] (square braces) for slicing, 42, 60
 - . (period)
 - for OOP dot notation, 45–46, 87, 97, 101, 155, 251, 296, 316, 394, 420, 428, 493
 - for OOP double-dot notation, 107, 125, 311, 507
 - ../ (dot dot slash) for relative path, 294
- A**
- Absolute path. *See* Full path file name
 - abspath, os.path function.
 See os module functions
 - Access by key, 337–338
 - activeView MapDocument property, 512–513

Add Field (Data Management) tool, 114, 207, 405, 477, 497

`addRow`, a `ValueTable` method, 89, 102

Aggregate Polygons (Cartography) tool, 92

`and`, Python keyword, 24, 146, 148, 151, 159

`append` list method, 61–62, 210, 293, 340

ArcCatalog preview, geography, 83–84

ArcCatalog preview, table, 83, 85

ArcCatalog tool search, 79–81

ArcGIS Python Window, 8–9, 16, 320, 510–511, 525

ArcGIS Resources site, 79–80

`arcpy` method, `setParameterAsText`, 466–467, 469, 471, 477–478, 497, 502

`arcpy` package, 84–87

ArcToolbox, 9, 77–78, 81, 95, 98, 249, 454, 466

Arguments, default, 270

Arguments, keyword, 439

Arguments, function, 24–25, 96, 266, 268–270

Arguments *vs.* parameters, 25

Arguments, positional, 439

Arguments, script, 20–21, 119–125, 161, 268–270

Array object, `arcpy`, 89, 325–328

`askdirectory` method, 438–440, 442–445

`askopenfile` method, 443–444

`askopenfilename` method, 437–438

`asksaveasfile` method, 443–444

`asksaveasfilename` method, 438, 442, 443

Assignment statement, 26–27

`AttributeError` exception, 31, 157, 159, 298, 317

`attrs` Tag property, 395–396

`author` MapDocument property, 512–513

B

Backslash. *See* \

`bandCount` Describe property, 155

Base name, 43

`baseName` Describe property, 155

`basename`, `os.path` function.
See `os` module functions

`BeautifulSoup` module, 393–398, 402–404, 406

Beginning and ending tags, 387

Binary file, 398–399

Block of code, 24, 143, 199, 262, 418

Block structure, 21, 23–24, 134, 137–139

Boolean expressions, 145–146, 172

Boolean operators, 145–146

Boolean type, 30

Branching, 133–135, 141, 338

Breakpoint conditions, 230, 233, 234

Breakpoints, 230–237

Buffer (Analysis) tool, 7–8, 78–79, 93, 96–98, 105, 197, 250

Bugs (coding errors), 200, 223, 283

Built-in constants, 30–31, 91, 145, 242, 274

Built-in exceptions, 30–31, 224, 245, 251

Built-in functions. *See* Functions, built-in

C

Calculate Field (Data Management) tool, 100–101, 113–114, 316, 477, 496–497

Calculate Value (Data Management) tool, 100

Calling functions, 25, 87, 261–265, 267, 271

Casting, 44, 53, 106, 147, 163, 227, 243, 292, 362, 366, 379, 405

`catalogPath` Describe property, 155

Catch exceptions, 244–245, 247–248, 252–255

Central Feature (Spatial Statistics) tool, 289

`CheckExtension` `arcpy` function, 88

`CheckOutExtension` `arcpy` function, 108, 110, 153, 174, 242, 494, 495

Check syntax, 65–66

`class`, Python keyword, 418–419

Class methods, 46, 61, 86, 418–420, 425, 430, 485

Class properties, 155–158, 484, 485, 493

Clip (Analysis) tool, 112–113

`close` file method, 358–359, 399, 444

Combo box, 99–100, 235, 458, 463, 472, 474–475, 497

Comment, in HTML code, 387

Comment, in Python code, 23

Comparison operators, 145–148, 151–152, 302, 328

Compressed file, 398, 400–401

`compression` Describe property, 155

Concatenating, 43–44, 52–53, 60, 126, 268, 454

Conditional constructs, 142–143, 159–161, 163, 165, 211, 218, 231, 255, 301, 338–339, 348, 350, 427

Conditional expressions, 145–150, 162, 300, 427

`contents` Tag method, 387, 395–396, 403–404, 406

Context highlighting, 21, 386

Context menus, 2, 48, 86, 115

Copy Features (Management) tool, 111

Copy (Management) tool, 126, 182, 207, 287, 331, 438, 466

`count` list method, 61–62

`count` string method, 51

Create File GDB (Data Management) tool, 166

Create Normal Raster (Spatial Analyst) tool, 174

CreateObject arcpy function, 88
 Create Thiessen Polygons (Analysis) tool, 78, 117, 204, 504
 CURRENT map, 510–512, 517, 520, 522, 525
 Cursor objects, 311–313, 317–319
 custom functions, 262–263, 265–272, 274, 498

D

DataFrame mapping class, 506–507
 DataFrame objects, 514–519, 524–525
 dataType Describe property, 155–161, 168, 170, 284, 382, 420
 datetime standard (built-in) module, 182, 273–274, 276–277, 301–303, 503
 dBASE file. *See* dBase tables
 dBase tables, dependent vs. independent (stand-alone) 5–6, 189–190, 238
 Debugger, 14, 200, 203, 228–236, 257, 263–265, 298, 406, 482
 Debugging a script, 65–69, 200–203, 223–237, 263–265
 Debugging toolbar, 200–201, 203, 228–236, 263
 Debug mode, 229, 230, 234–235, 243, 265
 Decision-making, 133–135, 138, 139, 141–166, 172, 200, 346
 Dedent, 24, 138, 143, 172, 175, 300–301, 359, 486
 Deep copy, 63–64, 281
 Default progressor, 480
 deleteRow Cursor method, 321–322, 372
 Describe object, arcpy, 154–161, 284, 353, 382, 437–438
 Description, DataFrame property, 514
 Dialog box, 89, 98, 234, 435–440, 442
 dict data type, 336, 350
 dirname, os.path function.
 See os module functions
 Docstrings, 265–267, 270–271, 278, 418, 484–485
 Dot notation, 45–46, 53, 61, 83, 85–87, 89, 95, 101, 107, 125, 155, 251, 291, 296, 316, 394, 415, 420, 427–428, 493, 507
 Dynamic typing, 26–27, 37

E

elif, Python keyword, 143–145, 147, 179, 338, 350
 ELSE, pseudocode keyword, 134–135, 138
 else, Python keyword, 143–144, 147, 160, 162–164
 endswith string method, 48–49, 51, 148, 179, 181, 218, 452, 461

End tag, 387
 enumerate built-in function, 211–213, 263, 367
 Environment settings, 89–91, 107, 181, 197, 269–270, 472, 507–508
 env.overwriteOutput.
 See overwriteOutput arcpy environment setting
 env.tileSize. *See* tileSize arcpy environment setting
 env.workspace. *See* workspace arcpy environment setting
 Erase (Analysis) tool, 116
 Error handling, 241–256, 320, 436
 Escape sequences, 49–51, 358, 366
 Exceptions, 30–31, 67, 224–225, 243–248, 268, 282, 292–293
 except, Python keyword, 243–247
 ExecuteError arcpy exception, 225, 251–255
 execute, Python toolbox tool class method, 489, 491, 494–496
 Exists function, arcpy, 88, 166, 191, 218
 exists, os.path function. *See* os module functions
 Expressions, for arcpy tool parameters, 100–101
 extend list method, 61
 extent Describe property, 155–156
 extent object, arcpy, 155–156, 515–516
 extent, DataFrame property, 515–516
 extractall zipfile method, 399–400
 Extract by Attributes (Spatial Analyst) tool, 153

F

False, built-in constant, 30, 48, 91, 145, 166, 341–342
 Feature class, 5–6, 98, 111, 153–154, 187–188, 315
 FeatureClass type data, 156, 158–161
 Feature layers, temporary, 111, 131, 153–154, 199
 Feature to Point (Data Management) tool, 117, 289, 474, 504
 Feature Vertices to Point (Data Management) tool, 117
 Fetch, 135–136, 397–401
 Field object, arcpy, 194–196, 272
 fields property, arcpy cursor, 313, 314
 __file__ built-in variable, 128, 162, 295
 file data type, 357–359, 362, 364, 365, 372, 374, 376–379, 393, 397–399, 404, 406, 437, 438, 443–444
 File dialog box, 436–437
 File i/o, 359–379

- filePath MapDocument property, 510–513
 - Filters, 472–475
 - findAll BeautifulSoup method, 396–398, 404, 406
 - find, BeautifulSoup method, 395
 - find, string method, 49, 55
 - float data type, 37–38, 287
 - for, Python keyword, 24, 175–181
 - FOR-loop, 135–136, 138–139, 171–172
 - in batch geoprocessing, 197–198
 - with dictionaries, 343–349
 - with the enumerate function, 212
 - with file dialogs, 441
 - with file reading, 364–365, 367–368
 - with GIS fields, 194–195
 - with GIS filenames, 189–190
 - with map layers, 516
 - with the `os.walk` method, 217
 - with xml tags, 396–397
 - with the zip function, 214
 - format, Describe property, 155–158
 - format, string method, 53–54, 58, 131, 152
 - forward slash. *See* /
 - Frequency (Analysis) tool, 116, 207, 232
 - Full path file name, 96, 121, 125–126, 165, 181–182, 215, 294–295, 513
 - Functional programming, 2, 421, 429
 - Functions, built-in, 24–26
 - arcpy, 87–89
 - os module (*see* os module functions)
 - parameters (*see* Parameters (for functions))
 - signatures (*see* Signatures (for functions))
 - user-defined, 261–283, 296–297
- G**
- geodatabase, 6–7, 165–166, 189, 291, 491
 - Geometry object, arcpy, 101, 113–114, 131, 207, 315–316, 325–327
 - geometry token for arcpy, 315–317, 325, 327
 - geoprocessing, 3, 30, 83–89, 95, 197
 - Geoprocessing Window, 249, 451–453
 - Get Count (Management) tool, 105, 111
 - getcwd, os function. *See* os module functions
 - getExtent Layer method, 515, 516
 - GetMessage, arcpy function, 250, 253, 255
 - GetMessageCount arcpy function, 249, 250
 - GetMessages arcpy function, 249–252
 - getmtime, os.path function. *See* os module functions
 - getOutput, a Result object method, 105–106
 - GetParameterAsText arcpy function, 120–121
 - getParameterInfo
 - Python toolbox method, 489–493, 497–498
 - ToolValidator method, 483, 484
 - getSelectedExtent Layer method, 516
 - getsize, os.path function.
 - See* os module functions
 - Graphical user interface (GUI), 9, 10, 435–437, 449, 499
 - GRID raster, 4–5, 108, 117, 168, 181
 - GUI. *See* Graphical user interface (GUI)
- H**
- Hard-coding, 119–120, 178, 314, 394
 - Hash sign. *See* #
 - has_key deprecated. *See* in keyword for dictionaries
 - help, built-in function, 26, 269
 - HTML, 385
 - HTML lists, 389, 392, 393
 - HTML tables, 389–390
- I**
- if, Python keyword, 24, 142
 - immutable, 42, 46, 61, 280–283
 - import, Python keyword, 23, 31–32, 85
 - indentation, 24, 143, 183
 - IndexError exception, 41, 123, 161, 245, 465
 - indexing, zero-based, 41–42
 - index list method, 61, 369
 - __init__ method, 418–420, 425
 - in keyword in FOR-loops, 175–176.
 - See also* FOR-loop
 - in keyword for Boolean expression, 146, 149
 - in keyword for dictionaries, 341–343
 - in keyword for iterating. *See* in keyword for FOR-loops
 - in keyword for membership, 44–45, 60, 149
 - in-place method, 61, 62, 340
 - InsertCursor arcpy method, 310, 322
 - insert list method, 61
 - insertRow Cursor method, 322, 323, 326
 - Instantiate (create an object instance), 420, 517
 - int data type, 37–38, 40
 - integer division, 38, 56, 226–227
 - integrated development environment (IDE), 13–15, 17, 21–23, 48, 69, 111, 200, 229
 - internal validation, 481, 485, 493
 - Intersect (Analysis) tool, 102, 199, 273

isLicensed
 Python toolbox method, 492
 ToolValidator method, 489
 item, 59–61, 64, 336–337
 items dictionary function, 342
 iterate, 136, 175, 217, 311, 318, 343, 365
 iterating variable, 174, 176–178, 197, 210,
 212, 214, 217, 233, 253, 254, 317

J

join, os.path function. *See* os module
 functions
 join string method, 47–48

K

key, 336–338
 KeyboardInterrupt exception, 173, 238
 KeyError exception, 337–338, 341, 342
 keys dictionary function, 342, 343, 423
 keywords, pseudocode, 134–135
 keywords, Python, 23–24, 29
 KML, 401–406

L

Layer file (.lyr extension), 6, 158, 471, 477,
 478, 520
 Layer mapping class. *See* Layer object
 Layer object, 506, 515–521, 525
 Layer object symbology, 521
 Layout element objects, 522–524
 len built-in function, 40–41, 60
 Linear unit, tool parameters, 99–100
 Line continuation. *See* \ (backslash) for line
 continuation
 list built-in function, 63
 List comprehension, 209–211, 330, 340, 366
 ListDataFrames mapping method, 514
 ListDatasets arcpy function, 197
 list data type, 59–64
 listdir, os function. *See* os module
 functions
 ListEnvironments arcpy function,
 91, 196–197
 ListFeatureClasses arcpy function,
 188–189
 ListFields arcpy function, 194–196,
 314, 315
 ListLayers mapping method, 515, 516
 List methods, 61–62, 72, 190, 400
 ListPrinterNames arcpy function, 88, 197
 ListRasters, arcpy function, 190–192

ListTables arcpy function, 189–190
 ListWorkspaces arcpy function, 189
 locking, data, 83–84, 317–320, 358, 525
 logical operators, 146, 148–151, 165
 Logic errors, 226–229
 Looping, 133, 138, 139, 171–172
 Looping variable, 136, 138, 139. *See also*
 Iterating variable

M

Make XY Event Layer (Management) tool,
 111, 153, 199
 Map algebra, 107, 109, 242
 MapDocument mapping class, 506
 MapDocument objects, 507, 513
 mapping module, 505–507
 math, standard Python module, 31, 355
 max built-in function, 226, 323
 Mean Center (Spatial Statistics) tool, 289
 median numpy function, 348
 membership operators. *See in* Keyword for
 membership
 Merge (Management) tool, 101
 min built-in function, 25, 29, 226
 Minimum Bounding Geometry (Data
 Management), 117, 120, 259
 Minus (Spatial Analyst) tool, 110
 mkdir, os function. *See* os module
 functions
 Modelbuilder, 81, 82
 model, Modelbuilder, 82, 92, 466, 499
 modules, standard (a.k.a. built-in), 31–32
 modules, user-defined, 291–308, 427–428
 Multivalued input, 98, 101–102, 468
 Multivalued parameters. *See* Multivalued input
 Multivalued string, 101, 487, 494
 Multi-way conditional construct, 338, 339, 350
 mutable, 61, 64, 280–283, 321, 336, 339–341
 mutable arguments, 280–283

N

name, arcpy DataFrame object property,
 514–515
 name, arcpy Field object property, 194–196
 name, arcpy Layer object property, 516–517
 name, arcpy Layout Element object
 property, 522–523
 name, arcpy Parameter object property, 491
 name, BeautifulSoup Tag object property, 395
 __name__ built-in variable, 300, 427
 named exceptions, 244–245, 248, 251–252
 NameError exception, 28, 129, 224, 263, 419

namelist, zipfile module method, 399–400
 nested looping, 138, 179–180
 nested conditional structures, 138, 158–160
 next, arcpy.da Cursor method, 312, 313, 316, 317, 321, 322
 None, a common mistake, 274–275
 None, built-in constant, 30, 145, 242, 274
 not, Python keyword, 24, 146, 149
 numpy module, 348, 349

O

object instance, 419–421, 426–428
 object-oriented programming (OOP), 2, 45–46, 416–417, 421, 423, 424, 428–430
 objects, 2, 61, 45, 86, 417
 obtained from, Script Tool parameter property, 475–477, 497
 open, built-in function, 357
 optional parameters, for arcpy tools, 97, 102–103
 optional parameters, for user-defined functions, 262, 270
 or, Python keyword, 24, 146, 148–149
 os module functions, 125–128
 os.getcwd, 292, 363
 os.listdir, 128, 180–181
 os.makedirs, 400
 os.mkdir, 165
 os.path.abspath, 128, 295
 os.path.basename, 125
 os.path.dirname, 125, 496
 os.path.exists, 165, 182
 os.path.getmtime, 182
 os.path.getsize, 126
 os.path.join, 125–126, 165, 182–183, 198, 295
 os.path.splitext, 127–128
 os.path.relpath, 408
 os.startfile, 513
 os.walk, 180, 214–219
 os standard (built-in) module, 31
 overwriteOutput, arcpy environment setting, 91

P

package (Python), 85–87, 293, 390
 parameter direction, in Script Tools, 465–467
 parameter for ModelBuilder, 81–83, 466
 parameter object
 for Python Toolboxes 491, 493, 494, 497
 for the ToolValidator Class, 484–487
 parameter properties

 in Python Toolboxes 491, 497–499
 in Script Tools 458, 464–478
 in the ToolValidator class 485, 487
 Parameters (for functions), 25–26
 parameter table

 in arcpy help 97–98, 461
 in the Script Tool Wizard, 458, 459
 parameter type
 in Python Toolboxes, 491
 in Script Tools 462, 465

parsing

 HTML 393–397
 KML 403–405
 Python code 224
 text files 365–370

pickle built-in module, 376–378

Placemark, kml tag, 402–404

Point object, arcpy, 88, 325, 327, 328, 406

Point Distance (Analysis) tool, 185, 289

Points to Line (Data Management) tool, 177, 199, 208, 213

Polygon object, arcpy, 325, 327, 328

Polygon Neighbors (Analysis) tool, 102

Polygon to Line (Data Management) tool, 205

Polyline object, arcpy, 325

pop, list method, 61, 73, 374, 375

prettyfy, BeautifulSoup method, 394

print_exc traceback module function, 247–248

procedures, 135, 137, 262

progressor, 478–481

pseudocode, 133–139

PyScripter, 15–20, 48, 69, 121, 129, 173, 200, 235–237

Python interpreter, 31

Python prompt, 15–17

Python script, 13

Python toolbox, 489–499

PythonWin, 14–21, 48, 65, 69, 129, 173, 183, 200, 224, 229–237, 363

PythonWin Interactive Window, 14–15

PythonWin script window, 15

Q

Quick Export (Data Interoperability) tool, 205, 258

R

r. *See* Raw strings

range built-in function, 62, 178

RasterDataset type data, 155, 156, 158

Raster object, 107–110

`raw_input` built-in function, 435–436
 Raw strings, 50–51
`read`, file method, 358, 444
`readline`, file method, 358, 364–366, 368, 371, 444
`readlines`, file method, 358, 444
`RefreshActiveView` arcpy method, 511, 512
 Relative paths, 294–295, 387–388, 457, 470
`relativePaths` MapDocument property, 512, 513
`relpath`, `os.path` function.
 See os module functions
`reload` built-in function, 298–299
`remove` list method, 61
 Repetition, 133, 136, 171
`replace` string method, 42, 46–47
`reset` Cursor method, 312, 313, 321, 322
 Result object, arcpy, 103–107, 116
 Return multiple values, 276–277
`return`, Python keyword, 271–275
`reverse` list method, 62–63
`round` built-in function, 25, 26
 Running a script, 19, 20, 69
 Running in debug mode, 234–235

S

`saveACopy`, MapDocument method, 513
`save`, MapDocument method, 513
`save`, Raster object function, 108
 Script level variable, 282, 283
 Script Tool, 114, 449–488
 Script Tool buttons, 454–455
`SearchCursor` arcpy method, 310, 311–313
 Select Layer By Attribute (Data Management) tool, 153–154
 Select (Analysis) tool, 150, 151
`self`, in user-defined classes, 419–420
 Sequence data objects, 45, 60, 176, 210, 211
 Sequence operations, 45, 60–61
 Sequential steps, 133–134
`SetParameterAsText` arcpy function, 466, 467, 469, 471, 477, 478, 497
`SetProgressorLabel` method, 479–481
`SetProgressor` method, 478–481
`SetProgressorPosition` method, 479–481
 Shallow copy, 63
 Shapefile, 5
`shapeType` Describe property, 158, 159
`shutil` standard (built-in) module, 371
 Signatures (for functions), 26, 190, 267, 310, 439
 Simplify Polygon (Cartography) tool, 204

Sine (Spatial Analyst) tool, 307, 490, 492
 Slicing, 42–43, 44, 45, 56, 60, 127, 128, 366
 Smooth Line (Cartography) tool, 158
 Soft-coding, 119–120
`sort` list method, 61, 62, 297
 Spatial analyst, calling tools, 107–108
 Spatial Analyst, importing the toolbox, 109
 Split Line (Data Management) tool, 208
`splitext`, `os.path` function.
 See os module functions
`split` string method, 47, 366, 369
 Split (Analysis) tool, 116
 Square root (Spatial analyst) tool, 107, 108, 286
 Start tag, 387
 Step buttons, for debugging, 200–203, 229–230, 234–236, 263–265
 Stepping through code. *See Step buttons, for debugging*
 Step progressor, 479–481
`str` data type, 27, 40, 51, 189
 String formatting, 53–54, 152, 153, 330, 362, 391, 454
 String literal, 38–40, 42, 111–113, 145
`String lstrip` method, 49, 211
 String variable, 38–43, 48
 Subroutines, 135, 236, 262
 Symbology
 for Feature Set Script Tool parameter, 470, 471, 477–478
 mapping module `Layer` object property, 520–522
 the Script Tool parameter property, 464
 storing in Layer files, 6
 ToolValidator Parameter object property, 484
`symbologyType` mapping module `Layer` object property, 520, 521
`SyntaxError` exception, 29, 30, 39, 65, 148, 152
 Syntax errors, 65, 66, 174, 224
`sys.argv` property, 121–122
`sys.path` variable, 31–32, 293
`sys` standard (built-in) module, 31, 32

T

Table to dBASE (Conversion) tool, 206
 TabNanny, 66, 183
 Tag attributes (HTML), 387, 395
 Tag contents
 HTML, 387, 395, 396
 KML, 403, 404
 Tags, 385–390, 392–396, 402–406
 Text box, 435

- Thrown exceptions, 30
 - `tileSize`, `arcpy` environment setting, 91
 - `time standard` (built-in) module, 301–303, 480
 - Times (Spatial analyst) tool, 110
 - `title` MapDocument property, 512, 523, 524
 - `title`, String method, 49, 321
 - `title`, `tkFileDialog` option, 440
 - `tkFileDialog` module, 436–445
 - Toolbox class, 489–490, 504
 - Tool class, 489–499, 504
 - `ToolValidator` class, 481–488
 - Tracebacks, 28–30, 65, 67–69, 225, 233, 247, 248
 - `traceback standard` (built-in) module, 247–248
 - `True`, built-in constant, 30, 91, 145
 - `try`, Python keyword, 24, 243–244, 247, 255
 - tuple data type, 64, 165, 176, 213, 276, 312, 321, 322
 - `type` built-in function, 27
 - `TypeError` exception, 30, 42, 43, 44, 64, 242, 244, 268, 361, 436, 454
- U**
- u. *See* unicode strings
 - `UnboundLocalError` exception, 282
 - unicode data type. *See* Unicode strings
 - Unicode strings, 51–52, 56, 189, 461
 - `UpdateCursor` `arcpy` method, 310–311, 321–322, 328
 - `updateMessages`
 - Python toolbox method, 489, 493–494
 - `ToolValidator` method, 487–488
 - `updateParameters`, Python toolbox tool
 - class method, 484, 487–489, 491, 493–494, 504
 - `updateRow` Cursor method, 321
 - `upper`, string method, 47, 52
 - URL (Universal Resource Locator), 397, 399
 - `urllib2 standard` (built-in) module, 397–399
 - `urlopen` `urllib2` function, 397, 399
 - User-defined module, 291–303, 427–428, 495
- V**
- Validation, 445
 - in Python toolboxes 489, 493–494, 499
 - in the `ToolValidator` class 481–483, 485;
 - Value, for variables, 26–27
 - Value object, 485, 496, 497
 - `values` dictionary function, 342
 - ValueTable object, 101, 102
 - Variables, 26–29
 - Vector data, 5, 6, 111, 519
- W**
- `walk`, `arcpy.da` function, 219, 276, 277
 - `walk`, `os` function. *See* `os` module functions
 - Watch window, 201–203, 232, 236, 238, 239
 - where clause
 - for `arcpy` selection tools, 150–154
 - for selecting rows with cursors, 328–330
 - `while`, Python keyword, 24, 172–174
 - WHILE-loop, 136, 138, 171–174, 176, 178, 253, 254
 - whitespace characters, 49–50, 183
 - Widgets, 458–460, 462–464, 474
 - wild card
 - for listing datasets, 190–193, 196, 459
 - for listing map layers, 516
 - for listing layout elements, 523
 - Window Explorer vs. ArcCatalog, 7, 8
 - Window focus, 18, 19, 34, 65
 - `with`, Python keyword, 24
 - and cursors, 320
 - and file objects, 359
 - Working directory, 182, 292, 293, 362–363, 441
 - `workspace`, `arcpy` environment setting, 32, 116, 165, 181–182, 188–189, 198, 201, 242, 436, 507
 - Wrapped tags, 386, 387, 389
 - `write`, file method, 359–362
- X**
- XMax, Extent object properties, 156, 515
 - XMin, Extent object properties, 515
 - XML, 402
- Y**
- YMax, Extent object properties, 515
 - YMin, Extent object properties, 515, 516
- Z**
- `ZeroDivisionError` exception, 245, 246
 - `zip` built-in function, 213–214
 - `zipfile standard` (built-in) module, 399–400
 - `zoomToSelectedFeatures` DataFrame method, 515