

Index

Symbols

- != (not-equal-to operator), [Compare Dictionaries](#)
- # (pound sign), [Comment with #](#)
- \$ (anchor), [Patterns: Using Specifiers](#)
- % (per cent sign), [Old style: %-Old style: %](#)
- & (set intersection operator), [Combinations and Operators-Combinations and Operators](#)
- * (asterisk)
 - duplicating lists with, [Duplicate All Items with *](#)
 - duplicating strings with, [Duplicate with *](#)
 - duplicating tuples with, [Duplicate Items with *](#)
 - exploding/gathering positional arguments with, [Explode/Gather Positional Arguments with *-Explode/Gather Positional Arguments with *](#)
 - multiplication operator, [Integer Operations](#)
 - wildcard, [Text Strings: Regular Expressions](#)
- ** (asterisks)
 - dictionary operator, [Combine Dictionaries with {**a, **b}, Pass an argument as part of the URL path](#)
 - exponentiation, [Integers and Variables](#)
 - keyword arguments with, [Explode/Gather Keyword Arguments with **, Named Tuples](#)
- + operator, [Integer Operations](#), [Combine by Using +](#), [Combine Tuples by Using +](#), [Combine Lists by Using extend\(\) or +](#)
- (difference), [Combinations and Operators](#)
- . (directory), [Check Type with isfile\(\)](#)
- .. (parent directory), [Check Type with isfile\(\)](#)
- / (slash), [Pathnames](#)
- <= (subset), [Combinations and Operators](#), [Combinations and Operators](#)
- = (assignment operator), [Assignment](#), [Assign with =](#), [Assign with =](#)
- == (equality operator), [Compare with if, elif, and else](#), [Compare Dictionaries](#)

> (proper superset), [Combinations and Operators](#)
>= (superset), [Combinations and Operators](#)
[key], [Add or Change an Item by \[key \]-Add or Change an Item by \[key \]](#), [Get an Item by \[key\] or with get\(\)](#)
[offset] (see offset)
[] (square brackets), [Create with \[\]](#)
\ (backslash), [Continue Lines with \](#), [Escape with \](#), [Escape with \](#), [Pathnames](#)
\n (new line), [Escape with \](#)
\N{name}, [Python 3 Unicode Strings](#)
\t (tab), [Escape with \](#)
\u, [Python 3 Unicode Strings](#)
\U, [Python 3 Unicode Strings](#)
_ (underscore), [Literal Integers](#), [Uses of _ and _ in Names](#)
_ (underscores), [Uses of _ and _ in Names](#)
__ (double underscores), [Docstrings](#), [Uses of _ and _ in Names](#), [Name Mangling for Privacy](#), [Magic Methods](#), [Modules Versus Objects](#)
__init__ () method, [Initialization](#)
__str__ () method, [Magic Methods](#)
{ } (curly brackets)
 for dictionary creation, [Create with {}-Create with dict\(\)](#), [Delete All Items with clear\(\)](#)
 for string formatting, [New style: {} and format\(\)-New style: {} and format\(\)](#)
| (vertical bar), [Combinations and Operators](#)
^ (anchor), [Patterns: Using Specifiers](#)
^ (exclusive or), [Combinations and Operators](#)

A

absolute imports, [Relative and Absolute Imports](#)
abspath() function, [Get a Pathname with abspath\(\)](#)
accumulate() function, [Iterate over Code Structures with itertools](#)
add() function, [Add an Item with add\(\)](#), [The Object-Relational Mapper \(ORM\)](#)
addition, [Integer Operations](#)
add_all () function, [The Object-Relational Mapper \(ORM\)](#)

aggregation, [Aggregation and Composition](#)

algebra, [Linear Algebra](#)

algorithms, data structures and, [Algorithms and Data Structures](#)

alias, [Import a Module with Another Name](#)

alignment functions, [Alignment](#)

Amazon Web Services (AWS), [Amazon Web Services](#)

Anaconda

conda package manager, [Install Anaconda's Package Manager](#)
[conda](#)

installation, [Install Anaconda](#)

anchor (\$), [Patterns: Using Specifiers](#)

animation, [3-D Graphics](#)

anonymous functions, [Anonymous Functions: lambda](#)

Ansible, [Remote Management Tools](#)

Apache web server, [Apache](#)

API (application programming interface), [DB-API](#), [Web Services and APIs](#), [Web APIs and REST](#)

append() function, [Add an Item to the End with append\(\)](#)

arange() function, [Make an Array with arange\(\)](#)

arguments, [Arguments and Parameters-Mutable and Immutable Arguments](#)

default parameter values for, [Specify Default Parameter Values](#)

defined, [Arguments and Parameters](#)

exploding/gathering keyword arguments, [Explode/Gather Keyword Arguments with **](#)

exploding/gathering positional arguments, [Explode/Gather Positional Arguments with *-Explode/Gather Positional Arguments with *](#)

keyword, [Keyword Arguments](#)

keyword-only, [Keyword-Only Arguments](#)

mutable and immutable, [Mutable and Immutable Arguments](#)

None, [None Is Useful](#)

positional, [Positional Arguments](#)

arithmetic calculations, [Perform Rational Arithmetic with fractions](#)

array() function, [Make an Array with array\(\)](#)

arrays, [Use Packed Sequences with array](#)

(see also NumPy)

changing with [reshape\(\)](#), [Change an Array's Shape with reshape\(\)](#)
getting elements from, [Get an Element with \[\]](#)
making with [arange\(\)](#), [Make an Array with arange\(\)](#)
making with [array\(\)](#), [Make an Array with array\(\)](#)
making with [zeros\(\)/ones\(\)/random\(\)](#), [Change an Array's Shape with reshape\(\)](#)
math functions and, [Array Math](#)
packed sequences with, [Use Packed Sequences with array](#)

art (see [graphics](#))

ASCII, [Text Strings: Unicode](#)

assembly language, [Assembler](#)

assertion, [Test with unittest](#)

assignment operator (=), [Assignment](#), [Assign with =](#), [Assign with =](#)

assignment, copying versus, [Copying](#)

asterisk (*)

duplicating lists with, [Duplicate All Items with *](#)

duplicating strings with, [Duplicate with *](#)

duplicating tuples with, [Duplicate Items with *](#)

exploding/gathering positional arguments with, [Explode/Gather Positional Arguments with *-Explode/Gather Positional Arguments with *](#)

multiplication operator, [Integer Operations](#)

wildcard, [Text Strings: Regular Expressions](#)

asterisks (**)

dictionary operator, [Combine Dictionaries with {**a, **b}](#), [Pass an argument as part of the URL path](#)

exponentiation, [Integers and Variables](#)

keyword arguments with, [Explode/Gather Keyword Arguments with **](#), [Named Tuples](#)

asynchronous (term), [Concurrency](#)

asynchronous functions, [Async Functions](#)

asynchronous tasks, [Something Completely Different: Async-Async Frameworks and Servers](#)

asyncio alternatives, [Asyncio Alternatives-Asyncio Alternatives](#)

coroutines and event loops, [Coroutines and Event Loops-Coroutines and Event Loops](#)

frameworks and servers, [Async Frameworks and Servers](#)

versus other approaches, [Async Versus...](#)
asyncio library
about, [asyncio](#), [Coroutines and Event Loops](#)
alternatives to, [Asyncio Alternatives-Asyncio Alternatives](#)
attributes, [Attributes](#)
accessing, [Attribute Access-Class and Object Attributes](#)
class and object, [Class and Object Attributes](#)
finding with self argument, [In self Defense](#)
getters and setters, [Getters and Setters](#)
initialization and, [Initialization](#)
name mangling for privacy, [Name Mangling for Privacy](#)
properties for access, [Properties for Attribute Access-Properties for Attribute Access](#)
properties for computed values, [Properties for Computed Values](#)
attrs package, [Attrs](#)
audio, [Audio and Music](#)
Azure, [Microsoft Azure](#)

B

back pressure technique, [Beyond Queues](#)
backslash (\), [Continue Lines with \](#), [Escape with \](#), [Escape with \](#),
[Pathnames](#)
bare metal, [Operating Systems](#)
bases, [Bases](#)
basicConfig() function, [Log Error Messages](#)
BeautifulSoup, [BeautifulSoup](#)
Beowulf cluster, [The Cloud](#)
best practices
code, testing, [Test-Continuous Integration](#)
debugging, [Debug Python Code-Use breakpoint\(\)](#)
finding code, [Find Python Code](#)
integrated development environments (IDEs), [Integrated Development Environments-JupyterLab](#)
logging, [Log Error Messages-Log Error Messages](#)
optimizing code, [Optimize-Numba](#)
resources, [How You Can Learn More-Getting a Python Job](#)

source control, [Source Control-Git](#)

big data, [Big Fat Data](#)

binary data, [Binary Data-Bit Operators](#)

bit-level integer operators, [Bit Operators](#)

bytes and bytearrays, [bytes and bytearray-bytes and bytearray](#)

converting bytes/strings with binascii(), [Convert Bytes/Strings with binascii\(\)](#)

converting with struct, [Convert Binary Data with struct-Convert Bytes/Strings with binascii\(\)](#)

practice exercise answers, [12. Wrangle and Mangle Data-12. Wrangle and Mangle Data](#)

practice exercises, [Things to Do](#)

reading binary files, [Read a Binary File with read\(\)](#)

third-party tools for, [Other Binary Data Tools](#)

writing binary files, [Write a Binary File with write\(\)](#)

binary files, [Binary Files](#)

HDF5, [HDF5](#)

padded files and memory mapping, [Padded Binary Files and Memory Mapping](#)

spreadsheets, [Spreadsheets](#)

TileDB, [TileDB](#)

binascii() function, [Convert Bytes/Strings with binascii\(\)](#)

bit-level integer operators, [Bit Operators](#)

bits, [In the Beginning Was the Bit](#)

Blender, [3-D Animation](#)

Bokeh, [Bokeh](#)

bool() function, [Booleans](#)

boolean operators, [Compare with if, elif, and else](#)

booleans, [Numbers-Things to Do](#)

Bottle, [Bottle-Bottle](#)

break statement, [Cancel with break, Cancel with break](#)

breakpoint() function, [Use breakpoint\(\)](#)

breakpoints, [Use pdb](#)

business applications, [Py at Work-Things to Do](#)

business tasks, [Carrying Out Business Tasks](#)

data processing, [Processing Business Data-Additional Sources of Information](#)

data security, [Business Data Security](#)
financial tools, [Python in Finance](#)
government data sources, [Additional Sources of Information](#)
maps, [Maps-Applications and Data](#)
Microsoft Office Suite, [The Microsoft Office Suite](#)
open source packages, [Open Source Python Business Packages](#)
practice exercise answers, [21. Py at Work](#)
practice exercises, [Things to Do](#)
bytearrays, [bytes and bytearray-bytes and bytearray](#)
bytes, [Text Strings: Unicode](#), [bytes and bytearray-bytes and bytearray](#),
[In the Beginning Was the Bit](#)
BytesIO, [BytesIO and StringIO-BytesIO and StringIO](#)

C

C, [Python Versus the Language from Planet X](#)
C#, [Python Versus the Language from Planet X](#)
C++, [Python Versus the Language from Planet X](#)
caches, [Memory and Caches](#)
caches, in Redis, [Caches and expiration](#)
calc() function, [concurrent.futures](#)
calendars/clocks, [Calendars and Clocks-Things to Do](#)
 challenges of date representation, [Calendars and Clocks](#)
 datetime module, [The datetime Module-The datetime Module](#)
 leap years, [Leap Year](#)
 practice exercise answers, [13. Calendars and Clocks](#)
 practice exercises, [Things to Do](#)
 reading/writing dates and times, [Read and Write Dates and Times-
 Read and Write Dates and Times](#)
 standard Python time interconversions, [All the Conversions](#)
 third-party modules for, [Alternative Modules](#)
 time module, [Using the time Module-Using the time Module](#)
call() function, [Create a Process with subprocess](#)
capitalization, [Case](#)
case, [Case](#)
centering of string, [Alignment](#)
chain() function, [Iterate over Code Structures with itertools](#)

characters, defined, [Text Strings](#)

chdir() function, [Change Current Directory with chdir\(\)](#)

check_output () function, [Create a Process with subprocess](#)

Cheese Shop (Python Package Index), [Find Python Code](#)

Chef, [Remote Management Tools](#)

chmod() function, [Change Permissions with chmod\(\)](#)

chown() function, [Change Permissions with chmod\(\)](#)

classes

- aggregation and composition, [Aggregation and Composition](#)

- attribute assignment, [Class and Object Attributes](#)

- attributes, [String Module Attributes](#)

- class methods, [Class Methods](#)

- composition, [Aggregation and Composition](#)

- defined, [Variables Are Names, Not Places](#)

- inheritance, [Inheritance-Mixins](#)

- method types, [Method Types-Static Methods](#)

- methods and, [Methods](#)

- modules and objects versus, [When to Use Objects or Something Else](#)

- practice exercise answers, [10. Oh Oh: Objects and Classes-10. Oh Oh: Objects and Classes](#)

- practice exercises, [Things to Do](#)

clear() function, [Delete All Items with clear\(\)](#), [Delete All Items with clear\(\)](#)

client-server system, [Web Clients](#)

closures, [Closures](#)

cloud computing, [Clouds-OpenStack](#), [The Cloud](#)

clusters, [The Cloud](#)

code, [Choose with if-Things to Do](#)

- comments, [Comment with #](#)

- continuation character (\), [Continue Lines with \](#)

- debugging, [Debug Python Code-Use breakpoint\(\)](#)

- distributing, [Distribute Your Programs](#)

- finding, [Find Python Code](#)

- optimizing, [Optimize-Numba](#)

- source control for, [Source Control-Git](#)

- testing, [Test-Continuous Integration](#)

code structures

comparative statements, [Compare with if, elif, and else-Compare with if, elif, and else](#)

iterating over, [Iterate over Code Structures with itertools](#)

membership operator, [Do Multiple Comparisons with in](#)

practice exercise answers, [4. Choose with if](#)

practice exercises, [Things to Do](#)

true/false values, [What Is True?](#)

walrus operator, [New: I Am the Walrus](#)

columns, [Relational Databases](#)

combining strings, [Combine by Using +](#)

command automation, [Command Automation](#)

comments, [Comment with #](#)

Common Gateway Interface (CGI), [Web Server Gateway Interface \(WSGI\)](#)

comparisons

lists, [Compare Lists](#)

operators for, [Compare with if, elif, and else](#)

tuples, [Compare Tuples](#)

complex numbers, [Working with Complex Numbers](#)

composition, [Aggregation and Composition](#)

comprehensions

dictionaries, [Dictionary Comprehensions](#)

generator, [There Are No Tuple Comprehensions](#), [Generator Comprehensions](#)

lists, [Create a List with a Comprehension](#)-[Create a List with a Comprehension](#)

sets, [Set Comprehensions](#)

computed values, [Properties for Computed Values](#)

concatenation

strings, [Combine by Using +](#)

tuples, [Modify a Tuple](#)

concurrency, [Concurrency-Beyond Queues](#)

asynchronous versus other approaches, [Async Versus...](#)

asyncio module, [asyncio](#)

concurrent.futures module, [concurrent.futures-concurrent.futures](#)

green threads, [Green Threads and gevent](#)-[Green Threads and gevent](#)

practice exercise answers, [15. Data in Time: Processes and Concurrency](#)

practice exercises, [Things to Do](#)

processes and, [Processes](#)

queues and, [Queues](#)

Redis, [Redis-Redis](#)

threads and, [Threads-Threads](#)

twisted framework, [twisted](#)

concurrent.futures module, [concurrent.futures-concurrent.futures](#)

conda, [Install Anaconda's Package Manager conda](#)

configparser module, [Configuration Files](#)

configuration files, [Configuration Files](#)

containers, [Containers](#)

continuation character (\), [Continue Lines with \](#)

continue statement, [Skip Ahead with continue](#), [Skip with continue](#)

continuous integration, [Continuous Integration](#)

cooperative multitasking, [Coroutines and Event Loops](#)

copy() function

copying files with, [Copy with copy\(\)](#)

copying keys and values from one dictionary to another, [Copy with copy\(\)](#)

copying list with, [Copy with copy\(\), list\(\), or a Slice](#)

copying, assignment versus, [Copying](#)

coroutines, [Coroutines and Event Loops-Coroutines and Event Loops](#)

count() function, [Count Occurrences of a Value with count\(\)](#)

counter() function, [Count Items with Counter\(\)](#)

CPU (central processing unit), [The CPU](#)

CPU bound, [Concurrency](#)

crawling, [Crawl and Scrape-Requests-HTML](#)

CRUD (Create-Read-Update-Delete), [SQL](#)

CSV (comma-separated values) format, [CSV-CSV](#)

curl, [Test with curl](#)

curly brackets ({ })

for dictionary creation, [Create with {}-Create with dict\(\)](#), [Delete All Items with clear\(\)](#)

for string formatting, [New style: {} and format\(\)-New style: {} and format\(\)](#)

cycle() function, [Iterate over Code Structures with itertools](#)
Cython, [Cython, NumPy, and C Extensions](#)

D

Dask, [Dask](#)

data munging, [Extracting, Transforming, and Loading](#)

data processing (business data), [Processing Business Data-Additional Sources of Information](#)

extracting/transforming/loading data, [Extracting, Transforming, and Loading-Extracting, Transforming, and Loading](#)

validation, [Data Validation](#)

data science, [Wrangle and Mangle Data](#), [Pandas](#)

data serialization, [Data Serialization-Other Serialization Formats](#)

data storage (see persistent storage)

data structures

algorithms and, [Algorithms and Data Structures](#)

assigning value to multiple variable names, [Assigning to Multiple Names](#)

binary data, [Binary Data-Bit Operators](#)

comparisons, [Data Structures So Far](#)

creating complex, [Make Bigger Data Structures](#)

literal values, [Literal Values](#)

mutability, [Mutability](#)

data types (see types)

data visualization, [Plots, Graphs, and Visualization-Bokeh](#)

Bokeh, [Bokeh](#)

Seaborn, [Seaborn](#)

data wrangling, [Extracting, Transforming, and Loading](#)

databases

document databases, [Document Databases](#)

full-text, [Full-Text Databases](#)

graph databases, [Graph Databases](#)

NoSQL data stores, [NoSQL Data Stores-Other NoSQL](#)

relational, [Relational Databases-Other Database Access Packages](#)

terminology, [Relational Databases](#)

time series databases, [Time Series Databases](#)

web framework support for, [Database Frameworks-Database Frameworks](#)

dataclasses, [Dataclasses](#)

DataFrame, [Pandas](#)

dates, representations of, [Calendars and Clocks](#)

(see also calendars/clocks)

datetime object, [The datetime Module-The datetime Module](#), [JSON](#)

daylight savings time, [Using the time Module](#)

DB-API, [DB-API](#)

dbm databases, [The dbm Family](#)

DDL (data definition language), [SQL](#)

debugging, [Debug Python Code-Use breakpoint\(\)](#)

best practices, [Debug Python Code-Use breakpoint\(\)](#)

breakpoint() for, [Use breakpoint\(\)](#)

decorators for, [Use Decorators](#)

printing out strings, [Use print\(\)](#)

with pdb, [Use pdb-Use pdb](#)

decimals, floating points with, [Calculate Accurate Floating Point with decimal](#)

decode() function, [Decode](#)

decorators, [Decorators-Decorators](#), [Use Decorators](#)

deepcopy() function, [Copy Everything with deepcopy\(\)](#)

defaultdict() function, [Handle Missing Keys with setdefault\(\) and defaultdict\(\)](#)

del statement, [Delete an Item by Offset with del](#)

delete an item from list with, [Delete an Item by Offset with del](#)

deleting individual dictionary items, [Delete an Item by Key with del](#)

deque, [Stack + Queue == deque](#)

dict() function

converting two-value sequences into dictionaries, [Convert with dict\(\)](#)

creating dictionaries, [Create with dict\(\)](#)

dictionaries, [Dictionaries and Sets-Dictionary Comprehensions](#)

adding/changing items, [Add or Change an Item by \[key \]-Add or Change an Item by \[key \]](#)

assignment versus copying, [Assign with =](#)

combining, [Combine Dictionaries with {**a, **b}](#)

comparing, [Compare Dictionaries](#)
comprehensions, [Dictionary Comprehensions](#)
converting two-value sequences into, [Create with dict\(\)](#)
copying, [Copy with copy\(\)-Copy Everything with deepcopy\(\)](#)
creating, [Create with {}-Create with dict\(\)](#)
defined, [Little Programs](#)
deleting all items, [Delete All Items with clear\(\)](#)
deleting individual items, [Delete an Item by Key with del](#)
getting all key-value pairs, [Get All Key-Value Pairs with items\(\)](#)
getting all keys, [Get All Keys with keys\(\)](#)
getting all values, [Get All Values with values\(\)](#)
getting items by [key] or with get(), [Get an Item by \[key\] or with get\(\)](#)
getting length, [Get Length with len\(\)](#)
iterating over, [Iterate with for and in](#)
practice exercise answers, [8. Dictionaries-8. Dictionaries](#)
practice exercises, [Things to Do](#)
sets versus, [Sets](#)
testing for keys, [Test for a Key with in](#)
DictReader() function, [CSV](#)
dict_keys () function, [Get All Keys with keys\(\)](#)
difference() function, [Combinations and Operators](#)
directory operations, [Directory Operations-List Matching Files with glob\(\)](#)
(see also pathnames)
changing current, [Change Current Directory with chdir\(\)](#)
creating, [Create with mkdir\(\)](#)
creating subdirectories, [List Contents with listdir\(\)](#)
definition of, [Directory Operations](#)
deleting, [Delete with rmdir\(\)](#)
listing contents of, [List Contents with listdir\(\)](#)
listing matching files, [List Matching Files with glob\(\)](#)
directory, defined, [Files and Directories](#)
Disco, [Disco](#)
distributed computing, [Distributed Computing and Networks](#)
division, [Integer Operations](#)
Django, [Django](#)

DML (data manipulation language), [SQL](#)
Docker, [Docker](#)
docstrings, [Docstrings](#)
doctest, [Test with doctest](#)
document databases, [Document Databases](#)
documentation, [Docstrings](#), [Name and Document](#)
Domain Name System (DNS), [Domain Name System](#)
double underscore (`__`), [Docstrings](#), [Uses of `_` and `__` in Names](#), [Name Mangling for Privacy](#), [Magic Methods](#), [Modules Versus Objects](#)
double() function, [Use Decorators](#)
duck typing, [Duck Typing](#)
dunder, [Docstrings](#)
(see also double underscores (`__`))
dynamic languages, [Python Versus the Language from Planet X](#)

E

electricity, basics of, [Electricity](#)
ElementTree module, [XML](#)
elif statement, [Compare with if, elif, and else](#)
else statement, [Compare with if, elif, and else](#), [Check break Use with else](#), [Check break Use with else](#)
email modules, [Python Email Modules](#)
empty set, [Sets](#)
encode() function, [Encode](#)
epoch, [Using the time Module](#)
equality operator (`==`), [Compare with if, elif, and else](#), [Compare Dictionaries](#)
error handling, [Exceptions-Make Your Own Exceptions](#)
(see also exceptions)
escape sequences, [Create with Quotes](#), [Escape with `\`](#), [Escape with `\`](#)
Esri, [Formats](#)
ETL (extract, transform, load), [Extracting, Transforming, and Loading-Extracting, Transforming, and Loading](#)
event loops, [Coroutines and Event Loops-Coroutines and Event Loops](#)
event-based programming, [Green Threads and gevent](#)
event-based servers, [Other Python Web Servers](#)

exceptions

defining exception types, [Make Your Own Exceptions](#)

handling errors with try and except, [Handle Errors with try and except](#)

logging error messages, [Log Error Messages-Log Error Messages](#)

exclusive or (^), [Combinations and Operators](#)

exists() function, [Check Existence with exists\(\)](#)

expiration dates, [Caches and expiration](#)

expire() function, [Caches and expiration](#)

expireat() function, [Caches and expiration](#)

exponentiation (**), [Integers and Variables](#)

Expression Language (SQLAlchemy), [The SQL Expression Language](#)

extend() function, [Combine Lists by Using extend\(\) or +](#)

F

f-strings, [Create with Quotes, Newest Style: f-strings](#)

false values, [What Is True?, None Is Useful](#)

fanin pattern, [Networking Patterns](#)

fanout pattern, [Networking Patterns](#)

FIFO (first in, first out) queue, [Get an Item by Offset and Delete It with pop\(\)](#)

file handling, [Files and Directories-Delete a File with remove\(\)](#)

(see also directory operations; pathnames)

changing file names, [Change Name with rename\(\)](#)

changing ownership, [Change Ownership with chown\(\)](#)

changing permissions, [Change Permissions with chmod\(\)](#)

checking for existence of files, [Check Existence with exists\(\)](#)

checking type of, [Check Type with isfile\(\)](#)

copying files, [Copy with copy\(\)](#)

creating/opening files, [Create or Open with open\(\)](#)

deleting files, [Delete a File with remove\(\)](#)

file operations, [File Operations-Delete a File with remove\(\)](#)

getting pathnames, [Get a Pathname with abspath\(\)](#)

getting symlink pathnames, [Get a symlink Pathname with realpath\(\)](#)

linking files, [Link with link\(\) or symlink\(\)](#)

memory mapping, [Memory Mapping](#)
pathnames, [Pathnames-Use pathlib](#)
practice exercise answers, [14. Files and Directories](#)
practice exercises, [Things to Do](#)
renaming files, [Change Name with rename\(\)](#)
file input/output, [File Input and Output-Change Position with seek\(\)](#)
changing position within a file, [Change Position with seek\(\)-Change Position with seek\(\)](#)
closing files automatically, [Close Files Automatically by Using with](#)
reading binary file with read(), [Read a Binary File with read\(\)](#)
writing binary file with write(), [Write a Binary File with write\(\)](#)
writing text file with print(), [Write a Text File with print\(\)](#)
writing text file with read(), [Read a Text File with read\(\), readline\(\), or readlines\(\)](#)
writing text file with write(), [Write a Text File with write\(\)](#)
file system, defined, [Directory Operations](#)
file, defined, [Files and Directories](#)
file-like object, [BytesIO and StringIO](#)
financial applications, [Python in Finance](#)
find() function, [Search and Select](#)
findall() function, [Find All Matches with findall\(\)](#)
fire-and-forget technique, [Beyond Queues](#)
Flask, [Flask-Pass an argument as part of the URL path](#)
flat file, [File Input and Output](#), [Flat Text Files](#)
floating-point division, [Integer Operations](#)
floats (floating-point numbers), [Floats](#), [Calculate Accurate Floating Point with decimal](#)
folder, defined, [Files and Directories](#)
fonts, Unicode characters and, [Decode](#)
for loops, [Iterate with for and in-Generate Number Sequences with range\(\)](#)
canceling, [Cancel with break](#)
check break use with else, [Check break Use with else](#)
generating number sequences, [Generate Number Sequences with range\(\)](#)
iterating lists, [Iterate with for and in](#)
iteration, [Iterate with for and in-Generate Number Sequences with](#)

[range\(\)](#)

format() function, [New style: {} and format\(\)-New style: {} and format\(\)](#)

formatting

f-strings, [Newest Style: f-strings](#)

new style: { } and format(), [New style: {} and format\(\)-New style: {} and format\(\)](#)

old style (%), [Old style: %-Old style: %](#)

strings, [Formatting-Newest Style: f-strings](#)

fractions, [Perform Rational Arithmetic with fractions](#)

frozenset() function, [Create an Immutable Set with frozenset\(\)](#)

ftplib module, [Other Protocols](#)

full-text databases, [Full-Text Databases](#)

functions, [Functions-Things to Do](#)

(see also methods; specific functions)

anonymous, [Anonymous Functions: lambda](#)

arguments and parameters, [Arguments and Parameters-Mutable and Immutable Arguments](#)

as first-class citizens, [Functions Are First-Class Citizens-Functions Are First-Class Citizens](#)

asynchronous, [Async Functions](#)

calling with parenthesis, [Call a Function with Parentheses](#)

closures, [Closures](#)

decorators and, [Decorators-Decorators](#)

default parameter values, [Specify Default Parameter Values](#)

defining, [Define a Function with def](#)

docstrings, [Docstrings](#)

elements of, [Booleans](#)

generator functions, [Generator Functions](#)

inner functions, [Inner Functions-Closures](#)

keyword arguments and, [Keyword Arguments](#)

methods and, [Methods](#)

namespaces and scope, [Namespaces and Scope-Namespaces and Scope](#)

positional arguments and, [Positional Arguments](#)

practice exercise answers, [9. Functions-9. Functions](#)

practice exercises, [Things to Do](#)

recursion with, [Recursion](#)

G

game development, [Games](#)

garbage collector, [Variables Are Names, Not Places](#)

generator comprehensions, [There Are No Tuple Comprehensions](#),
[Generator Comprehensions](#)

generators, [Generators](#)

generator comprehensions, [There Are No Tuple Comprehensions](#),
[Generator Comprehensions](#)

generator functions, [Generator Functions](#)

geopandas, [Geopandas](#)

get() function, [Get an Item by \[key\] or with get\(\)](#)

getaddrinfo() function, [Domain Name System](#)

gethostbyname() function, [Green Threads and gevent](#), [Domain Name System](#)

getoutput() function, [Create a Process with subprocess](#)

getstatusoutput() function, [Create a Process with subprocess](#)

getter methods, [Getters and Setters](#)

gevent library, [Green Threads and gevent](#)-[Green Threads and gevent](#)
Git, [Git-Git](#)

glob() function, [List Matching Files with glob\(\)](#)

Global Interpreter Lock (GIL), [Threads](#)

global variables, [Namespaces and Scope](#)

globals() function, [Namespaces and Scope](#)

gmtime() function, [Using the time Module](#)

Go (Golang), [Python Versus the Language from Planet X](#)

Google Cloud Platform, [Google Cloud](#)

graph databases, [Graph Databases](#)

graphical user interfaces (GUIs), [Graphical User Interfaces](#)-[Graphical User Interfaces](#)

graphics

2-D, [2-D Graphics-ImageMagick](#)

3-D, [3-D Graphics](#)

3-D animation, [3-D Animation](#)

games, [Games](#)

GUIs, [Graphical User Interfaces-Graphical User Interfaces](#)
plots/graphs/visualization, [Graphical User Interfaces-Graphical User Interfaces](#)
practice exercise answers, [20. Py Art](#)
practice exercises, [Things to Do](#)
graphs, [Plots, Graphs, and Visualization-Bokeh](#)
green threads, [Green Threads and gevent-Green Threads and gevent](#)
gRPC, [gRPC](#)
GTK+, [Graphical User Interfaces](#)

H

Hadoop, [Hadoop](#)
hard drives (HDD), [Storage](#)
hard links, [Link with link\(\) or symlink\(\)](#)
hardware, evolution of computing and, [Caveman Computers-Relative Access Times](#)
hashes, [Redis-Document Databases](#)
HDF5, [HDF5](#)
higher-level languages, [Higher-Level Languages](#)
Houdini, [3-D Animation](#)
HTML (Hypertext Markup Language), [HTML Entities](#), [HTML](#)
httpbin, [Test with httpbin](#)
httpie, [Test with httpie](#)

I

I/O bound, [Concurrency](#)
IDLE, [IDLE](#)
if statements, [Compare with if, elif, and else-Compare with if, elif, and else](#)
ImageMagick, [ImageMagick](#)
immutability, [Mutability](#)
import statement, [Import a Module-Import a Module](#)
imports, relative/absolute, [Relative and Absolute Imports](#)
in (membership operator)
 iterating lists, [Iterate with for and in](#)
 iterating over all items in set, [Iterate with for and in](#)

multiple comparisons with, [Do Multiple Comparisons with in](#)
test for existence of value in list with, [Test for a Value with in](#)
test for value in set with, [Test for a Value with in](#)
testing for keys in a dictionary, [Test for a Key with in](#)
index() function
find list items offset by value, [Find an Item's Offset by Value with index\(\)](#)
for finding offset of substring, [Search and Select](#)
index, in relational databases, [Relational Databases](#)
infinite loop, [Cancel with break](#)
inheritance, [Inheritance-Mixins](#)
adding a method, [Add a Method](#)
calling parent method, [Get Help from Your Parent with super\(\)](#)
from parent class, [Inherit from a Parent Class](#)
mixins, [Mixins](#)
multiple, [Multiple Inheritance-Multiple Inheritance](#)
overriding a method, [Override a Method](#)
initialization, [Assignment](#), [Initialization](#)
inner() function, [Inner Functions-Closures](#), [Closures](#)
insert() function, [Add an Item by Offset with insert\(\)](#)
installation
Anaconda, [Install Anaconda](#)
conda, [Install Anaconda's Package Manager conda](#)
package manager, [Use a Package Manager](#)
packages, [Install Packages-Install from Source](#)
pip, [Use pip](#), [Install the pip Package Manager](#)
pipenv, [Use pipenv](#)
Python 3, [Install Python 3-Install Anaconda's Package Manager conda](#)
SQLAlchemy, [SQLAlchemy](#)
virtualenv, [Use virtualenv](#), [Install virtualenv](#)
instance methods, [Instance Methods](#)
instruction set, [Machine Language](#)
int() function, [Type Conversions-Type Conversions](#)
integer division, [Integer Operations](#)
integer overflow, [How Big Is an int?](#)
integers, [Integers-How Big Is an int?](#)

bases, [Bases](#)

bit-level operators, [Bit Operators](#)

literal, [Literal Integers](#)

operators, [Integer Operations-Integer Operations](#)

precedence of mathematical operations, [Precedence](#)

size in Python 2 versus Python 3, [How Big Is an int?](#)

type conversions, [Type Conversions-Type Conversions](#)

variables and, [Integers and Variables-Integers and Variables](#)

integrated development environments (IDEs), [Integrated Development Environments-JupyterLab](#)

IDLE, [IDLE](#)

IPython, [IPython](#)

Jupyter Notebook, [Jupyter Notebook](#)

JupyterLab, [JupyterLab](#)

PyCharm, [PyCharm](#)

interactive interpreter, [Using the Interactive Interpreter](#), [Create with Quotes](#)

internet (see World Wide Web)

intersection() function, [Combinations and Operators](#)

invoke package, [Invoke](#)

IPython, [IPython](#)

isabs() function, [Check Type with isfile\(\)](#)

isfile() function, [Check Type with isfile\(\)](#)

islink() function, [Link with link\(\) or symlink\(\)](#)

isoformat() function, [The datetime Module](#), [The datetime Module](#)

issubset() function, [Combinations and Operators](#)

issuperset() function, [Combinations and Operators](#)

items() function, [Get All Key-Value Pairs with items\(\)](#)

iteration

code structures, [Iterate over Code Structures with itertools](#)

dictionaries, [Iterate with for and in](#)

lists, [Iterate with for and in](#)

loops, [Iterate with for and in-Generate Number Sequences with range\(\)](#)

multiple sequences, [Iterate Multiple Sequences with zip\(\)](#)

sets, [Iterate with for and in](#)

tuples, [Iterate with for and in](#)

iterators, [Iterate with for and in-Generate Number Sequences with range\(\)](#), [Generators](#)
itertools, [Iterate over Code Structures with itertools](#)

J

Java, [Python Versus the Language from Planet X](#)
join() function, [Combine by Using join\(\)](#), [Convert a List to a String with join\(\)](#)
JSON (JavaScript Object Notation), [A Bigger Program](#), [JSON-JSON](#)
JSON-RPC, [JSON RPC](#)
Jupyter Notebook, [Jupyter Notebook](#)
JupyterLab, [JupyterLab](#)
justification, [Alignment](#)

K

key-value stores (see NoSQL data stores)
keys
 copying, [Combine Dictionaries with update\(\)](#), [Copy with copy\(\)](#)
 deleting items by, [Delete an Item by Key with del](#)
 dictionaries and, [Dictionaries](#)
 getting all, [Get All Keys with keys\(\)](#)
 getting item by, [Get an Item by Key and Delete It with pop\(\)](#)
 handling missing, [Handle Missing Keys with setdefault\(\) and defaultdict\(\)-Handle Missing Keys with setdefault\(\) and defaultdict\(\)](#)
 sets and, [Sets](#)
keys() function, [Get All Keys with keys\(\)](#)
keyword arguments, [Keyword Arguments](#)
 exploding/gathering, [Explode/Gather Keyword Arguments with **](#)
 in named tuples, [Named Tuples](#)
keyword-only arguments, [Keyword-Only Arguments](#)
Kivy, [Graphical User Interfaces](#)
Kubernetes, [Kubernetes](#), [Kubernetes](#)

L

lambda functions, [Anonymous Functions: lambda](#)

layers, [TCP/IP](#)

leap years, [Leap Year](#)

left justification, [Alignment](#)

len() function

counting key-value pairs in dictionary, [Get Length with len\(\)](#)

counting Unicode characters with, [Python 3 Unicode Strings](#)

determining string length with, [Get Length with len\(\)](#)

get list length with, [Get Length with len\(\)](#)

getting length of set with, [Get Length with len\(\)](#)

libraries (see specific libraries)

LibreOffice, [The Microsoft Office Suite](#)

LIFO (last in, first out) queue, [Get an Item by Offset and Delete It with pop\(\)](#)

linear algebra, [Linear Algebra](#)

link() function, [Link with link\(\) or symlink\(\)](#)

links, [Link with link\(\) or symlink\(\)](#)

Linux

package manager, [Use a Package Manager](#)

Python 3 installation, [Linux or Unix](#)

shell, [Python Versus the Language from Planet X](#)

list comprehension, [Create a List with a Comprehension](#)-[Create a List with a Comprehension](#)

list() function

copying lists with, [Copy with copy\(\), list\(\), or a Slice](#)

creating empty list with, [Create or Convert with list\(\)](#)

creating/converting lists with, [Create or Convert with list\(\)](#)

dictionaries and, [Get All Keys with keys\(\)](#)

listdir() function, [List Contents with listdir\(\)](#)

lists, [Lists-Lists of Lists](#)

adding item by offset to, [Add an Item by Offset with insert\(\)](#)

adding item to end, [Add an Item to the End with append\(\)](#)

assign values to sublist with slice, [Change Items with a Slice](#)

assigning to more than one variable, [Assign with =](#)

changing item by [offset], [Change an Item by \[offset \]](#)

combining, [Combine Lists by Using extend\(\) or +](#)

comparing, [Compare Lists](#)

convert to string, [Convert a List to a String with join\(\)](#)

copying with `copy()`, `list()`, or `slice`, [Copy with copy\(\), list\(\), or a Slice](#)

copying with `deepcopy()`, [Copy Everything with deepcopy\(\)](#)

count occurrences of a value, [Count Occurrences of a Value with count\(\)](#)

create with a comprehension, [Create a List with a Comprehension-Create a List with a Comprehension](#)

creating from string, [Create from a String with split\(\)](#)

creating with `[]`, [Create with \[\]](#)

creating/converting with `list()`, [Create or Convert with list\(\)](#)

defined, [Copying](#)

delete all items, [Delete All Items with clear\(\)](#)

delete an item by offset with `del`, [Delete an Item by Offset with del](#)

delete an item by value, [Delete an Item by Value with remove\(\)](#)

duplicate all items with `*`, [Duplicate All Items with *](#)

extracting single value from, [Get an Item by \[offset \]](#)

extracting subsequence from, [Get Items with a Slice](#)

find items offset by value, [Find an Item's Offset by Value with index\(\)](#)

get item by offset and delete, [Get an Item by Offset and Delete It with pop\(\)](#)

get length, [Get Length with len\(\)](#)

in Redis, [Lists](#)

iterating multiple sequences, [Iterate Multiple Sequences with zip\(\)](#)

iterating with `for` and `in`, [Iterate with for and in](#)

of lists, [Lists of Lists](#)

practice exercise answers, [7. Tuples and Lists-7. Tuples and Lists](#)

practice exercises, [Things to Do](#)

reordering items, [Reorder Items with sort\(\) or sorted\(\)](#)

test for existence of value in, [Test for a Value with in](#)

tuples versus, [Tuples Versus Lists](#)

literal integers, [Literal Integers](#)

literal values, [Literal Values](#)

`load()` function, [YAML](#)

`locals()` function, [Namespaces and Scope](#)

`localtime()` function, [Using the time Module](#)

logging, [Log Error Messages-Log Error Messages](#)

logical (boolean) operators, [Compare with if, elif, and else](#)

lookup() function, [Python 3 Unicode Strings](#)
loops, [Loop with while and for-Things to Do](#)
canceling, [Cancel with break](#)
coroutines and event loops, [Coroutines and Event Loops-Coroutines and Event Loops](#)
for loops, [Iterate with for and in-Generate Number Sequences with range\(\)](#)
iterating lists, [Iterate with for and in](#)
iterating over, [Iterate with for and in-Generate Number Sequences with range\(\)](#)
practice exercise answers, [6. Loop with while and for-6. Loop with while and for](#)
practice exercises, [Things to Do](#)
skipping ahead, [Skip Ahead with continue](#)
while loops, [Repeat with while-Check break Use with else](#)

M

machine language, [Machine Language](#)
macOS
package managers, [Use a Package Manager](#)
path separators, [Pathnames](#)
Python 3 installation, [macOS-macOS](#)
magic methods, [Magic Methods-Magic Methods](#)
magnetic disks, [Storage](#)
MapReduce, [Big Fat Data](#)
maps, [Maps-Applications and Data](#)
applications and data, [Applications and Data](#)
formats, [Formats](#)
geopandas and, [Geopandas](#)
packages, [Other Mapping Packages](#)
shapefiles and, [Draw a Map from a Shapefile-Draw a Map from a Shapefile](#)
marshalling (see data serialization)
match() function
exact matches, [Find Exact Beginning Match with match\(\)](#)
specifying output, [Patterns: Specifying match\(\) Output](#)

math functions, [Math Functions-Math Functions](#)
mathematics, [Math and Statistics in the Standard Library-Matrix Multiplication](#)
complex numbers, [Working with Complex Numbers](#)
floating points with decimals, [Calculate Accurate Floating Point with decimal](#)
math functions, [Math Functions-Math Functions](#)
matrix multiplication, [Matrix Multiplication](#)
operator precedence, [Precedence](#), [Compare with if, elif, and else](#), [Operator Precedence](#)
packed sequences with array, [Use Packed Sequences with array](#)
Pandas, [Pandas](#)
rational arithmetic with fractions, [Perform Rational Arithmetic with fractions](#)
SciPy, [SciPy](#)
statistics module, [Handling Simple Stats with statistics](#)
matplotlib, [Matplotlib-Matplotlib](#)
matrix multiplication, [Matrix Multiplication](#)
Maya, [3-D Animation](#)
membership operator (see in (membership operator))
memcached server, [Memcached](#)
memory, [Memory and Caches](#)
memory mapping, [Memory Mapping](#), [Padded Binary Files and Memory Mapping](#)
Mercurial, [Mercurial](#)
MessagePack RPC, [MessagePack RPC](#)
methods, [Methods](#)
adding, [Add a Method](#)
class methods, [Class Methods](#)
finding with self argument, [In self Defense](#)
instance methods, [Instance Methods](#)
magic methods, [Magic Methods-Magic Methods](#)
method types, [Method Types-Static Methods](#)
overriding, [Override a Method](#)
resolution order, [Multiple Inheritance](#)
static methods, [Static Methods](#)
Microsoft Azure, [Microsoft Azure](#)

Microsoft Office Suite, [The Microsoft Office Suite](#)
mixins, [Mixins](#)
mkdir() function, [Create with mkdir\(\)](#)
modules, [Modules and the import Statement-Import Only What You Want from a Module](#)
(see also specific modules)
classes and objects versus, [When to Use Objects or Something Else](#)
importing, [Import a Module-Import a Module](#)
importing specific part of, [Import Only What You Want from a Module](#)
importing with alias, [Import a Module with Another Name](#)
packages and, [Modules Versus Objects](#)
practice exercise answers, [11. Modules, Packages, and Goodies](#)
practice exercises, [Things to Do](#)
most_common () function, [Count Items with Counter\(\)](#)
mro() method, [Multiple Inheritance](#)
multiplication, [Integer Operations](#)
multiprocessing module, [Create a Process with multiprocessing](#)
music, [Audio and Music](#)
mutability, [Mutability](#)
MySQL, [MySQL](#)

N

name() function, [Python 3 Unicode Strings](#)
named tuples, [Named Tuples](#)
namespace packages, [Namespace Packages](#)
namespaces, [Namespaces and Scope-Namespaces and Scope](#)
ndarray, [NumPy](#)
Netcat, [Netcat](#)
networks/networking, [Data in Space: Networks-Things to Do](#)
basics, [Distributed Computing and Networks](#)
data serialization, [Data Serialization-Other Serialization Formats](#)
Docker, [Docker](#)
Domain Name System and, [Domain Name System](#)
email modules, [Python Email Modules](#)
for big data, [Big Fat Data](#)

internet services, [Internet Services](#)

patterns for, [Networking Patterns-Other Pub-Sub Tools](#)

practice exercise answers, [17. Data in Space: Networks-17. Data in Space: Networks](#)

practice exercises, [Things to Do](#)

publish-subscribe model, [The Publish-Subscribe Pattern-Other Pub-Sub Tools](#)

remote management tools, [Remote Management Tools](#)

Remote Procedure Calls, [Remote Procedure Calls-Twirp](#)

request-reply patterns, [The Request-Reply Pattern-Other Messaging Tools](#)

web services/APIs, [Web Services and APIs](#)

NGINX web server, [NGINX](#)

None value, [None Is Useful](#)

nonvolatile storage, [Storage](#)

normalization, of Unicode characters, [Normalization](#)

nose, [Test with nose](#)

NoSQL data stores, [NoSQL Data Stores-Other NoSQL](#)

dbm family, [The dbm Family](#)

document databases, [Document Databases](#)

graph databases, [Graph Databases](#)

memcached, [Memcached](#)

Redis, [Redis-Document Databases](#)

time series databases, [Time Series Databases](#)

now() function, [The datetime Module](#)

null set, [Sets](#)

Numba, [Numba](#)

numbers, [Numbers-Things to Do](#)

booleans and, [Numbers-Things to Do](#)

floating-point, [Floats](#)

integers, [Integers-How Big Is an int?](#)

practice exercise answers, [3. Numbers](#)

practice exercises, [Things to Do](#)

NumPy, [NumPy-Linear Algebra](#)

array math, [Array Math](#)

arrays with arange() function, [Make an Array with arange\(\)](#)

arrays with array() function, [Make an Array with array\(\)](#)

arrays with zeros()/ones()/random(), [Make an Array with zeros\(\), ones\(\), or random\(\)](#)
change array shape, [Change an Array's Shape with reshape\(\)](#)
getting elements from array, [Get an Element with \[\]](#)
linear algebra functions, [Linear Algebra](#)

O

objects

accessing attributes, [Attribute Access-Class and Object Attributes](#)
adding methods, [Add a Method](#)
attributes, [Attributes](#)
calling parent methods, [Get Help from Your Parent with super\(\)](#)
class attribute assignment and, [Class and Object Attributes](#)
class definition for, [Define a Class with class](#)
classes versus, [When to Use Objects or Something Else](#)
composition, [Aggregation and Composition](#)
data as, [Python Data Are Objects](#)
defined, [What Are Objects?](#)
finding attributes/methods, [In self Defense](#)
inheritance and, [Inheritance-Mixins](#)
initialization, [Initialization](#)
magic methods, [Magic Methods-Magic Methods](#)
method types, [Method Types-Static Methods](#)
methods and, [Methods](#)
modules versus, [When to Use Objects or Something Else](#)
named tuples, [Named Tuples](#)
overriding methods, [Override a Method](#)
packages and, [Modules Versus Objects](#)
polymorphism in, [Duck Typing](#)
practice exercise answers, [10. Oh Oh: Objects and Classes-10. Oh Oh: Objects and Classes](#)
practice exercises, [Things to Do](#)
private attributes, [Getters and Setters](#)
self argument, [In self Defense](#)
simple, [Simple Objects-Initialization](#)
special methods, [Magic Methods-Magic Methods](#)

ODM (Object Data Manager/Object Document Mapper), [Document Databases](#)

offset, [Little Programs](#)

adding item to list by, [Add an Item by Offset with insert\(\)](#)

changing list item by, [Change an Item by \[offset \]](#)

character extraction from string, [Get a Character with \[\]](#)

delete a list item by, [Delete an Item by Offset with del](#)

extracting single value from list with, [Get an Item by \[offset \]](#)

find() function, [Search and Select](#)

getting list item and deleting, [Get an Item by Offset and Delete It with pop\(\)](#)

index() function, [Search and Select](#), [Find an Item's Offset by Value with index\(\)](#)

seek() function and, [Change Position with seek\(\)-Change Position with seek\(\)](#)

substring extraction with slice, [Get a Substring with a Slice-Get a Substring with a Slice](#)

tell() function, [Change Position with seek\(\)](#)

ones() function, [Make an Array with zeros\(\), ones\(\), or random\(\)](#)

opcodes, [Machine Language](#)

open() function, [Create or Open with open\(\)](#)

OpenOffice, [The Microsoft Office Suite](#)

OpenStack, [OpenStack](#)

operating systems (generally), [Operating Systems](#)

operator precedence, [Precedence](#), [Compare with if, elif, and else](#),

[Operator Precedence](#)

optimization

Cython/NumPy/C extensions for, [Cython, NumPy, and C Extensions](#)

Numba for, [Numba](#)

of code, [Optimize-Numba](#)

PyPI for, [PyPy](#)

timing, [Measure Timing-Measure Timing](#)

OrderedDict() function, [Order by Key with OrderedDict\(\)](#)

ORM (Object-Relational Mapper), [The Object-Relational Mapper \(ORM\)-The Object-Relational Mapper \(ORM\)](#)

os module, [Check Type with isfile\(\)](#), [Get System Info with os](#)

os.path.join() function, [Build a Pathname with os.path.join\(\)](#)

package managers, [Use a Package Manager](#)
packages, [Packages-Modules Versus Objects](#)
(see also specific packages)
installing, [Install Packages-Install from Source](#)
installing from source, [Install from Source](#)
module search path, [The Module Search Path](#)
modules versus objects, [Modules Versus Objects](#)
namespace packages, [Namespace Packages](#)
practice exercise answers, [11. Modules, Packages, and Goodies](#)
practice exercises, [Things to Do](#)
relative/absolute imports, [Relative and Absolute Imports](#)
third-party for science and math, [Scientific Python-Pandas](#)
packed sequences, [Use Packed Sequences with array](#)
padded binary files, [Padded Binary Files and Memory Mapping](#)
padded text files, [Padded Text Files](#)
padding characters, removing, [Strip with strip\(\)](#)
Panda3D, [3-D Animation](#)
pandas, [Pandas](#)
Pandas, [Pandas](#)
parameters
 default values for, [Specify Default Parameter Values](#)
 defined, [Arguments and Parameters](#)
parent class, [Inherit from a Parent Class](#)
parent method, [Get Help from Your Parent with super\(\)](#)
path separators, [Pathnames](#)
pathlib module, [Use pathlib](#)
pathnames, [Check Type with isfile\(\)](#), [Pathnames-Use pathlib](#)
 building, [Build a Pathname with os.path.join\(\)](#)
 getting, [Get a Pathname with abspath\(\)](#)
 getting symlink pathnames, [Get a symlink Pathname with realpath\(\)](#)
 pathlib module, [Use pathlib](#)
patterns, in searches (see regular expressions)
pdb (Python debugger), [Use pdb-Use pdb](#)
PDF, [The Microsoft Office Suite](#)

pep8, [Check with pylint, pyflakes, flake8, or pep8](#)
per cent sign (%), [Old style: %-Old style: %](#)
Perl, [Python Versus the Language from Planet X](#)
permissions, [Change Permissions with chmod\(\)](#)
persistence (term), [Data in a Box: Persistent Storage](#)
persistent storage, [Data in a Box: Persistent Storage-Things to Do](#)
binary files, [Binary Files](#)
flat text files, [Flat Text Files](#)
full-text databases, [Full-Text Databases](#)
nonvolatile, [Storage](#)
NoSQL data stores, [NoSQL Data Stores-Other NoSQL](#)
padded text files, [Padded Text Files](#)
practice exercise answers, [16. Data in a Box: Persistent Storage-16. Data in a Box: Persistent Storage](#)
practice exercises, [Things to Do](#)
relational databases, [Relational Databases-Other Database Access Packages](#)
tabular text files, [Tabular Text Files-Configuration Files](#)
pex files, [Distribute Your Programs](#)
pickle module, [Serialize with pickle](#)
Pillow, [PIL and Pillow-PIL and Pillow](#)
pip, [Use pip](#)
conda and, [Install Anaconda's Package Manager conda](#)
installation, [Install the pip Package Manager](#)
pipenv, [Use pipenv](#), [Other Packaging Solutions](#)
plotting, [Plots, Graphs, and Visualization-Bokeh](#), [Matplotlib-Matplotlib](#)
PNG files, [Convert Binary Data with struct-Convert Binary Data with struct](#)
poetry, [Other Packaging Solutions](#)
polymorphism, [Duck Typing](#)
pop() function, [Get an Item by Offset and Delete It with pop\(\)](#), [Get an Item by Key and Delete It with pop\(\)](#)
positional arguments, [Positional Arguments](#), [Explode/Gather Positional Arguments with *-Explode/Gather Positional Arguments with *](#)
PostgreSQL, [PostgreSQL](#)
pound sign (#), [Comment with #](#)

pprint() function, [Print Nicely with pprint\(\)](#)

precedence, [Precedence](#), [Compare with if, elif, and else](#), [Operator Precedence](#)

pretty printer, [Print Nicely with pprint\(\)](#)

primary key, [Relational Databases](#)

print() function

debugging and, [Use print\(\)](#)

quotes with, [Create with Quotes](#)

write() versus, [Write a Text File with write\(\)](#)

writing text file with, [Write a Text File with print\(\)](#)

processes

concurrency and, [Processes](#)

creating with multiprocessing module, [Create a Process with multi-processing](#)

creating with subprocess module, [Create a Process with subprocess](#)
killing, [Kill a Process with terminate\(\)](#)

practice exercise answers, [15. Data in Time: Processes and Concurrency](#)

practice exercises, [Things to Do](#)

process information, [Get Process Info with psutil](#)

system information, [Get System Info with os](#)

programs (generally), [Mysteries](#)

(see also processes)

elements of, [Mysteries](#)

example of complex program, [A Bigger Program-A Bigger Program](#)

example of simple program, [Little Programs-Little Programs](#)

proper subsets, [Combinations and Operators](#)

proper superset, [Combinations and Operators](#)

properties

computed values returned by, [Properties for Computed Values](#)

for attribute privacy, [Properties for Attribute Access-Properties for Attribute Access](#)

getting/setting attribute values, [Attribute Access-Class and Object Attributes](#)

psutil package, [Get Process Info with psutil](#)

publish-subscribe (pub-sub) pattern, [The Publish-Subscribe Pattern-Other Pub-Sub Tools](#)

defined, [Networking Patterns](#)

Redis, [Redis-Redis](#)

ZeroMQ, [ZeroMQ](#)

pull pattern, [Networking Patterns](#)

Puppet, [Remote Management Tools](#)

push pattern, [Networking Patterns](#)

PyCharm, [PyCharm](#)

pyflakes, [Check with pylint, pyflakes, flake8, or pep8-Check with pylint, pyflakes, flake8, or pep8](#)

pylint, [Check with pylint, pyflakes, flake8, or pep8-Check with pylint, pyflakes, flake8, or pep8](#)

PyPI, [PyPy](#)

PySimpleGUI, [Graphical User Interfaces](#)

Python (generally)

basics, [A Taste of Py-Things to Do](#)

interactive interpreter and, [Using the Interactive Interpreter](#)

language style, [Your Moment of Zen](#)

noninteractive programs, [Using Python Files](#)

other languages compared to, [Python Versus the Language from Planet X-Python Versus the Language from Planet X](#)

practice exercise answers, [1. A Taste of Py](#)

practice exercises, [Things to Do](#)

Python 2 versus Python 3, [Python 2 Versus Python 3](#)

real-world applications, [Python in the Real World](#)

reasons to use, [Why Python?-Why Python?](#)

running, [Running Python-What's Next?](#)

situations not to use, [Why Not Python?](#)

Zen of, [Your Moment of Zen](#)

Python 2, integer limits in, [How Big Is an int?](#)

Python 3

business applications, [Py at Work-Things to Do](#)

dataclasses, [Dataclasses](#)

installation, [Install Python 3-Install Anaconda's Package Manager conda](#)

integer size in, [How Big Is an int?](#)

Linux installation, [Linux or Unix](#)

macOS installation, [macOS-macOS](#)

math/statistics applications, [Py Sci-Things to Do](#)

resources for learning, [How You Can Learn More-Getting a Python Job](#)

Unicode text strings, [Python 3 Unicode Strings-Python 3 Unicode Strings](#)

Unix installation, [Linux or Unix](#)

Windows installation, [Windows](#)

Python community, [Groups](#)

Python debugger (pdb), [Use pdb-Use pdb](#)

Python Image Library (PIL), [PIL and Pillow-PIL and Pillow](#)

Python Package Index (PyPI), [Find Python Code](#)

Python standard library, [Goodies in the Python Standard Library-Get Random](#)

benefits of, [Goodies in the Python Standard Library](#)

counting items, [Count Items with Counter\(\)](#)

deque, [Stack + Queue == deque](#)

finding code, [Find Python Code](#)

handling missing keys, [Handle Missing Keys with setdefault\(\) and defaultdict\(\)-Handle Missing Keys with setdefault\(\) and defaultdict\(\)](#)

iterating over code structures, [Iterate over Code Structures with itertools](#)

order by key, [Order by Key with OrderedDict\(\)](#)

pretty printer, [Print Nicely with pprint\(\)](#)

third-party alternatives to, [More Batteries: Get Other Python Code](#)

Q

Qt, [Graphical User Interfaces](#)

queues, [Queues](#)

quotes, [Create with Quotes-Create with Quotes](#)

R

RAM (Random Access Memory), [Data in a Box: Persistent Storage, Memory and Caches](#)

randint() function, [Get Random](#)

random numbers, [Get Random](#)

random() function, [Make an Array with zeros\(\), ones\(\), or random\(\)](#)

randrange() function, [Get Random](#)

range() function, [Generate Number Sequences with range\(\)](#), [Get Random](#)

rational arithmetic, [Perform Rational Arithmetic with fractions](#)

raw string, [Create with Quotes](#), [Escape with \](#)

read() function, [Read a Text File with read\(\), readline\(\), or readlines\(\)](#),
[Read a Binary File with read\(\)](#)

reader() function, [CSV](#)

readline() function, [Read a Text File with read\(\), readline\(\), or readlines\(\)](#)

readlines() function, [Read a Text File with read\(\), readline\(\), or readlines\(\)](#)

realpath() function, [Get a symlink Pathname with realpath\(\)](#)

record, defined, [Data in a Box: Persistent Storage](#)

recursion, [Recursion](#)

Redis, [Redis-Document Databases](#)

- cached/expiration, [Caches and expiration](#)

- concurrency and, [Redis-Redis](#)

- hashes, [Hashes](#)

- lists, [Lists](#)

- publish-subscribe system with, [Redis-Redis](#)

- sets, [Sets](#)

- sorted sets, [Sorted sets](#)

- strings, [Strings](#)

regressions, [Test with unittest](#)

regular expressions, [Text Strings: Regular Expressions-Patterns: Specifying match\(\) Output](#)

- all matches, [Find All Matches with findall\(\)](#)

- basics, [Text Strings: Regular Expressions](#)

- exact matches, [Find Exact Beginning Match with match\(\)](#)

- first match, [Find First Match with search\(\)](#)

- pattern specifiers, [Patterns: Using Specifiers-Patterns: Using Specifiers](#)

- replace at match, [Replace at Matches with sub\(\)](#)

- special characters, [Patterns: Special Characters-Patterns: Special Characters](#)

specifying match output, [Patterns: Specifying match\(\) Output](#)
split at matches, [Split at Matches with split\(\)](#)
relational databases, [Relational Databases-Other Database Access Packages](#)
DB-API, [DB-API](#)
MySQL, [MySQL](#)
PostgreSQL, [PostgreSQL](#)
SQL, [SQL-SQL](#)
SQLAlchemy, [SQLAlchemy-The Object-Relational Mapper \(ORM\)](#)
SQLite, [SQLite-SQLite](#)
relative access times, [Relative Access Times](#)
relative imports, [Relative and Absolute Imports](#)
relative/absolute imports, [Relative and Absolute Imports](#)
Remote Procedure Calls (RPCs), [Remote Procedure Calls-Twirp](#)
gRPC, [gRPC](#)
JSON-RPC, [JSON RPC](#)
MessagePack RPC, [MessagePack RPC](#)
Twirp, [Twirp](#)
XML RPC, [XML RPC](#)
Zerorpc, [Zerorpc](#)
remove() function
delete an item from list by value, [Delete an Item by Value with remove\(\)](#)
deleting file with, [Delete a File with remove\(\)](#)
deleting value from set, [Delete an Item with remove\(\)](#)
rename() function, [Change Name with rename\(\)](#)
replace() function, [Substitute by Using replace\(\)](#)
Representational State Transfer (REST), [Web APIs and REST](#)
request-reply patterns, [The Request-Reply Pattern-Other Messaging Tools](#)
request-reply technique, [Beyond Queues](#)
requests module, [Beyond the Standard Library: requests](#)
requests-html library, [Requests-HTML](#)
reshape() function, [Change an Array's Shape with reshape\(\)](#)
right justification, [Alignment](#)
rmdir() function, [Delete with rmdir\(\)](#)
rows, [Relational Databases](#)

Ruby, [Python Versus the Language from Planet X](#)

Ruby on Rails, [Python Versus the Language from Planet X](#)

Rust, [Python Versus the Language from Planet X](#)

S

s (step), [Use pdb](#)

safe_load () function, [YAML](#)

Salt, [Remote Management Tools](#)

sample() function, [Get Random](#)

scapy library, [Scapy](#)

scientific applications, [Py Sci-Things to Do](#)

domain-specific libraries, [Python and Scientific Areas](#)

NumPy, [NumPy-Linear Algebra](#)

practice exercise answers, [22. PySci](#)

practice exercises, [Things to Do](#)

SciKit, [SciKit](#)

SciPy, [SciPy](#)

SciKit library, [SciKit](#)

SciPy library, [SciPy](#)

scraping, [Crawl and Scrape-Requests-HTML](#)

scrapy, [Scrapy](#)

Seaborn, [Seaborn](#)

search() function, [Find First Match with search\(\)](#)

secondary index, [Relational Databases](#)

security issues

business data, [Business Data Security](#)

debug in production web servers, [Flask](#)

pickle module, [Serialize with pickle](#)

SQL injection, [SQLite](#)

XML and, [An XML Security Note](#)

seek() function, [Change Position with seek\(\)-Change Position with seek\(\), Padded Text Files](#)

self argument, [In self Defense](#)

sentinel value, [Redis](#)

sequences, [Generate Number Sequences with range\(\)](#)

(see also lists; text strings)

converting into dictionaries, [Convert with dict\(\)](#)
creating with generators, [Generators](#)
deque and, [Stack + Queue == deque](#)
escape sequences, [Escape with \](#)
extracting subsequences from lists, [Get Items with a Slice](#)
generating number sequences, [Generate Number Sequences with range\(\)](#)
iterating over multiple, [Iterate Multiple Sequences with zip\(\)](#)
packed, [Use Packed Sequences with array](#)
strings as, [Text Strings](#)
tuples, [Tuples-Modify a Tuple](#)
serialization (see data serialization)
servers, [Distributed Computing and Networks](#)
set intersection operator (&), [Combinations and Operators-Combinations and Operators](#)
set() function, [Create with set\(\)](#)
setdefault() function, [Handle Missing Keys with setdefault\(\) and defaultdict\(\)-Handle Missing Keys with setdefault\(\) and defaultdict\(\)](#)
setlocale() function, [Read and Write Dates and Times-Read and Write Dates and Times](#)
sets, [Sets-Create an Immutable Set with frozenset\(\)](#)
 adding items to, [Add an Item with add\(\)](#)
 checking for combinations of set values, [Combinations and Operators-Combinations and Operators](#)
 comprehensions, [Set Comprehensions](#)
 conversion to, [Convert with set\(\)](#)
 creating, [Create with set\(\)](#)
 deleting items from, [Delete an Item with remove\(\)](#)
 dictionaries versus, [Sets](#)
 getting length of, [Get Length with len\(\)](#)
 immutable, [Create an Immutable Set with frozenset\(\)](#)
 in Redis, [Sets](#)
 iterating over all items in, [Iterate with for and in](#)
 null/empty, [Sets](#)
 practice exercise answers, [8. Dictionaries-8. Dictionaries](#)
 practice exercises, [Things to Do](#)
 sorted, [Sorted sets](#)

testing for values, [Test for a Value with in](#)
setter methods, [Getters and Setters](#)
setUp() function, [Test with unittest](#)
shapefile
about, [Draw a Map from a Shapefile-Draw a Map from a Shapefile](#)
drawing a map from, [Formats](#)
shell program, [Python Versus the Language from Planet X](#)
slash (/), [Pathnames](#)
slice
assign values to sublist with, [Change Items with a Slice](#)
copying lists with, [Copy with copy\(\), list\(\), or a Slice](#)
extracting subsequence from lists with, [Get Items with a Slice](#)
reversing a string with, [Stack + Queue == deque](#)
substring extraction with, [Get a Substring with a Slice-Get a Substring with a Slice](#)
social media sites, APIs and, [Web Services and APIs](#)
sockets, [Sockets-Sockets](#)
software (generally)
assembler, [Assembler](#)
basics, [Software-Kubernetes](#)
bits/bytes, [In the Beginning Was the Bit](#)
cloud computing, [The Cloud](#)
containers, [Containers](#)
distributed computing/networks, [Distributed Computing and Networks](#)
higher-level languages, [Higher-Level Languages](#)
Kubernetes, [Kubernetes](#)
machine language, [Machine Language](#)
operating systems, [Operating Systems](#)
virtual machines, [Virtual Machines](#)
sort() function, [Reorder Items with sort\(\) or sorted\(\)](#)
sorted() function, [Reorder Items with sort\(\) or sorted\(\)](#)
source control systems, [Source Control-Git](#)
Git, [Git-Git](#)
Mercurial, [Mercurial](#)
Spark, [Spark](#)
special (magic) methods, [Magic Methods-Magic Methods](#)

special characters, [Patterns: Special Characters](#)-[Patterns: Special Characters](#)

split() function, [Split with split\(\)](#), [Create from a String with split\(\)](#), [Split at Matches with split\(\)](#)

spreadsheets, [Spreadsheets](#), [Processing Business Data](#)

SQL (structured query language), [SQL-SQL](#)

SQLAlchemy, [SQLAlchemy-The Object-Relational Mapper \(ORM\)](#)

engine layer, [The engine layer](#)

Expression Language, [The SQL Expression Language](#)

installing, [SQLAlchemy](#)

Object-Relational Mapper, [The Object-Relational Mapper \(ORM\)](#)-[The Object-Relational Mapper \(ORM\)](#)

SQLite, [SQLite-SQLite](#)

SSD (Solid State Drive), [Storage](#)

stack (LIFO queue), [Get an Item by Offset and Delete It with pop\(\)](#)

static languages, [Python Versus the Language from Planet X](#)

static methods, [Static Methods](#)

statistics applications, [Math and Statistics in the Standard Library](#)-[Matrix Multiplication](#), [SciPy](#)

statistics module, [Handling Simple Stats with statistics](#)

storage (see persistent storage)

str() function, [Create with str\(\)](#)

strftime() function, [Read and Write Dates and Times](#)

string methods, [String Methods](#)

string module, [String Module Attributes](#)

StringIO, [BytesIO and StringIO](#)-[BytesIO and StringIO](#)

strings, [Text Strings-Things to Do](#)

case, [Case](#)

combining, [Combine by Using +](#), [Combine by Using join\(\)](#)

converting to list, [Convert a List to a String with join\(\)](#)

creating with quotes, [Create with Quotes](#)-[Create with Quotes](#)

creating with str(), [Create with str\(\)](#)

defined, [Little Programs](#)

determining length of, [Get Length with len\(\)](#)

duplicating, [Duplicate with *](#)

escape sequences and, [Escape with \](#)

formatting, [Formatting-Newest Style: f-strings](#)

getting single character from, [Get a Character with \[\]](#)
in Redis, [Strings](#)
practice exercise answers, [5. Text Strings-5. Text Strings](#), [12. Wrangle and Mangle Data-12. Wrangle and Mangle Data](#)
practice exercises, [Things to Do](#), [Things to Do](#)
regular expressions, [Text Strings: Regular Expressions-Patterns: Specifying match\(\) Output](#)
searching and selecting, [Search and Select-Search and Select](#)
splitting, [Split with split\(\)](#)
string methods, [String Methods](#)
stripping padding characters from, [Strip with strip\(\)](#)
substring extraction with slice, [Get a Substring with a Slice-Get a Substring with a Slice](#)
substring substitution, [Substitute by Using replace\(\)](#)
Unicode, [Text Strings: Unicode-For More Information](#)
strip() function, [Strip with strip\(\)](#)
strong typing, [Mutability](#)
strptime() function, [Read and Write Dates and Times](#)
struct module, [Convert Binary Data with struct-Convert Binary Data with struct](#)
struct_time objects, [Using the time Module](#)
sub() function, [Replace at Matches with sub\(\)](#)
subdirectories, creating, [List Contents with listdir\(\)](#)
sublists, [Change Items with a Slice](#)
subprocess module, [Create a Process with subprocess](#)
subsets, [Combinations and Operators](#)
substrings
 extraction with slice, [Get a Substring with a Slice-Get a Substring with a Slice](#)
 substitution with replace(), [Substitute by Using replace\(\)](#)
subtraction, [Integer Operations](#)
sum() function, [Functions Are First-Class Citizens](#)
super() function, [Get Help from Your Parent with super\(\)](#)
superset, [Combinations and Operators](#)
symbolic link, [Link with link\(\) or symlink\(\)](#)
symlink() function, [Link with link\(\) or symlink\(\)](#)
symmetric_difference () function, [Combinations and Operators](#)

synchronous (term), [Concurrency](#)
system() function, [Get System Info with os](#)

T

tables, [Relational Databases](#)

tablib, [Tablib](#)

tabular text files, [Tabular Text Files-Configuration Files](#)

configuration files, [Configuration Files](#)

CSV, [CSV-CSV](#)

HTML, [HTML](#)

JSON, [JSON-JSON](#)

pandas, [Pandas](#)

tablib, [Tablib](#)

XML, [XML-An XML Security Note](#)

YAML, [YAML-YAML](#)

TCP (Transmission Control Protocol), [TCP/IP](#), [Sockets-Sockets](#)

TCP/IP (Transmission Control Protocol/Internet Protocol), [TCP/IP-Netcat](#), [Web Clients](#)

Netcat and, [Netcat](#)

scapy library, [Scapy](#)

sockets, [Sockets-Sockets](#)

tearDown() function, [Test with unittest](#)

tell() function, [Change Position with seek\(\)](#)

telnet, [Test with telnet](#)

terminate() function, [Kill a Process with terminate\(\)](#)

testing, [Test-Continuous Integration](#)

continuous integration, [Continuous Integration](#)

doctest, [Test with doctest](#)

nose, [Test with nose](#)

Python code checkers, [Check with pylint, pyflakes, flake8, or pep8-Check with pylint, pyflakes, flake8, or pep8](#)

unittest, [Test with unittest-Test with unittest](#)

text files

configuration files, [Configuration Files](#)

CSV, [CSV-CSV](#)

flat, [Flat Text Files](#)

HTML, [HTML](#)

JSON, [JSON-JSON](#)

pandas, [Pandas](#)

tablib, [Tablib](#)

tabular, [Tabular Text Files-Configuration Files](#)

XML, [XML-An XML Security Note](#)

YAML, [YAML-YAML](#)

text strings (see strings)

threads, [Threads-Threads](#), [Green Threads and gevent-Green Threads and gevent](#)

3-D graphics, [3-D Graphics](#)

3-D animation, [3-D Animation](#)

throttling, [Beyond Queues](#)

TileDB, [TileDB](#)

time (see calendars/clocks)

time module, [Using the time Module-Using the time Module](#)

time series databases, [Time Series Databases](#)

time zones, UTC versus, [Using the time Module](#)

time-to-live, [Redis-Document Databases](#)

timedelta object, [The datetime Module](#)

timeit() function, [Measure Timing](#)

timing, measurement of, [Measure Timing-Measure Timing](#)

today() function, [The datetime Module](#)

true values, [What Is True?](#)

tuple() conversion function, [Create with tuple\(\)](#)

tuples, [Tuples-Modify a Tuple](#)

combining, [Combine Tuples by Using +](#)

comparing, [Compare Tuples](#)

concatenating, [Modify a Tuple](#)

creating with commas and (), [Create with Commas and \(\)](#)

creating with tuple(), [Create with tuple\(\)](#)

duplicating with *, [Duplicate Items with *](#)

generator comprehension and, [There Are No Tuple Comprehensions](#)

iteration, [Iterate with for and in](#)

lists versus, [Tuples Versus Lists](#)

named tuples, [Named Tuples](#)

practice exercise answers, [7. Tuples and Lists-7. Tuples and Lists](#)

practice exercises, [Things to Do](#)

unpacking, [Create with Commas and \(\)](#), [Create a List with a Comprehension](#)

Twirp, [Twirp](#)

twisted framework, [twisted](#)

2-D graphics, [2-D Graphics-ImageMagick](#), [Standard Library](#), [PIL and Pillow-PIL and Pillow](#), [ImageMagick](#)

type hints (type annotations), [Add Type Hints](#)

type() function, [Variables Are Names, Not Places](#)

types

basic, [Types](#)

booleans, [Numbers-Things to Do](#)

converting to integers, [Type Conversions-Type Conversions](#)

integers, [Integers-How Big Is an int?](#)

practice exercise answers, [2. Data: Types, Values, Variables, and Names](#)

practice exercises, [Things to Do](#)

variables, names, and objects, [Variables Are Names, Not Places-Variables Are Names, Not Places](#)

U

UDP (User Datagram Protocol), [TCP/IP](#)

underscore (_), [Literal Integers](#), [Uses of _ and _ in Names](#)

Unicode text strings, [Create with Quotes](#), [Text Strings: Unicode-For More Information](#)

decoding, [Decode](#)

encode() function, [Encode](#)

HTML entities, [HTML Entities](#)

in Python 3 strings, [Python 3 Unicode Strings-Python 3 Unicode Strings](#)

normalization, [Normalization](#)

practice exercise answers, [12. Wrangle and Mangle Data-12. Wrangle and Mangle Data](#)

practice exercises, [Things to Do](#)

UTF-8 encoding, [UTF-8](#)

unicodedata module, [Python 3 Unicode Strings](#)

union() function, [Combinations and Operators](#)

unittest, [Test with unittest-Test with unittest](#)

Unix

file operations, [File Operations-Delete a File with remove\(\)](#)

path separators, [Pathnames](#)

Python 3 installation, [Linux or Unix](#)

time representation, [Using the time Module](#)

update() function, [Combine Dictionaries with update\(\)](#)

UTC, [Using the time Module](#)

V

validation, data, [Data Validation](#)

values

arguments versus parameters, [Arguments and Parameters-Mutable and Immutable Arguments](#)

assigning to multiple variable names, [Assigning to Multiple Names](#)

assigning to sublist, [Change Items with a Slice](#)

assigning to variable, [Assignment](#)

computed, [Properties for Computed Values](#)

copying from one dictionary to another, [Copy with copy\(\)](#)

counting occurrences in list, [Count Occurrences of a Value with count\(\)](#)

default parameter, [Specify Default Parameter Values](#)

deleting from set, [Delete an Item with remove\(\)](#)

dictionaries and, [Dictionaries](#)

extracting from list, [Get an Item by \[offset \]](#)

false, [What Is True?](#), [None Is Useful](#)

getting/setting for attributes, [Attribute Access-Class and Object Attributes](#)

in set, [Test for a Value with in](#)

key-value stores (see NoSQL data stores)

literal, [Literal Values](#)

None, [None Is Useful](#)

returned by property, [Properties for Computed Values](#)

sentinel, [Redis](#)

test for existence in list, [Test for a Value with in](#)
true, [What Is True?](#)
variables, [Variables](#)
values() function, [Get All Values with values\(\)](#)
variables, [Variables](#)
(see also attributes)
assigning one list to more than one variable, [Assign with =](#)
assigning value to multiple names, [Assigning to Multiple Names](#)
assignment, [Assignment](#)
integers and, [Integers and Variables-Integers and Variables](#)
names versus places, [Variables Are Names, Not Places-Variables
Are Names, Not Places](#)
naming, [Choose Good Variable Names](#)
reassigning names, [Reassigning a Name](#)
type hints (type annotations), [Add Type Hints](#)
version control (see source control systems)
vertical bar (|), [Combinations and Operators](#)
virtual machines, [Virtual Machines](#)
virtualenv, [Use virtualenv](#), [Install virtualenv](#)
visual art (see graphics)
visualization, [Plots, Graphs, and Visualization-Bokeh](#)
Bokeh, [Bokeh](#)
Seaborn, [Seaborn](#)
VPython, [3-D Animation](#)

W

walrus operator, [New: I Am the Walrus](#)
web browser, [Web Clients](#)
web clients, [Web Clients-Beyond the Standard Library: requests](#)
Python's standard web libraries, [Python's Standard Web Libraries-
Python's Standard Web Libraries](#)
requests module, [Beyond the Standard Library: requests](#)
testing with curl, [Test with curl](#)
testing with httpbin, [Test with httpbin](#)
testing with httpie, [Test with httpie](#)
testing with telnet, [Test with telnet](#)

web crawling, [Crawl and Scrape-Requests-HTML](#)
web scraping, [Crawl and Scrape-Requests-HTML](#)
Web Server Gateway Interface (WSGI), [The Simplest Python Web Server](#)

web servers, [Web Servers-Other Frameworks](#)

Apache, [Apache](#)

ASGI, [ASGI](#)

automation and, [Web Services and Automation-webview](#)

bare-bones Python HTTP server, [The Simplest Python Web Server](#)

Bottle framework, [Bottle-Bottle](#)

database frameworks, [Database Frameworks-Database Frameworks](#)

Django framework, [Django](#)

event-based, [Other Python Web Servers](#)

Flask framework, [Flask-Pass an argument as part of the URL path frameworks, Web Server Frameworks-Other Frameworks](#)

NGINX, [NGINX](#)

WSGI, [The Simplest Python Web Server](#)

webbrowser module, [webbrowser](#)

webview module, [webview](#)

while loops, [Repeat with while-Check break Use with else](#)

canceling, [Cancel with break](#)

check break use with else, [Check break Use with else](#)

skipping ahead, [Skip Ahead with continue](#)

white space, in Python program structure, [Choose with if](#)

whitespace characters, stripping, [Strip with strip\(\)](#)

wildcard patterns (*), [Text Strings: Regular Expressions](#)

Windows

package managers, [Use a Package Manager](#)

path separators, [Pathnames](#)

Python 3 installation, [Windows](#)

shell, [Python Versus the Language from Planet X](#)

World Wide Web, [The Web, Untangled-Things to Do](#)

(see also web entries)

APIs and REST, [Web APIs and REST](#)

crawling and scraping, [Crawl and Scrape-Requests-HTML](#)

history of, [The Web, Untangled](#)

practice exercise answers, [18. The Web, Untangled](#)

practice exercises, [Things to Do](#)

video search program example, [Let's Watch a Movie-Let's Watch a Movie](#)

web clients, [Web Clients-Beyond the Standard Library: requests](#)

web servers, [Web Servers-Other Frameworks](#)

web services and automation, [Web Services and Automation-web-view](#)

web services/APIs, [Web Services and APIs](#)

write() function

print() versus, [Write a Text File with write\(\)](#)

writing binary file with, [Write a Binary File with write\(\)](#)

writing text file with, [Write a Text File with write\(\)](#)

writeheader() function, [CSV](#)

writer() function, [CSV](#)

WxPython, [Graphical User Interfaces](#)

X

XML (Extensible Markup Language) format, [XML-An XML Security Note](#)

xmlrpc, [XML RPC](#)

Y

YAML (YAML Aint Markup Language), [YAML-YAML](#)

Z

Zen of Python, [Your Moment of Zen](#)

ZeroMQ library, [ZeroMQ-ZeroMQ](#), [ZeroMQ](#)

Zerorpc, [Zerorpc](#)

zeros() function, [Make an Array with zeros\(\), ones\(\), or random\(\)](#)

zip files, [Distribute Your Programs](#)

zip() function, [Iterate Multiple Sequences with zip\(\)](#)

zsets (sorted sets), [Sorted sets](#)