Index

Symbols

! (exclamation point), Identifiers

!= (not equal to operator), Comparisons

!a format marker, Value Conversion

!r format marker, Value Conversion

!s format marker, Value Conversion

" (double quote), Delimiters, Strings

""" (triple double quotes), Strings

# (hash sign), Lines and Indentation, Delimiters

# (radix indicator) (f-strings), Radix indicator (#)

#! (shebang), Running Python Programs, Character Sets

$ (dollar sign), Identifiers, Delimiters

% (percent sign)

- legacy string format operator, Legacy String Formatting with %
- log message placeholder, The logging module
- modulus division operator, Division
- SQL statement placeholder, Parameter Style

%= character, Delimiters, Augmented assignment

%a format marker, Format Specifier Syntax

%r format marker, Format Specifier Syntax

%s character, Parameter Style

%s format marder, Format Specifier Syntax

& (bitwise AND operator), Expressions and Operators

&= character, Delimiters, Augmented assignment

' (single quote), Delimiters, Strings

''' (triple single quotes), Strings

'<' (align left), Fill and alignment

'=' (add fill characters), Fill and alignment

'>' (align right), Fill and alignment

'^' (align center), Fill and alignment

() (parentheses)

- function calls, Calling Functions
- in generator expressions, Generator expressions
- lines and indentation, Lines and Indentation

* (asterisk)

- argument collector parameters, Argument collector parameters
- concatenation and repetition, Concatenation and repetition
- extended unpacking, Plain assignment
- in fnmatch module, The fnmatch Module
- import statement, from...import *

** (double star) operator

- dict unpacking, Dictionaries
- exponentiation operator, Exponentiation
- glob recursive wildcard, The pathlib Module, The glob Module
- kwargs, Argument collector parameters, Positional and named arguments
- kwds, Argument collector parameters, “Keyword-only” parameters
- mapping pattern (structural pattern matching), Mapping patterns
- metaclass arguments, The class Statement, Dynamic class definition using the type built-in function
- named argument collector, Named argument collector-Argument collector parameters

**= character, Delimiters, Augmented assignment

*= character, Delimiters, Augmented assignment

*args, Argument collector parameters

*seq, Positional and named arguments

+ (plus sign), Concatenation and repetition

+= character, Delimiters, Augmented assignment

, (comma)

- in delete statements, del Statements
- delimiters, Delimiters
- in dictionaries, Dictionaries

- in global statements, The global statement
- in lists, Lists
- in tuples, Tuples

-= operator, Delimiters, Augmented assignment

. (period)
- floating-point numbers, Floating-point numbers
- import statement, The import Statement
- object attributes, Object attributes and items
- treatment by glob module, The glob Module

... (ellipsis), Interactive Sessions, Ellipsis (...)

.pyi stub files, Using .pyi Stub Files

/ (positional-only marker), Positional-only marker

/ (slash)
- in arithmetic operations, Division
- in file paths, Command-Line Syntax and Options
- in glob module, The glob Module
- pathlib.Path join operator, The pathlib Module

// (floor division operator), Division

//= character, Delimiters, Augmented assignment

/= character, Delimiters, Augmented assignment

0 (leading zero indicator) (f-strings), Leading zero indicator (0)

: (colon), Delimiters, Compound statements, Dictionaries

:= (assignment operator)
- assignment expressions, Assignment Expressions
- regular expressions and, REs and the := Operator
- simple statements, Simple statements

; (semicolon)
- in simple statements, Simple statements
- lines and indentation, Lines and Indentation

< (less than symbol)
- alignment indicator (f-strings), Fill and alignment
- comparisons, Comparisons

< (little-endian byte order), The struct Module

<<= character, Augmented assignment

<= (less than or equal to operator), Comparisons

= (equal sign)
- alignment indicator (f-strings), Fill and alignment

  - assignment statements, **Simple statements**
  - delimiters, **Delimiters**
  - in plain assignment, **Plain assignment**
  - struct format strings, **The struct Module**
- == (equal to operator), **Comparisons**, **Floating-Point Values**
- > (greater than symbol)
  - alignment indicator (f-strings), **Fill and alignment**
  - comparison operator, **Comparisons**
- >, ! (big endian/network standard byte order), **The struct Module**
- >= (greater than or equal to operator), **Comparisons**
- >>= operator, **Delimiters**, **Augmented assignment**
- ? (question mark), **Pattern String Syntax**, **The fnmatch Module**
- @ (commercial at sign)
  - decorators, **Decorators**-**Decorators**
  - matrix multiplication, **The operator Module**
  - struct format strings, **The struct Module**
- @= (matrix multiplication assignment operator), **Augmented assignment**
- \" (double quote escaped), **Strings**
- \' (single quote escaped), **Strings**
- \<newline> (ignore end of line), **Strings**
- \a (bell), **Strings**
- \b (backspace), **Strings**
- \DDD (octal value escape sequence in strings), **Strings**
- \f (form feed), **Strings**
- \n (newline), **Strings**
- \N{name} (Unicode character), **Strings**
- \r (carriage return), **Strings**
- \t (tab), **Strings**
- \U (eight hex digits), **Strings**
- \u (four hex digits), **Strings**
- \v (vertical tab), **Strings**
- \x XX (hexadecimal escape sequence), **Strings**
- \\ (backslash escaped), **Strings**
- ^ (alignment indicator) (f-strings), **Fill and alignment**
- ^= character, **Delimiters**, **Augmented assignment**
- _ (underscore)

with gettext module, Using gettext for localization

in identifiers, Identifiers, Class-private variables

in interactive interpreter, Identifiers

module-private variables, Module-private variables

numeric literals, Underscores in numeric literals

simple statements, Simple statements

wildcard pattern, The wildcard pattern

_getframe function (sys module), The Frame Type

__ (double underscore), Python Classes, Class-private variables, Special Methods

__abs__ (numeric special method), Special Methods for Numeric Objects

__add__ (numeric special method), Sequences, Special Methods for Numeric Objects

__and__ (numeric special method), Special Methods for Numeric Objects

__annotations__, Type Annotation Syntax, TypedDict, Using Type Annotations at Runtime

__bool__ (special method), General-Purpose Special Methods

__builtins__ dictionary, Python built-ins

__bytes__ (special method), General-Purpose Special Methods

__call__ (special method), General-Purpose Special Methods, General-Purpose Special Methods, ABCs in the collections module

__code__ attribute (function or method object), The Code Object Type

__complex__ (numeric special method), Special Methods for Numeric Objects

__contains__ (container method), Container methods, ABCs in the collections module

__debug__ variable, The assert Statement

__delattr__ (special method), General-Purpose Special Methods, General-Purpose Special Methods

__delete__ (descriptor method), Descriptors

__delitem__ (container method), Container methods

__del__ (special method), General-Purpose Special Methods, General-Purpose Special Methods __dir__ (special method), General-Purpose Special Methods

__divmod__ (numeric special method), Special Methods for Numeric Objects

__doc__ attribute, Class documentation strings, Module documentation strings

__eq__ (special method), General-Purpose Special Methods, General-Purpose Special Methods

__float__ (numeric special method), Special Methods for Numeric Objects

__floordiv__ (numeric special method), Special Methods for Numeric Objects

__format__ (special method), General-Purpose Special Methods, General-Purpose Special Methods

__future__ module, Applying the Code Changes

__getattribute__ (special method), __getattribute__, General-Purpose Special Methods, General-Purpose Special Methods

__getattr__ (special method), General-Purpose Special Methods, General-Purpose Special Methods, Attributes of module objects

__getitem__ (container method), Container methods

__get__ (descriptor method), Descriptors

__ge__ (special method), General-Purpose Special Methods, General-Purpose Special Methods

__gt__ (special method), General-Purpose Special Methods, General-Purpose Special Methods

__hash__ (special method), General-Purpose Special Methods, General-Purpose Special Methods, ABCs in the collections module

__ident, Class-private variables

__import__ (built-in function), Built-in Modules, Rebinding __import__

__index__ (numeric special method), Special Methods for Numeric Objects

__init__ (special method), __init__, General-Purpose Special Methods, General-Purpose Special Methods

__int__ (numeric special method), Special Methods for Numeric Objects

__invert__ (numeric special method), Special Methods for Numeric Objects
__iter__ (container method), Container methods, ABCs in the collections module
__len__ (container method), Container methods, ABCs in the collections module
__le__ (special method), General-Purpose Special Methods, General-Purpose Special Methods
__lshift__ (numeric special method), Special Methods for Numeric Objects
__lt__ (special method), General-Purpose Special Methods, General-Purpose Special Methods
__matmul__ (numeric special method), Special Methods for Numeric Objects
__missing__ (dict type), defaultdict
__mod__ (numeric special method), Special Methods for Numeric Objects
__mul__ (numeric special method), Sequences, Special Methods for Numeric Objects
__neg__ (numeric special method), Special Methods for Numeric Objects
__new__ (special method), General-Purpose Special Methods, General-Purpose Special Methods
__ne__ (special method), General-Purpose Special Methods, General-Purpose Special Methods
__or__ (numeric special method), Special Methods for Numeric Objects
__pos__ (numeric special method), Special Methods for Numeric Objects
__pow__ (numeric special method), Special Methods for Numeric Objects
__repr__ (special method), General-Purpose Special Methods
__rshift__ (numeric special method), Special Methods for Numeric Objects
__setattr__ (special method), General-Purpose Special Methods, General-Purpose Special Methods
__setitem__ (container method), Container methods

- __set__ (descriptor method), Descriptors
- __slots__ attribute, __slots__
- __str__ (special method), General-Purpose Special Methods, General-Purpose Special Methods
- __sub__ (numeric special method), Special Methods for Numeric Objects
- __truediv__ (numeric special method), Special Methods for Numeric Objects
- __xor__ (numeric special method), Special Methods for Numeric Objects
- \ (backslash)
  - lines and indentation, Lines and Indentation
  - re module, Pattern String Syntax
  - special meaning of in tokens, Delimiters
  - strings, Strings-Strings
- {} (braces)
  - command-line syntax and options, Command-Line Syntax and Options
  - delimiters, Delimiters
  - in dictionaries, Dictionaries
  - lines and indentation, Lines and Indentation
  - in sets, Sets
- | (vertical bar), Command-Line Syntax and Options, Dictionaries
- |= character, Delimiters, Augmented assignment
- [] (brackets)
  - command line syntax, Command-Line Syntax and Options
  - in-place list operations, In-place operations on a list
  - indexing a dictionary, Indexing a Dictionary
  - lines and indentation, Lines and Indentation
  - in lists, Lists
  - slicing a sequence, Slicing a sequence
- ~ (tilde), Expressions and Operators
- · (middle dot), Identifiers
- >>> (prompt), Interactive Sessions, Simple statements

A

- abc module, The abc module

abort (threading module), **Barrier objects**

abs (built-in function), **Built-in Functions**

absolute imports, **Absolute Versus Relative Imports**

abspath (os.path module), **The os.path Module**

abstract base classes (ABCs), **Abstract Base Classes**-**ABCs in the numbers module**, **Abstract Base Classes**

abstract syntax tree (AST), **Upgrade Automation Using pyupgrade**

access (os module), **File and directory functions of the os module**

accumulate (itertools module), **The itertools Module**

acquire (threading module), **Lock and RLock objects**, **Condition objects**, **Semaphore and BoundedSemaphore objects**

add (set method), **Set Methods**

add fill characters ('='), **Fill and alignment**

addCleanup (unittest.TestCase class), **The TestCase class**

add_history (readline module), **The readline Module**

algorithms, **Large-Scale Optimization**-**Summary of big-O times for operations on Python built-in types**

align center ('^'), **Fill and alignment**

align left ('<'), **Fill and alignment**

align right ('>'), **Fill and alignment**

alignment indicators, **Fill and alignment**

all (built-in function), **Built-in Functions**

Anaconda, **Anaconda and Miniconda**, **Other Environment Management Solutions**, **Additional numeric packages**

anchoring, **Anchoring at String Start and End**

and keyword, **Expressions and Operators**-**Expressions and Operators**

annotations, **Function Annotations**

any (built-in function), **Built-in Functions**

Any (typing module), **Types**

append method

  collections.deque, **deque**

  list, **List methods**

appendleft (collections.deque), **deque**

archived files, **Archived and Compressed Files**-**The ZipInfo class**

argparse module, **The argparse Module**

argument collector parameters, **Argument collector parameters** arguments, Defining Functions: The def Statement, Positional and named arguments-The semantics of argument passing

argv (sys module), The sys Module

arithmetic context, The decimal Module

arithmetic operations, Arithmetic Operations

Array class (multiprocessing module), The Array class

array module

- array type, The array Module
- methods and properties of array objects, The array Module
- type codes for, The array Module

array processing, Array Processing-Additional numeric packages

as (file opening), Creating a File Object with open

as (import statement), The from Statement

as (match statement), as patterns

as (with statement), The with Statement and Context Managers, Creating a File Object with open

as patterns, as patterns

ascii (built-in function), Built-in Functions

ASCII characters, Character Sets, Encoding Binary Data as ASCII Text-The uu Module

asctime (time module), The time Module

ASGI (Asynchronous Server Gateway Interface), ASGI

assert statement, The assert Statement, Testing with pytest

assertEqual (unittest.TestCase class), The TestCase class

assertRaises (unittest.TestCase class), The TestCase class

assignment expressions, Assignment Expressions

assignment operator (:=)

- assignment expressions, Assignment Expressions
- regular expressions and, REs and the := Operator
- simple statements, Simple statements

assignment statements, Simple statements, Assignment Statements-Augmented assignment, The copy Module

AST (abstract syntax tree), Upgrade Automation Using pyupgrade

ast module, Standard Input

ast.literal_eval function, Standard Input

asterisk (*)

  argument collector parameters, **Argument collector parameters**
  concatenation and repetition, **Concatenation and repetition**
  extended unpacking, **Plain assignment**
  in fnmatch module, **The fnmatch Module**
  import statement, **from...import ***
astimezone (datetime.datetime class), **The datetime Class**
async keyword, **Concurrency: Threads and Processes**
asynchronous (async) programming, **Concurrency: Threads and Processes**, **ASGI**
Asynchronous Server Gateway Interface (ASGI), **ASGI**
AsyncResult class, **The AsyncResult class**
as_integer_ratio (float method), **Floating-point numbers**
as_integer_ratio (int method), **Integer numbers**
at sign (@)
  decorators, **Decorators**-**Decorators**
  delimiters, **Delimiters**
  matrix multiplication, **The operator Module**
  matrix multiplication assignment operator, **Augmented assignment**
  struct format strings, **The struct Module**
atexit module, **Termination Functions**
atomic actions, **Threads in Python**
attrgetter (operator module), **The operator Module**
attributes
  attribute reference basics, **Attribute Reference Basics**-**Setting an attribute**
  attribute reference special methods, **General-Purpose Special Methods**
  attribute references, **Plain assignment**
  denoting, **Object attributes and items**
  of instance objects, **Attributes of instance objects**
  mutability, **Data Types**
  of file objects, **Attributes and Methods of File Objects**-**Attributes and Methods of File Objects**
  special attributes of package objects, **Special Attributes of Package Objects**

specifying, **Attributes of class objects**

audit (sys module), **The sys Module**

augmented assignment, **Augmented assignment**

augmented operators, **Augmented assignment**

automatic code conversion, **Upgrade Automation Using pyupgrade**

available_timezones (zoneinfo module), **The zoneinfo Module**

await keyword, **Concurrency: Threads and Processes**, **Python 3.9**

aware, **The datetime Module** (see also datetime module)

B

b64decode (base64 module), **The base64 Module**

b64encode (base64 module), **The base64 Module**

backslash (\)

lines and indentation, **Lines and Indentation**

special meaning of in tokens, **Delimiters**

strings, **Strings**-**Strings**

backslash escaped (\\), **Strings**

backspace (\b), **Strings**

backward compatibility, **Python Development and Versions**

bare excepts, **try/except**

Barrier (threading module), **Barrier objects**

Barrier objects, **Barrier objects**

base-classes, **The class Statement**

base64 module, **The base64 Module**-**The base64 Module**

basename (os.path module), **The os.path Module**

BDD (behavior-driven development), **Unit Testing and System Testing**

BDFL (Benevolent Dictator for Life), **Python Development and Versions**

Beaker, **Considerations When Using Lightweight Frameworks**

BeautifulSoup package

BeautifulSoup class, **The BeautifulSoup Class**

benefits of, **The BeautifulSoup Third-Party Package**

building HTML with, **Building HTML with bs4**

CSS selectors, **bs4 CSS Selectors**

editing and creating HTML with, Editing and Creating HTML with bs4-Replacing and removing nodes
installing versus importing, The BeautifulSoup Third-Party Package
navigable classes of, The Navigable Classes of bs4-next_element, previous_element, next_elements, and previous_elements
parsing example, An HTML Parsing Example with BeautifulSoup
search methods, bs4 find... Methods (aka Search Methods)-Other named arguments
behavior-driven development (BDD), Unit Testing and System Testing
bell (\a), Strings
benchmarking, Benchmarking
Benevolent Dictator for Life (BDFL), Python Development and Versions
Berkeley DB library, DBM Modules
Berkeley socket interface
client/server computing, Client/Server Computing-Connection-oriented client and server structures
networking using sockets, The Berkeley Socket Interface
socket addresses, Socket Addresses
socket clients
connection-oriented, A Connection-Oriented Socket Client
connectionless, A Connectionless Socket Client
socket module, The socket Module-The socket Module
socket objects, Socket Objects-Socket Objects
socket servers
connection-oriented, A Connection-Oriented Socket Server
connectionless, A Connectionless Socket Server
big endian/network standard byte order (>, !), The struct Module
big-O notation, Large-Scale Optimization-Summary of big-O times for operations on Python built-in types

operations on Python built-in types, **Summary of big-O times for operations on Python built-in types**

bin function (built-in), **Built-in Functions**

binary data, encoding as ASCII text, **Encoding Binary Data as ASCII Text**-**The uu Module**

binary literals, **Integer numbers**

binary mode, **Binary and text modes**

binding

attributes, **General-Purpose Special Methods**

port numbers, **The Berkeley Socket Interface**

unbinding with del, **del Statements**

variables, **Variables**-**Plain assignment**

bitwise AND operator (&), **Expressions and Operators**

bitwise operations on integers, **Bitwise Operations on Integers**

bit_count (int method), **Integer numbers**

bit_length (int method), **Integer numbers**

black tool, **Typographic Conventions**, **Tools for Checking Python Programs**, **Strings**

black-box testing, **Unit Testing and System Testing**

blank lines, **Lines and Indentation**

block structure, **Lines and Indentation**

blocks, **Compound statements**, **The if Statement**

blue tool, **Tools for Checking Python Programs**, **Strings**

bodies, **Compound statements**

bool (built-in type), **Boolean Values**, **Built-in Types**

Boolean expressions, **Short-circuiting of Boolean expressions**

Boolean values, **Boolean Values**, **General-Purpose Special Methods**

bottlenecks, **Optimization**

bound methods, **Bound and Unbound Methods**-**Bound and Unbound Methods**, **Class methods**, **General-Purpose Special Methods**

bounded iterables, **Iterables**

BoundedSemaphore (threading module), **Semaphore and BoundedSemaphore objects**

BoundedSemaphore objects, **Semaphore and BoundedSemaphore objects** braces ({})

command-line syntax and options, **Command-Line Syntax and Options**

delimiters, **Delimiters**

in dictionaries, **Dictionaries**

lines and indentation, **Lines and Indentation**

in sets, **Sets**

brackets ([])

command line syntax, **Command-Line Syntax and Options**

in-place list operations, **In-place operations on a list**

indexing a dictionary, **Indexing a Dictionary**

lines and indentation, **Lines and Indentation**

in lists, **Lists**

slicing a sequence, **Slicing a sequence**

break (pdb module), **The pdb Module**

break keyword

break statement, **The break Statement**

in else clauses, **The else Clause on Loop Statements**

in for statements, **The for Statement**

in generators, **Generators**

in try/except, **try/except**

in while statements, **The while Statement**

breakpoint (built-in function), **Built-in Functions**

broken (threading module), **Barrier objects**

browsers, running Python in, **Running Python in the Browser**-**Jupyter**

bs4 module (see BeautifulSoup package)

bsddb3 package, **DBM Modules**

buffering, **File and Text Operations**, **Buffering**

bugs, fixing, **Unit Testing and System Testing**

built-ins

core built-ins, **Built-in Types**-**Built-in Types**

definition of built-in, **Core Built-ins and Standard Library Modules**

functions, **Built-in Functions**-**Built-in Functions**

hiding, **Built-in Functions**

inheritance from types, **Inheritance from Built-in Types** bytearray (built-in type), **Built-in Types**

bytearray objects, **bytearray objects**

bytes (built-in type), **Built-in Types**

bytes objects, **bytes objects**

bytes type, **Strings and Things**

BytesIO class (io module), **In-Memory Files: io.StringIO and io.BytesIO**

bytestring/byte string, **Strings and Things**

bzip2 file compression, **File and Text Operations**

C

C API, **Extending and Embedding Classic Python**

cache (functools module), **The functools Module**

cached_property (functools module), **The functools Module**

caching, **Memoizing**

calcsize (struct module), **The struct Module**

calendar module, **The calendar Module**

callable (built-in function), **Built-in Functions**

Callable (typing module), **Types**

callable attributes, **Object attributes and items**

callable instances, **General-Purpose Special Methods**

callables, **Callables**

cancel (threading module), **Timer objects**

capitalize method (string method), **Methods of String Objects**

capture patterns, **Capture patterns**

carriage return (\r), **Strings**

case

  in class names, **Identifiers**

  in identifier names, **Identifiers**

case clauses, **The match Statement**, **Guards**

case soft keyword, **Keywords**, **The match Statement**

casefold (str method), **Methods of String Objects**

cast (typing module), **Utilities and Decorators**

center (string method), **Methods of String Objects**

CFFI, **The Python Standard Library and Extension Modules**, **Online Material**

CGI (Common Gateway Interface), **Serving HTTP** chain (itertools module), **The itertools Module**

chain.from_iterable (itertools module), **The itertools Module**

ChainMap class (collections module), **ChainMap**

character sets, **Character Sets**, **Sets of Characters**

chdir (contextlib module), **The with Statement and Context Managers**

chdir (os module), **File and directory functions of the os module**

checkpointing, **Handling Errors in Large Programs**

chmod (os module), **File and directory functions of the os module**

chmod (pathlib.Path class), **The pathlib Module**

choice (random module), **The random Module**

choice (secrets module), **Crypto-Quality Random Numbers: The secrets Module**

chr (built-in function), **Built-in Functions**

chroot-jail process, **The Berkeley Socket Interface**

circular imports, **Circular Imports**

class names, **Identifiers**

class patterns, **Class patterns** (see also match)

class statements, **The class Statement**

class-private variables, **Class-private variables**

classes, **Configuring classes for positional matching**, **Classes and Instances**-**Inheritance from Built-in Types**

- attribute reference basics, **Attribute Reference Basics**
- bound and unbound methods, **Bound and Unbound Methods**
- class body, **The Class Body**
- class statements, **The class Statement**
- class-level methods, **Class-Level Methods**
- configuring for positional matching, **Configuring classes for positional matching**
- descriptors, **Descriptors**
- inheritance, **Inheritance**
- inheritance from built-in types, **Inheritance from Built-in Types**
- instances, **Instances**
- per instance methods, **Per Instance Methods**
- properties, **Properties**
- __getattribute__ (special method), **__getattribute__**

  __slots__ attribute, **__slots__**
CLDR (Common Locale Data Repository), **More Internationalization Resources**
cleanup handlers, **try/finally**
clear (threading module), **Event objects**
clear method
  collections.deque, **deque**
  dict type, **Dictionary Methods**
  list type, **List methods**
  set type, **Set Methods**
client codes, **File-Like Objects and Polymorphism**
client-side network modules (see network protocol modules)
client/server computing, **Client/Server Computing**-**Connection-oriented client and server structures**
close (file method), **Attributes and Methods of File Objects**
close (fileinput module), **The fileinput Module**
closing (contextlib module), **The with Statement and Context Managers**
closures, **Nested functions and nested scopes**
cmath module, **The math and cmath Modules**-**The math and cmath Modules**
code conversion, **Upgrade Automation Using pyupgrade**
code examples, obtaining and using, **Using Code Examples**
code hosting, **Optimizing loops**
code objects, **compile and Code Objects**, **The Code Object Type**
code, untrusted, **Never exec or eval Untrusted Code**
codecs module, **The codecs Module**
coding directive comments, **Character Sets**
collect (gc module), **The gc Module**
collections module, **ABCs in the collections module**, **The collections Module**-**deque**
colon (:), **Delimiters**, **Compound statements**, **Dictionaries**
colorama package, **Console I/O**
colored text, **Console I/O**
comb (math module), **The math and cmath Modules**
combinatorial generators, **The itertools Module**
combine (datetime.datetime class), **The datetime Class**

comma (,)

in delete statements, del Statements

in dictionaries, Dictionaries

in global statements, The global statement

in lists, Lists

in tuples, Tuples

command-line options, Command-Line Syntax and Options-Command-Line Syntax and Options

parsing (argparse module), The argparse Module

commands (executing system), Running Other Programs with the os Module

comments and questions, How to Contact Us

Common Gateway Interface (CGI), Serving HTTP

Common Locale Data Repository (CLDR), More Internationalization Resources

commonpath (os.path module), The os.path Module

commonprefix (os.path module), The os.path Module

comparison

comparison chaining, Comparison Chaining

comparison operators, Comparisons

lexicographic comparison, Sequences

special methods for, General-Purpose Special Methods

compile (built-in function), Built-in Functions, compile and Code Objects

compile (re module), Functions of the re Module

complex (built-in type), Built-in Types

complex numbers, Complex numbers

compound statements, Compound statements

compress (itertools module), The itertools Module

compressed files, Archived and Compressed Files-The ZipInfo class

concatenation, Concatenation and repetition

concrete classes, Abstract Base Classes

concurrency (see threads and processes)

concurrent.futures module, The concurrent.futures Module-The concurrent.futures Module conda package management system, Other Environment Management Solutions

Condition (threading module), Condition objects

Condition objects, Condition objects

conditional operator, The conditional operator

conferences, Python conferences

conjugate, Complex numbers

connect function (DBAPI-compliant modules), The connect Function

connection objects (DBAPI-compliant modules), Connection Objects

connection-oriented protocols, Networking Basics, Client/Server Computing

connection-oriented socket clients, A Connection-Oriented Socket Client

connection-oriented socket servers, A Connection-Oriented Socket Server

connectionless protocols, Networking Basics, Client/Server Computing

connectionless socket clients, A Connectionless Socket Client

connectionless socket servers, A Connectionless Socket Server

console I/O, Console I/O

constant time performance (O(1)), Large-Scale Optimization

containers

- indexing and, Plain assignment
- object attributes and, Object attributes and items
- special methods for, Special Methods for Containers-Container methods

context managers, try/finally, The with Statement and Context Managers-The with Statement and Context Managers

contextlib module, The with Statement and Context Managers

contextmanager (contextlib module), The with Statement and Context Managers

continuation lines, Lines and Indentation

continue statements, The continue Statement

control characters, Delimiters

control flow statements, Control Flow Statements-The with Statement

break, The break Statement

continue, The continue Statement

else clause, The if Statement, The while Statement, The else Clause on Loop Statements, try/except

for statement, The for Statement

if, The if Statement

match, The match Statement

pass, The pass Statement

try and raise, The try and raise Statements

while, The while Statement

with, The with Statement

controlled deployment, Use a Controlled Deployment Process

copy function

copy module, The copy Module

shutil module, The shutil Module

copy method

dict type, Dictionary Methods

set type, Set Methods

copy module, The copy Module

copyfile function (shutil module), The shutil Module

copyfileobj function (shutil module), The shutil Module

copymode function (shutil module), The shutil Module

copyreg module, Pickling customization with the copyreg module

copysign (math module), The math and cmath Modules

copystat function (shutil module), The shutil Module

copytree function (shutil module), The shutil Module

coroutines, Generators as near-coroutines

cos (math module), The math and cmath Modules

count (itertools module), The itertools Module

count (string method), Methods of String Objects

count method

list type, List methods

sequence type, Sequences

str and bytes types, Methods of String Objects tuple type, Tuples

Counter class (collections module), Counter

counting semaphores, Semaphore and BoundedSemaphore objects

cProfile module, Profiling

CPU-bound programs, Concurrency: Threads and Processes, Using multiprocessing for heavy CPU work

CPython (Classic Python), CPython-Choosing Between CPython, PyPy, and Other Implementations, Extending and Embedding Classic Python

cryptography applications, Crypto-Quality Random Numbers: The secrets Module

CSS selectors, bs4 CSS Selectors

csv module, The csv Module-A csv example

ctime (datetime.datetime class), The datetime Class

ctypes module, The Python Standard Library and Extension Modules, The Value class, The Array class, Online Material

cultural conventions, Internationalization

curdir (os module), Path-string attributes of the os module

currency (locale module), The locale Module

currency values, The locale Module-The locale Module, The decimal Module

currentframe (inspect module), The inspect Module

curses module, curses

cursor objects (DBAPI-compliant), Cursor Objects

custom exception classes, Custom Exception Classes-Other Exceptions Used in the Standard Library

custom importers, Custom Importers

custom types

base classes for defining, Defining Custom Types

generics and TypeVars, Generics and TypeVars

NamedTuple, NamedTuple

NamedTuple versus TypedDict, TypedDict

NewType, NewType

restricting TypeVar to specific types, Restricting TypeVar to specific types

TypeAlias, TypeAlias

TypedDict, TypedDict-TypedDict

customizing execution (see execution, customizing)

cwd (pathlib.Path class), The pathlib Module

cycle (itertools module), The itertools Module

cyclic garbage loops, The gc Module

Cython, The Python Standard Library and Extension Modules, Modules and Packages, Online Material

D

data descriptors, Overriding and nonoverriding descriptors

data interchange, Serialization

data types, Data Types-Boolean Values (see also lists; strings)

Boolean values, Boolean Values

callables, Callables

dictionaries, Dictionaries

Ellipsis (...), Ellipsis (...)

frozensets, Sets

None, None

numbers, Numbers

sequences, Sequences

sets, Sets

tuples, Tuples, Tuples, The time Module

Database API 2.0 standard (DBAPI)

connect function, The connect Function

connection objects, Connection Objects

cursor objects, Cursor Objects

DBAPI-compliant modules, DBAPI-Compliant Modules-A sqlite3 example

exception classes, Exception Classes

factory functions, Factory Functions

importing DBAPI-compliant modules, The Python Database API (DBAPI)

parameter style, Parameter Style

SQLite, SQLite

thread safety, Thread Safety

type description attributes, Type Description Attributes

DataBase Manager (DBM) modules, DBM Modules-Examples of DBM-Like File Use

databases (see persistence and databases)

dataclasses module, Data Classes

datagram protocols, Networking Basics

date (datetime.datetime class), The datetime Class

date class (datetime module), The date Class-The date Class

datetime class (datetime module), The datetime Class-The datetime Class

datetime module, The datetime Module-The timezone Class

- date class, The date Class
- datetime class, The datetime Class
- time class, The time Class
- timedelta class, The timedelta Class
- timezone class, The timezone Class
- tzinfo abstract class, The tzinfo Abstract Class

dateutil module, The dateutil Module

DBAPI (see Database API)

DBM (DataBase Manager) modules, DBM Modules-Examples of DBM-Like File Use

deadlocks, Concurrency: Threads and Processes

debugging

- definition of term, Testing, Debugging, and Optimizing
- initial steps, Before You Debug
- inspect module, The inspect Module-An example of using inspect
- ipdb module, Other Debugging Modules
- pdb module, The pdb Module-The pdb Module
- printing versus logging information, Debugging
- pudb module, pudb
- timing of, Debugging
- traceback module, The traceback Module
- using f-strings, Debug printing with f-strings

decimal literals, Integer numbers

decimal module, The decimal Module-The decimal Module

decode method (bytes type), bytes objects decorate–sort–undecorate (DSU) idiom, **The Decorate–Sort–Undecorate Idiom**

decorators, **Decorators**-**Decorators**, **Utilities and Decorators**-**Utilities and Decorators**

dedent (textwrap module), **Text Wrapping and Filling**

deepcopy (copy module), **The copy Module**

def statement, **Defining Functions: The def Statement**, **Function definitions in a class body**

default parameters, **Named parameters**

default values, **Named parameters**, **Mutable default parameter values**

defaultdict class (collections module), **defaultdict**

deflate file compression, **File and Text Operations**

del statement, **del Statements**

delattr (built-in function), **Built-in Functions**

delimiters, **Delimiters**

denominators, **The fractions Module**

deployment, **Use a Controlled Deployment Process**

DeprecationWarning class, **Classes**

deque (double-ended queue) (collections module)

  deque class (collections module), **deque**

descendants, **The class Statement**

descriptors, **Python Classes**, **Descriptors**

deserialization, **Serialization**

development environments, **Python Development Environments**-**Tools for Checking Python Programs**

  free text editors, **Free Text Editors with Python Support**

  Integrated Development and Learning Environment (IDLE), **IDLE**

  other Python IDEs, **Other Python IDEs**

  tools for checking programs, **Tools for Checking Python Programs**

dict (built-in type), **Built-in Types**, **defaultdict**

dictionaries, **Dictionaries**-**Dictionaries**

dictionary comprehensions, **Dictionary comprehensions**

dictionary operations, **Dictionary Operations**-**Dictionary Methods**, **Dictionary operations**

- dictionary membership, Dictionary Membership
- dictionary methods, Dictionary Methods
- indexing dictionaries, Indexing a Dictionary
- optimizing, Dictionary operations

difference (set method), Set Methods

digit grouping (f-strings), Grouping option

dir (built-in function), Built-in Functions

direct subclasses, The class Statement

dirname (os.path module), The os.path Module

disable (gc module), The gc Module

discard (set method), Set Methods

displayhook (sys module), The sys Module

dist (math module), The math and cmath Modules

distributed computing, Concurrency: Threads and Processes-Concurrency: Threads and Processes

distribution utilities (distutils), Distribution Utilities (distutils) and setuptools-Distribution Utilities (distutils) and setuptools

Diversity and Inclusion workgroup, Workgroups

division, Division

divmod (built-in function), Division, Built-in Functions

Django, A Few Popular Full-Stack Frameworks

docstrings (documentation strings), Docstrings, Class documentation strings

doctest module, The doctest Module-The doctest Module

documentation, Documentation-Books

documentation strings (docstrings), Docstrings, Class documentation strings

dollar sign ($), Identifiers, Delimiters

double quote ("), Delimiters, Strings

double quote escaped (\"), Strings

double star (**) operator

- dict unpacking, Dictionaries
- exponentiation operator, Exponentiation
- glob recursive wildcard, The pathlib Module, The glob Module
- kwargs, Argument collector parameters, Positional and named arguments

kwds, Argument collector parameters, “Keyword-only” parameters

mapping pattern (structural pattern matching), Mapping patterns

metaclass arguments, The class Statement, Dynamic class definition using the type built-in function

named argument collector, Named argument collector-Argument collector parameters

double-ended queue (deque), deque

dropwhile (itertools module), The itertools Module

dst (datetime.time class), The time Class

dst (tzinfo class), The tzinfo Abstract Class

DSU (decorate–sort–undecorate) idiom, The Decorate–Sort–Undecorate Idiom

dtype, NumPy

duck typing, Built-in Types

dunder (double underscore)

attribute references, Attribute Reference Basics

availability of, Class-private variables

definition of term, Python Classes, Special Methods

general-purpose special methods, General-Purpose Special Methods

import hooks, Import hooks

inheritance and, Python Classes

module metadata, Attributes of module objects

object type, The Built-in object Type

operator module, The operator Module

overriding and nonoverriding descriptors, Overriding and nonoverriding descriptors

dynamic arrays, List operations

dynamic execution, Dynamic Execution and exec-Never exec or eval Untrusted Code

dynamically typed languages, History

E

EAFP (It’s easier to ask forgiveness than permission), LBYL Versus EAFP, The queue Module elements (collections.Counter class), Counter

ElementTree

- building an ElementTree from scratch, Building an ElementTree from Scratch
- Element class, The Element Class-The Element Class
- ElementTree class, The ElementTree Class
- functions in ElementTree module, Functions in the ElementTree Module-Functions in the ElementTree Module
- packages implementing, ElementTree
- parsing XML with ElementTree.parse, Parsing XML with ElementTree.parse-Editing an ElementTree

elif clauses, The if Statement

ellipsis (...), Interactive Sessions, Ellipsis (...)

else clause

- if/else, The if Statement
- try-except-else, try/except
- on while and for loops, The else Clause on Loop Statements

Emacs text editor, Free Text Editors with Python Support

email module

- creating messages, Creating Messages
- email.encoders module, The email.encoders Module
- email.Generator module, The email.Generator Module
- email.message module, The email.message Module-The email.message Module
- email.utils module, The email.utils Module
- example uses of, Example Uses of the email Package
- functions in, Functions in the email Package
- versus imaplib, poplib, and smtplib modules, MIME and Email Format Handling

email protocols, Email Protocols-The smtplib Module

embedding, Extending and Embedding Classic Python

empty lines, Lines and Indentation

enable (gc module), The gc Module

encode (str method), Methods of String Objects

encoding binary data, Encoding Binary Data as ASCII Text-The uu Module

encoding declarations, Character Sets

end-to-end testing, Unit Testing and System Testing

endswith (str and bytes method), Methods of String Objects

English language, Identifiers

enum module, Enumerated Types (Enums)

enumerate (built-in function), Built-in Functions

enumerated types (enums), Enumerated Types (Enums)-Enumerated Types (Enums)

environment variables

- documentation, Environment Variables
- PYTHONDONTWRITEBYTECODE, Command-Line Syntax and Options
- PYTHONHOME, Environment Variables
- PYTHONINSPECT, Command-Line Syntax and Options
- PYTHONOPTIMIZE, Command-Line Syntax and Options
- PYTHONPATH, Environment Variables
- PYTHONSTARTUP, Environment Variables
- PYTHONUNBUFFERED, Command-Line Syntax and Options
- PYTHONVERBOSE, Command-Line Syntax and Options

epochs, Time Operations

- (see also time operations)

equal sign (=)

- alignment indicator (f-strings), Fill and alignment
- assignment statements, Simple statements
- delimiters, Delimiters
- in plain assignment, Plain assignment
- struct format strings, The struct Module

equal to operator (==), Comparisons, Floating-Point Values

equality checking, Built-in Types

errno module, The errno Module

error-checking strategies

- handling errors in large programs, Handling Errors in Large Programs
- logging errors, Logging Errors-Configuring logging
- look before you leap (LBYL), LBYL Versus EAFP

errors

- diagnosing, Accessing nonexistent references
- ValueError, Sequences in General

escape sequences, Strings, Alternatives

eval (built-in function), Built-in Functions, Expressions-Never exec or eval Untrusted Code

Event (threading module), Event objects

Event objects, Event objects

eviction policies, Memoizing

except clause, try/except-try/except/finally, The Hierarchy of Standard Exceptions, Timer objects

excepthook (sys module), The sys Module

exception (sys module), The sys Module

exception classes, Exception Classes

exception handlers, try/except

exception handling, The try and raise Statements, Exceptions

exceptions

- assert statement, The assert Statement
- custom classes, Custom Exception Classes-Other Exceptions Used in the Standard Library
- error-checking strategies, Error-Checking Strategies-Configuring logging
- exception objects, Exception Objects-Exceptions “wrapping” other exceptions or tracebacks
- exception propagation, Exception Propagation
- ExceptionGroup and except*, ExceptionGroup and except*-ExceptionGroup and except*
- generators and, Generators and Exceptions
- purpose of, Exceptions
- raise statement, The raise Statement
- standard exceptions, Standard Exception Classes
- try statement, The try Statement-try/except/finally
- with statement and context managers, The with Statement and Context Managers-The with Statement and Context Managers

exclamation point (!), Identifiers

exc_info (sys module), The sys Module

exec (built-in function), Built-in Functions, Dynamic Execution and exec-Never exec or eval Untrusted Code, Running Other Programs with the os Module, Avoid exec and from ... import * executable files, Running Other Programs with the os Module

executing system commands, Running Other Programs with the os Module

execution, customizing

- dynamic execution and exec function, Dynamic Execution and exec-Never exec or eval Untrusted Code
- garbage collection, Garbage Collection-The weakref Module
- internal types, Internal Types
- per-site customization, Per-Site Customization
- termination functions, Termination Functions

Executor class (concurrent package), The concurrent.futures Module

execvpe (os module), Running Other Programs with the os Module

exists (os.path module), The os.path Module

exists (pathlib.Path class), The pathlib Module

exit (sys module), The sys Module

exp (math module), The math and cmath Modules

expandtabs method (string method), Methods of String Objects

expanduser (os.path module), The os.path Module

expanduser (pathlib.Path class), The pathlib Module

expression is true/false, Boolean Values

expression statements, Expressions

expressions and operators

- assignment expressions, Assignment Expressions
- Boolean context of, The if Statement
- comparison chaining, Comparison Chaining
- evaluation of versus returns from, The if Statement
- expressions as simple statements, Simple statements
- obtaining expression values, Expressions
- short-circuiting operators, Short-Circuiting Operators

extend method

- array, The array Module, Building up a string from pieces
- collections.deque, deque
- list, List methods
- MutableSequence, ABCs in the collections module
- sequence type, Sequences

extendleft (collections.deque), **deque**

Extensible HyperText Markup Language (XHTML), **Structured Text: HTML**

eXtensible Markup Language (XML), **Structured Text: HTML**, **Structured Text: XML**

Extensible Markup Language remote procedure call (XML-RPC) protocol, **Other Network Protocols**

extension modules

- benefits of, **Modules and Packages**
- versus built-in modules, **Core Built-ins and Standard Library Modules**
- in collections.abc, **ABCs in the collections module**
- documentation, **Extension modules and Python sources**
- introduction to, **The Python Standard Library and Extension Modules**
- packaging programs and extensions, **Packaging Programs and Extensions**-**Online Material**

F

f-strings (formatted string literals), **Strings**, **Formatted String Literals (F-Strings)**

factory functions (DBAPI-compliant), **Factory Functions**

falsy

- definition of term, **Boolean Values**
- evaluates as versus returns, **The if Statement**
- evaluation in Boolean contexts, **General-Purpose Special Methods**, **Built-in Types**

FastAPI, **FastAPI**-**FastAPI**

fdopen (os module), **File descriptor operations**

field width (f-strings), **Field width**

file and text operations (see also structured text)

- archived and compressed files, **Archived and Compressed Files**-**The ZipInfo class**
- chapters discussing files, **File and Text Operations**
- definition of file, **File and Text Operations**
- errno module, **The errno Module**
- filecmp module, **The filecmp Module**-**The filecmp Module**

fileinput module, The fileinput Module-The fileinput Module

fnmatch module, The fnmatch Module

glob module, The glob Module

in-memory files, In-Memory Files: io.StringIO and io.BytesIO

internationalization, Formatted String Literals (F-Strings), Internationalization-More Internationalization Resources

io module, The io Module-File-Like Objects and Polymorphism

os module, The os Module-The shutil Module

overview of, File and Text Operations

pathlib module, The pathlib Module-The pathlib Module

richer-text I/O, Richer-Text I/O-The msvcrt module

shutil module, The shutil Module-The shutil Module

stat module, The stat Module

struct module, The struct Module-The struct Module

tempfile module, The tempfile Module-The tempfile Module

text input and output, Text Input and Output-The getpass Module

file descriptor (fd), Creating a File Object with open, File and directory functions of the os module, File descriptor operations-File descriptor operations

file-like objects, File-Like Objects and Polymorphism, In-Memory Files: io.StringIO and io.BytesIO

filecmp module, The filecmp Module-The filecmp Module

FileInput class (fileinput module), The fileinput Module

fileinput module, The fileinput Module-The fileinput Module

filelineno (fileinput module), The fileinput Module

filename (fileinput module), The fileinput Module

filename matching, The fnmatch Module

fileno method

socket module, Socket Objects

fileno method)

file object, Attributes and Methods of File Objects

filesystem

definition of term, File and Text Operations

searching for modules, Searching the Filesystem for a Module-Searching the Filesystem for a Module filesystem operations (os module)

- file and directory functions, File and directory functions of the os module-File and directory functions of the os module
- file descriptor operations, File descriptor operations-File descriptor operations
- path-string attributes, Path-string attributes of the os module
- permissions, Permissions

fill (textwrap module), Text Wrapping and Filling

fill character (f-strings), Fill and alignment

filter (built-in function), Built-in Functions

filter (fnmatch module), The fnmatch Module

filterfalse (itertools module), The itertools Module

final (typing module), Utilities and Decorators

finalization, General-Purpose Special Methods

finally clauses, try/finally

find method

- bs4.BeautifulSoup, bs4 find… Methods (aka Search Methods)
- ElementTree.Element, The Element Class
- mmap module, Methods of mmap Objects
- string and bytes types, Methods of String Objects

first-class objects, Plain assignment, Functions

flake 8 tool, Tools for Checking Python Programs

Flask, Flask-Flask response objects

flit tool, Packaging Programs and Extensions

float (built-in type), Built-in Types

float methods, Floating-point numbers

floating-point numbers

- basics of, Floating-point numbers
- caching and, Memoizing
- comparing for equality, Comparisons
- decimal numbers, The decimal Module
- formatting, Format type
- numeric processing of, Floating-Point Values-Floating-Point Values

floor division operator (//), Division

fnmatch module, The fnmatch Module

for statement, The for Statement-Dictionary comprehensions

- basics of, The for Statement
- dictionary comprehensions, Dictionary comprehensions
- iterables versus iterators, Iterables versus iterators
- iterators, Iterators
- list comprehensions (listcomps), List comprehensions
- mutable objects and, The for Statement
- range function, range
- set comprehensions, Set comprehensions
- syntax of, The for Statement

form feed (\f), Strings

formal parameters, Parameters

format (built-in function), Built-in Functions

format (str method), Methods of String Objects, Formatted String Literals (F-Strings)

format calls, Formatting Using format Calls

format specifier

- f-strings, Value Formatting: The Format Specifier-Format type
- legacy str formatting, Format Specifier Syntax

format types, Format type

formatargspec (inspect module), The inspect Module

formatargvalues (inspect module), The inspect Module

formatted string literals (f-strings), Strings, Formatted String Literals (F-Strings)

format_map method (str method), Methods of String Objects

forward-referencing types, Type Annotation Syntax

fractions module, The fractions Module

frame objects, The Frame Type

free variables, Nested functions and nested scopes

freeze (gc module), The gc Module

from ... import, The from Statement-Handling import failures, Avoid exec and from ... import *

fromordinal (datetime.datetime class), The datetime Class

fromtimestamp (datetime.datetime class), The datetime Class

from_bytes (int method), Integer numbers

from_hex (float method), Floating-point numbers

frozenset (built-in type), Built-in Types

frozensets, Sets

fsum (math module), The math and cmath Modules
ftplib module, HTTP and URL Clients
full-stack frameworks, "Full-Stack" Versus "Lightweight" Frameworks
function calls, Functions
function signatures
- *args and **kwds in, Argument collector parameters
- adding type annotations to new code, Adding Type Annotations to New Code
- argument collector parameters, Argument collector parameters
- Array class (multiprocessing module), The Array class
- callable object, Types
- forward-referencing types, Type Annotation Syntax
- generics and TypeVars, Generics and TypeVars
- keyword-only parameters, "Keyword-only" parameters
- matching arguments to parameters, Matching arguments to parameters
- overload decorator, Utilities and Decorators
- Pool class (multiprocessing module), The Pool class
- Popen class (subprocess module), Using the Subprocess Module
- positional argument collector (*), Positional argument collector
- positional-only marker (/), Positional-only marker
- Process class (multiprocessing module), The Process class
- Value class (multiprocessing module), The Value class

functional testing, Unit Testing and System Testing
functions
- attributes of function objects, Attributes of Function Objects
- built-in functions, Built-in Functions-Built-in Functions
- calling functions, Calling Functions
- defining with def statement, Defining Functions: The def Statement
- definition of term, Functions
- function annotations, Function Annotations
- function definitions in class bodies, Function definitions in a class body

- generators, Generators
- higher-order functions, Decorators, The operator Module
- namespaces, Namespaces
- nested functions, Nested functions and nested scopes
- parameters, Parameters
- positional and named arguments, Positional and named arguments-Positional and named arguments
- recursion, Recursion
- return statement, The return Statement

functools module, The functools Module-The functools Module

Future class (concurrent.futures module), The concurrent.futures Module

fuzzing, Unit tests dealing with large amounts of data

G

garbage collection, Variables, del Statements, Garbage Collection-The weakref Module

gc module, The gc Module-The weakref Module

generator expressions (genexps), List comprehensions, Generator expressions

generator functions, Generators

generator objects, Generators

generators, Generators-Generator expressions, Generators and Exceptions, The itertools Module

Generic (typing module), Defining Custom Types

get (dict method), Dictionary Methods

getargspec (inspect module), The inspect Module

getargvalues (inspect module), The inspect Module

getattr (built-in function), Built-in Functions

getch (msvcrt module), The msvcrt module

getcwd (os module), File and directory functions of the os module

getdefaultlocale (locale module), The locale Module

getdoc (inspect module), The inspect Module

getfile (inspect module), The inspect Module

gethostbyaddr (socket module), The socket Module

gethostbyname (socket module), The socket Module getlocale (locale module), The locale Module

getmembers (inspect module), The inspect Module

getmodule (inspect module), The inspect Module

getmro (inspect module), The inspect Module

getpass (getpass module), The getpass Module

getpass module, The getpass Module

getrecursion limit (sys module), The sys Module

getrefcount (sys module), The sys Module, Garbage Collection

getsizeof (sys module), The sys Module

getsource (inspect module), The inspect Module

getsourcefile (inspect module), The inspect Module

getsourcelines (inspect module), The inspect Module

gettext module, The gettext Module-Essential gettext functions

getuser (getpass module), The getpass Module

getvalue (io.BytesIO, io.StringIO), In-Memory Files: io.StringIO and io.BytesIO

getweakrefcount (weakref module), The weakref Module

getweakrefs (weakref module), The weakref Module

get_stats (gc module), The gc Module

get_threshold (gc module), The gc Module

GIL (global interpreter lock), Concurrency: Threads and Processes

GitHub repository, How to Contact Us, Packaging Programs and Extensions

glob (glob module), The glob Module

glob (pathlib.Path class), The pathlib Module

global interpreter lock (GIL), Concurrency: Threads and Processes

global variables, Variables, Namespaces, Avoiding exec

globals (built-in function), Built-in Functions

gmpy2, Additional numeric packages

GNU Public License (GPL), Licensing and Price Issues

GNU Readline Library, The readline Module

GraalVM Python, Python Implementations, Jython and IronPython

gradual typing, Adding Type Annotations to Existing Code (Gradual Typing)

greedy matching, Common Regular Expression Idioms

group patterns, Group patterns

groupby (itertools module), The itertools Module

grouping option, Grouping option

groups, Groups

gRPC protocol, Other Network Protocols

grpcio module, Other Network Protocols

guards, Short-Circuiting Operators, The match Statement, Guards

Gunicorn, WSGI Servers

H

hardlink_to (pathlib.Path class), The pathlib Module

hasattr (built-in function), Built-in Functions

hash (built-in function), Built-in Functions

hash sign (#), Lines and Indentation, Delimiters

hashing

- hashable items (sets), Sets
- hashable keys, Dictionaries
- special methods for, General-Purpose Special Methods

hatch tool, Packaging Programs and Extensions

header lines, The if Statement

headers, Compound statements, The email.message Module

heapq module, The heapq Module-The Decorate–Sort–Undecorate Idiom

help (built-in function), Interactive Sessions, Built-in Functions

help argument, The argparse Module

help, getting, Mailing lists

hex (built-in function), Built-in Functions

hex (float method), Floating-point numbers

hexadecimal literals, Integer numbers

- escape sequence (\x XX), Strings

higher-order functions, Decorators, The operator Module

home (pathlib.Path class), The pathlib Module

homoglyphs, Identifiers

hot spots, Optimization

HTML (HyperText Markup Language), Structured Text: HTML, Generating HTML-Building HTML with jinja2

html.entities module, The html.entities Module

HTTP (Hypertext Transfer Protocol), **Serving HTTP**

http.server module, **http.server**

httpx module, **HTTP and URL Clients**

Hypercorn, **FastAPI**

HyperText Markup Language (HTML), **Structured Text: HTML**, **Generating HTML**-**Building HTML with jinja2**

Hypertext Transfer Protocol (HTTP), **Serving HTTP**

I

I/O (input/output), **File and Text Operations** (see also console I/O; file and text operations; richer-text I/O)

I/O-bound programs, **Concurrency: Threads and Processes**, **Optimizing I/O**

i18n (internationalization), **Formatted String Literals (F-Strings)**, **Internationalization**-**More Internationalization Resources**

id (built-in function), **Built-in Functions**

identifiers, **Identifiers**, **Plain assignment**

IDLE (Integrated Development and Learning Environment), **IDLE**

if statement, **The if Statement**

- if/else conditional expression, **Comparison Chaining**-**The conditional operator**

ignore end of line (\<newline>), **Strings**

IMAP4 (Internet Message Access Protocol version 4), **Email Protocols**

imaplib module, **Email Protocols**

immutable objects, **Data Types**

immutable strings, **Strings**

import

- absolute versus relative imports, **Absolute Versus Relative Imports**
- from M import, **from...import ***
- import failures, **Handling import failures**
- import hooks, **Import hooks**
- import statement, **The import Statement**-**Module-private variables**, **from versus import**
- importer factory, **Import hooks**
- __import__ function, **Built-in Functions**

in keyword

- dict membership, **Dictionary Membership**
- in for statement, **The for Statement**
- membership testing, **Expressions and Operators**, **Membership testing**
- searching and sorting, **Searching and sorting**
- set membership, **Set Membership**

in-memory files, **In-Memory Files: io.StringIO and io.BytesIO**

in-place assignment, **Augmented assignment**

indentation, **Lines and Indentation**

index (list method), **List methods**

index (string method), **Methods of String Objects**

indexfile, **Examples of DBM-Like File Use**

indexing

- arrays, **Shape, indexing, and slicing**
- dictionaries, **Indexing a Dictionary**
- index, **Object attributes and items**
- in plain assignment, **Plain assignment**
- sequences, **Indexing a sequence**

inequality operator (!=), **Comparisons**

inf (math module), **The math and cmath Modules**

infj (cmath module), **The math and cmath Modules**

inheritance

- basics of, **Python Classes**
- from built-in types, **Inheritance from Built-in Types**
- checking instances to support, **Built-in Types**
- custom exceptions and multiple inheritance, **Custom Exceptions and Multiple Inheritance**
- details of, **Inheritance**-**“Deleting” class attributes**

initialization, **General-Purpose Special Methods**

input (built-in function), **Built-in Functions**

input (fileinput module), **The fileinput Module**

input/output (I/O), **File and Text Operations** (see also console I/O; file and text operations; richer-text I/O)

insert (list method), **List methods**

inspect module, **The Frame Type**, **The inspect Module**-**An example of using inspect**

installation, Installation-Installing after the build
  from binaries, Installing Python from Binaries
  from source code, Installing Python from Source Code
  macOS, Installing Python from Binaries
  Microsoft Windows, Microsoft Windows
  pre-installed versions, Installation
  Unix-like platforms, Unix-Like Platforms
installers, Other Developments, Implementations, and Distributions
instances
  basics of, Python Classes
  details of, Instances-__new__
  per instance methods, Per Instance Methods
instantiation, Python Classes
instants (time), Time Operations
int (built-in type), Built-in Types
int type
  int methods, Integer numbers
integer numbers, Integer numbers, Format type
Integrated Development and Learning Environment (IDLE), IDLE
integration testing, Unit Testing and System Testing
interactive sessions, Interactive Sessions-Interactive Sessions, Per-Site Customization
internal types, Internal Types
internationalization (i18n), Formatted String Literals (F-Strings), Internationalization-More Internationalization Resources
Internet Message Access Protocol version 4 (IMAP4), Email Protocols
internet sockets, The Berkeley Socket Interface
interpreter (see Python interpreter)
interprocess communication (IPC), Concurrency: Threads and Processes, Using mmap Objects for IPC
intersection (set method), Set Methods
intervals (time), Time Operations
introspection, Debugging
io module, The io Module-File-Like Objects and Polymorphism
  io.BytesIO, In-Memory Files: io.StringIO and io.BytesIO

io.StringIO, In-Memory Files: io.StringIO and io.BytesIO

IPC (interprocess communication), Concurrency: Threads and Processes, Using mmap Objects for IPC

ipdb debugger, ipdb

IPython, IPython

IronPython, Jython and IronPython

is keyword, Enumerated Types (Enums), Built-in Types, The os.path Module, The socket Module-The socket Module

isclose (math module), Floating-Point Values, The math and cmath Modules

isdir (os.path module), The os.path Module

isdisjoint (set method), Set Methods

isenabled (gc module), The gc Module

isfile (os.path module), The os.path Module

isfirstline (fileinput module), The fileinput Module

isinstance (built-in function), Instances, Built-in Types, Built-in Functions

islice (itertools module), The itertools Module

isocalendar (datetime.date class), The date Class

isocalendar (datetime.datetime class), The datetime Class

isoformat (datetime.date class), The date Class

isoformat (datetime.datetime class), The datetime Class

isoformat (datetime.time class), The time Class

isoweekday (datetime.date class), The date Class

isoweekday (datetime.datetime class), The datetime Class

isstdin (fileinput module), The fileinput Module

issubclass (built-in function), Built-in Functions

issubset (set method), Set Methods

issuperset (set method), Set Methods

is_dir (pathlib.Path class), The pathlib Module

is_file (pathlib.Path class), The pathlib Module

is_integer (float method), Floating-point numbers

itemgetter (operator module), The operator Module

items, Data Types, Object attributes and items

items (dict method), Dictionary Methods

iter (built-in function), Built-in Functions

iterables, Iterables

iterators, Iterators, Short-circuiting of iterators

itertools module, The itertools Module-The itertools Module

It’s easier to ask forgiveness than permission (EAFP), LBYL Versus EAFP, The queue Module

J

jinja2 package, The jinja2 Package-Building HTML with jinja2

join (os.path module), The os.path Module

join (string method), Methods of String Objects

json module, The json Module-A json example

Jupyter, Jupyter

Jython, Jython and IronPython

K

key argument

- heapq, The Decorate–Sort–Undecorate Idiom
- itertools.groupby, The itertools Module
- sequence comparisons, Sequences in General
- sorting, List methods

key value pairs, Dictionaries

key: pattern pairs, Mapping patterns

keydefaultdict class (see default dict)

keyed access, DBM Modules

keys, Dictionaries, Object attributes and items

keys (dict method), Dictionary Methods

keyword module, Keywords

keyword parameters, Named parameters

keyword-only parameters, “Keyword-only” parameters

kill commands, try/except

L

l10n (localization), The gettext Module-Essential gettext functions

lambda, Expressions and Operators, lambda Expressions, Type Annotation Syntax

last-in, first-out (LIFO), Recursion

lazy evaluation, Generators, The itertools Module

LBYL (look before you leap), LBYL Versus EAFP, The queue Module

lc-clauses (in list comprehensions), List comprehensions

leading zero indicator (0) (f-strings), Leading zero indicator (0)

lefthand side (LHS), Plain assignment

legacy string formatting, Legacy String Formatting with %

len (built-in function), Sequences in General, Built-in Functions

less than or equal to operator (<=), Comparisons

less than symbol (<)

- alignment indicator (f-strings), Fill and alignment
- comparison operator, Comparisons

Lesser GPL (LGPL), Licensing and Price Issues

lexical structure, Lexical Structure-Compound statements

- character sets, Character Sets
- definition of term, Lexical Structure
- lines and indentation, Lines and Indentation
- statements, Statements
- tokens, Tokens

lexicographic comparison, Sequences

LGPL (Lesser GPL), Licensing and Price Issues

LHS (lefthand side), Plain assignment

licensing issues, Licensing and Price Issues

LIFO (last-in, first-out), Recursion

lightweight frameworks, “Full-Stack” Versus “Lightweight” Frameworks-FastAPI

lightweight processes, Concurrency: Threads and Processes

linear time performance (O(N)), Large-Scale Optimization

lineno (fileinput module), The fileinput Module

lines, Lines and Indentation

linesep (os module), Path-string attributes of the os module

link (os module), File and directory functions of the os module

list (built-in type), Built-in Types

list comprehension filters, := in a list comprehension filter

list comprehensions (listcomps), List comprehensions-List comprehensions

listdir (os module), File and directory functions of the os module

lists
- basics of, Lists
- versus generator expressions, List comprehensions
- in-place operations on, In-place operations on a list
- methods, List methods
- modifying, Modifying a list
- optimizing operations, List operations
- sorting, Sorting a list

Literal (typing module), Types

literals, Literals

literal_eval function (ast module), Standard Input

little-endian byte order (<), The struct Module

ljust (string method), Methods of String Objects

load testing, Benchmarking

local namespace/scope, Namespaces

local scope, Namespaces

local variables, Variables

locale module, The locale Module-The locale Module

localization (l10n), The gettext Module-Essential gettext functions

localize (locale module), The locale Module

locals (built-in function), Built-in Functions

Lock (threading module), Lock and RLock objects-Condition objects

Lock objects, Lock and RLock objects

locked (threading module), Lock and RLock objects

log (math module), The math and cmath Modules

logging
- debugging information, Debugging
- errors, Logging Errors-Configuring logging
- logging module, The logging module

logical lines, Lines and Indentation

look before you leap (LBYL), LBYL Versus EAFP, The queue Module

lookup tables, Precomputing a lookup table

loops
- else clause, The else Clause on Loop Statements

floating-point values and, Floating-Point Values
optimizing, Optimizing loops-Optimizing loops
while statement, The while Statement
lossless compression, File and Text Operations
lower (string method), Methods of String Objects
lowercase letters, Identifiers-Identifiers
lru_cache (functools module), The functools Module
lstrip (string method), Methods of String Objects
LZMA file compression, File and Text Operations

M

mailing lists, Mailing lists
main program, The Main Program
Manager class (multiprocessing module), The Manager class
map (built-in function), Built-in Functions
mapping patterns, Mapping patterns
mappings
in dictionaries, Dictionaries
special methods for containers, Mappings
markup, Structured Text: HTML
marshaling, Serialization, Pickling customization with the copyreg module
match, The match Statement
as patterns, as patterns
building patterns, Building patterns
capture patterns, Capture patterns
class patterns, Class patterns
configuring classes for positional matching, Configuring classes for positional matching
group patterns, Group patterns
guards, Guards
literal patterns, Literal patterns
mapping patterns, Mapping patterns
OR patterns, OR patterns
pattern matching with, The match Statement
in regular expressions, Regular Expression Objects
sequence patterns, Sequence patterns

- value patterns, **Value patterns**
- wildcard pattern, **The wildcard pattern**

match objects, **Match Objects**-**Match Objects**

math module, **The math and cmath Modules**-**The math and cmath Modules**

matrix multiplication, **The operator Module**, **Matrix operations in NumPy**

max (built-in function), **Sequences in General**, **Built-in Functions**

membership testing, **Membership testing**

memoization, **The functools Module**, **Memoizing**-**Memoizing**

memory-mapped files, **Concurrency: Threads and Processes**, **The mmap Module**

memoryview (built-in type), **Built-in Types**

merge (heaapq module), **The heapq Module**

metaclasses, **Metaclasses**-**Enumerated Types (Enums)**

- alternatives to, **Alternatives to Custom Metaclasses for Simple Class Customization**
- class creation by, **How a Metaclass Creates a Class**
- custom metaclass example, **A substantial custom metaclass example**
- data classes, **Data Classes**
- defining and using your own, **Defining and using your own metaclasses**
- definition of term, **Metaclasses**

methodcaller (operator module), **The operator Module**

methods

- basics of, **Python Classes**
- bound and unbound methods, **Bound and Unbound Methods**-**Bound and Unbound Methods**
- class-level methods, **Class-Level Methods**
- definition of term, **Object attributes and items**, **Functions**
- function definitions in class bodies, **Function definitions in a class body**
- mixin methods, **ABCs in the collections module**
- of file objects, **Attributes and Methods of File Objects**-**Attributes and Methods of File Objects**
- per instance methods, **Per Instance Methods**

special methods, Callables
MicroPython, MicroPython
Microsoft Windows
installing Python, Microsoft Windows
py launcher, The Windows py Launcher
python.exe, The python Program
running Python programs, Running Python Programs
virtual environments, Working with Virtual Environments
middle dot (·), Identifiers
middleware, WSGI Servers
migration, v3.7 to v3.n Migration-Summary
MIME (Multipurpose Internet Mail Extensions), Email, MIME, and Other Network Encodings
min (built-in function), Sequences in General, Built-in Functions
Miniconda, Anaconda and Miniconda
mixin methods, ABCs in the collections module
mkdir (os module), File and directory functions of the os module
mkdir (pathlib.Path class), The pathlib Module
mkdtemp (tempfile module), The tempfile Module
mkstemp (tempfile module), The tempfile Module
mmap module, The mmap Module-Using mmap Objects for IPC
mocks, Unit Testing and System Testing
mode (files), Permissions
modf (math module), The math and cmath Modules
module spec concept, Import hooks
modules (see also extension modules; standard library modules)
attribute (sys module), The sys Module
basics of, Modules and Packages
built-in versus extension, Core Built-ins and Standard Library Modules
DBAPI-compliant modules, DBAPI-Compliant Modules
definition of term, Running Python Programs, Modules and Packages
distutils and setuptools, Distribution Utilities (distutils) and setuptools-Distribution Utilities (distutils) and setuptools
module objects, Module Objects-from versus import
module-loading operations, Module Loading-Import hooks

- Python environments, Python Environments-Best Practices with Virtualenvs
- sharing objects among, Packages

monetary formatting, The decimal Module

MonkeyType, Other Type Checkers

monotonic (time module), The time Module

most_common (collections.Counter class), Counter

move function (shutil module), The shutil Module

msvcrt module, The msvcrt module

multipart messages, The email.message Module

multiple inheritance, Custom Exceptions and Multiple Inheritance

multiprocessing module, The multiprocessing Module-The ThreadPool class

- basics of, The multiprocessing Module
- multiprocessing versus threading, Differences Between multiprocessing and threading
- process pools, Process Pools
- sharing state, Sharing State: Classes Value, Array, and Manager

Multipurpose Internet Mail Extensions (MIME), Email, MIME, and Other Network Encodings

multithreading, Concurrency: Threads and Processes (see also threads and processes)

multiversion testing, Multiversion Testing

mutable objects, Data Types

mutual reference loops, Garbage Collection

mypy utility, History-mypy

MySQL modules, DBAPI-Compliant Modules

N

naive (see datetime module)

named argument collector, Named argument collector

named arguments, Dictionaries, Positional and named arguments

named parameters, Named parameters

NamedTemporaryFile class (tempfile module), The tempfile Module

NamedTuple, NamedTuple, TypedDict

NamedTuple (typing module), Defining Custom Types

names and naming

- avoiding reuse of built-in names, Built-in Functions
- of classes, Identifiers
- of identifiers, Identifiers
- project .py files, Searching the Filesystem for a Module

namespaces, Namespaces-lambda Expressions

nan (math module), The math and cmath Modules

nanj (cmath module), The math and cmath Modules

navigable classes (BeautifulSoup), The Navigable Classes of bs4-next_element, previous_element, next_elements, and previous_elements

ndarray class (NumPy package), NumPy

negative numbers, Sign indication

negative zero, Zero normalization (z)

nested format specifications, Nested Format Specifications

nested functions, Nested functions and nested scopes

nested scopes, Nested functions and nested scopes

net location (netloc), URL Access

network encodings

- encoding binary data as ASCII text, Encoding Binary Data as ASCII Text-The uu Module
- MIME and email format handling, MIME and Email Format Handling-Example Uses of the email Package

network endpoints, Networking Basics

Network News Transfer Protocol (NNTP), Other Network Protocols, MIME and Email Format Handling

network protocol modules (client-side)

- availability of, Client-Side Network Protocol Modules
- email protocols, Email Protocols-The smtplib Module
- HTTP and URL clients, HTTP and URL Clients-The Response class
- other network protocols, Other Network Protocols

Network Time Protocol (NTP), Other Network Protocols

networking basics
- Berkeley socket interface, The Berkeley Socket Interface-A Connection-Oriented Socket Server
- overview of, Networking Basics
- SSLContext class, SSLContext
- transport layer security (TLS), Transport Layer Security

new features
- significant changes, Significant Changes in Python Through 3.11
- version notations in text, Version Conventions
- versions 3.7 to 3.11, New Features and Changes in Python 3.7 Through 3.11-Python 3.11

newline (\n), Strings

NewType, NewType

NewType (typing module), Utilities and Decorators

next (built-in function), Built-in Functions

nextafter (math module), The math and cmath Modules

nextfile (fileinput module), The fileinput Module

NFKC normalization, Identifiers

nlargest (heapq module), The heapq Module

NNTP (Network News Transfer Protocol), Other Network Protocols, MIME and Email Format Handling

non-multipart messages, The email.encoders Module

nonblocking operations, Concurrency: Threads and Processes

None, None

nongreedy matching, Common Regular Expression Idioms

nonlocal keyword, Nested functions and nested scopes

nonoverriding descriptors, Overriding and nonoverriding descriptors

nonsequential file access, Sequential and nonsequential ("random") access

normalization strategies, Identifiers, Zero normalization (z)

normpath (os.path module), The os.path Module

nose2 utility and framework, Testing with nose2

not, Expressions and Operators, Enumerated Types (Enums)

not equal to operator (!=), Comparisons

notify (threading module), Condition objects

NotImplemented, **General-Purpose Special Methods**, **The abc module**, **Special Methods for Numeric Objects**, **Special Methods for Numeric Objects**

now (datetime.datetime class), **The datetime Class**

nsmallest (heapq module), **The heapq Module**

NTP (Network Time Protocol), **Other Network Protocols**

ntplib module, **Other Network Protocols**

Nuitka, **Python Implementations**

null objects (none), **None**

nullcontext (contextlib module), **The with Statement and Context Managers**

Numba, **Numba**, **Additional numeric packages**

numbers, **Numbers**-**Underscores in numeric literals**, **Sign indication**

numbers module, **ABCs in the numbers module**

numeric conversions, **Numeric Conversions**

numeric objects, special methods for, **Special Methods for Numeric Objects**-**Special Methods for Numeric Objects**

numeric operations, **Numeric Operations**-**Bitwise Operations on Integers**

- arithmetic operations, **Arithmetic Operations**
- bitwise operations on integers, **Bitwise Operations on Integers**
- numeric conversions, **Numeric Conversions**

numeric processing

- array processing, **Array Processing**-**Additional numeric packages**
- decimal module, **The decimal Module**-**The decimal Module**
- floating-point values, **Floating-Point Values**
- fractions module, **The fractions Module**
- math and cmath modules, **The math and cmath Modules**-**The math and cmath Modules**
- operator module, **The operator Module**-**The operator Module**
- random and pseudorandom numbers, **Random and Pseudorandom Numbers**-**Crypto-Quality Random Numbers: The secrets Module**
- statistics module, **The statistics Module**

NumPy, **NumPy**-**Matrix operations in NumPy**

O

object (built-in type), Built-in Types

object-oriented Python (see also Python language)

- benefits of, Object-Oriented Python
- classes and instances, Classes and Instances-Inheritance from Built-in Types
- decorators, Decorators-Decorators
- metaclasses, Metaclasses-Enumerated Types (Enums)
- polymorphism, File-Like Objects and Polymorphism
- special methods, Special Methods-Special Methods for Numeric Objects

objects

- attributes of, Object attributes and items
- built-in object type, Python built-ins
- deleting, del Statements
- first-class objects, Plain assignment, Functions
- items of, Object attributes and items
- mutable and immutable, Data Types
- null objects, None

oct (built-in function), Built-in Functions

octal value (\DDD escape sequence in strings), Strings

online material

- extending and embedding Python, Online Material
- packaging programs and extensions, Packaging Programs and Extensions

open (built-in function), Built-in Functions

open (pathlib.Path class), The pathlib Module

open (with open), Creating a File Object with open

Open DataBase Connectivity (ODBC), DBAPI-Compliant Modules

operator module, The operator Module-The operator Module

operators

- precedence (in expressions), Expressions and Operators
- short-circuiting operators, Short-Circuiting Operators

optimization

- avoiding premature, Optimization
- benchmarking, Benchmarking

definition of term, Testing, Debugging, and Optimizing
fast-enough applications, Developing a Fast-Enough Python Application-Developing a Fast-Enough Python Application
large-scale, Large-Scale Optimization-Summary of big-O times for operations on Python built-in types
profiling, Profiling-The pstats module
small-scale, Small-Scale Optimization-Memoizing

Optional (typing module), Type Expression Parameters

optional parameters, Named parameters

options, command line
py launcher, The Windows py Launcher
Python interpreter, Command-Line Syntax and Options-Command-Line Syntax and Options

or, Expressions and Operators

OR patterns, OR patterns

ord (built-in function), Built-in Functions

OrderedDict class (collections module), OrderedDict

os module
basics of, The os Module
filesystem operations, Filesystem Operations-File descriptor operations
os.path module, The os.path Module-The os.path Module
(see also pathlib module)
OSError exceptions, OSError Exceptions
in process environments, Process Environment
running programs with, Running Other Programs with the os Module

OSError, OSError subclasses, The io Module, OSError Exceptions

Oso, Considerations When Using Lightweight Frameworks

outer functions, Nested functions and nested scopes

overload (typing module), Utilities and Decorators

overriding descriptors, Overriding and nonoverriding descriptors

owning threads, Lock and RLock objects, Threaded Program Architecture

P

pack function (struct module), The struct Module

packages

- absolute versus relative imports, Absolute Versus Relative Imports
- basics of, Packages
- definition of term, Modules and Packages
- distutils and setuptools, Distribution Utilities (distutils) and setuptools-Distribution Utilities (distutils) and setuptools
- Python environments, Python Environments-Best Practices with Virtualenvs
- sharing objects among modules in, Packages
- special attributes of package objects, Special Attributes of Package Objects

packaging programs and extensions, Packaging Programs and Extensions-Online Material

packet sniffing, The Berkeley Socket Interface

pack_into function (struct module), The struct Module

pairs, Tuples

pairwise (itertools module), The itertools Module

parameter style (DBAPI-compliant), Parameter Style

parameter substitution, Parameter Style

parameters, Parameters-Argument collector parameters, “Keyword-only” parameters-Matching arguments to parameters

paramiko package, Other Network Protocols

parentheses (())

- function calls, Calling Functions
- in generator expressions, Generator expressions
- lines and indentation, Lines and Indentation

parents, The class Statement

parse_and_bind (readline module), The readline Module

partial (functools module), The functools Module

parties (threading module), Barrier objects

pass statement, The pass Statement

path (sys module), The sys Module

PATH environment variable, Python Environments path patterns, **The glob Module**

pathlib module, **The pathlib Module**-**The pathlib Module**

pathlib.Path class, **The pathlib Module**

paths, **Path-string attributes of the os module**-**File and directory functions of the os module**

pattern expressions, **The match Statement**

(see also match)

pattern matching (see fnmatch module; match; regular expressions)

pattern string syntax, **Pattern String Syntax**

payload, **The email.message Module**

pdb module, **The pdb Module**-**The pdb Module**

peer-to-peer networking, **Client/Server Computing**

PEPs (Python Enhancement Proposals), **Python Development and Versions**

per instance methods, **Per Instance Methods**

per-site customization, **Per-Site Customization**

per-thread data, **Thread Local Storage**

percent sign (%)

legacy string format operator, **Legacy String Formatting with %**

log message placeholder, **The logging module**

modulus division operator, **Division**

SQL statement placeholder, **Parameter Style**

perf_counter (time module), **The time Module**

period (.)

floating-point numbers, **Floating-point numbers**

import statement, **The import Statement**

object attributes, **Object attributes and items**

treatment by glob module, **The glob Module**

perm (math module), **The math and cmath Modules**

permission bits, **Permissions**

persistence and databases

DBM modules, **DBM Modules**-**Examples of DBM-Like File Use**

overview of, **Persistence and Databases**

Python Database API 2.0 standard (DBAPI), **The Python Database API (DBAPI)**-**A sqlite3 example**

serialization, **Serialization**-**A shelving example** phase (cmath module), The math and cmath Modules

physical lines, Lines and Indentation

pickle module, The pickle Module-Pickling customization with the copyreg module

pipenv package, Other Environment Management Solutions

pipes, Concurrency: Threads and Processes

placeholders, Templating

plain assignment statements, Plain assignment

platform (sys module), The sys Module

plus sign (+), Concatenation and repetition

polar (cmath module), The math and cmath Modules

polling, Threaded Program Architecture

polymorphism, File-Like Objects and Polymorphism

Pool class (multiprocessing module), Process Pools

pop method

- dict type, Dictionary Methods
- list type, List methods
- set type, Set Methods

POP3 (Post Office Protocol version 3), Email Protocols

popen (os module), Running Other Programs with the os Module

Popen class (subprocess module), Using the Subprocess Module

popitem (dict method), Dictionary Methods

poplib module, The poplib Module

port numbers, The Berkeley Socket Interface

positional argument collector, Positional argument collector

positional arguments, Positional and named arguments

positional matching, Configuring classes for positional matching

positional parameters, Positional parameters

- positional-only parameters, Positional-only marker

positional-only marker (/), Positional-only marker

positive numbers, Sign indication

Post Office Protocol version 3 (POP3), Email Protocols

PostgreSQL modules, DBAPI-Compliant Modules

pow (built-in function), Built-in Functions

pow (math module), The math and cmath Modules

pprint module, The pprint Module

pre-commit package, Tools for Checking Python Programs precision specification (f-strings), **Precision specification**

price issues, **Licensing and Price Issues**

print (built-in function), **Built-in Functions**, **The print Function**

private versus public variables, **Module-private variables**

procedural programming, **Object-Oriented Python**

Process class (multiprocessing module), **The Process class**

process environment, **Process Environment**

process pools, **Process Pools**

processes, definition of term, **Concurrency: Threads and Processes** (see also threads and processes)

process_time (time module), **The time Module**

profile module, **The profile module**

profiling, **Developing a Fast-Enough Python Application**, **Profiling**-**The pstats module**

programs

- block structure of, **Lines and Indentation**
- CPU-bound programs, **Concurrency: Threads and Processes**
- I/O-bound programs, **Concurrency: Threads and Processes**, **Optimizing I/O**
- running, **Running Python Programs**
- running in browsers, **Running Python in the Browser**
- threaded program architecture, **Threaded Program Architecture**-**Threaded Program Architecture**
- tools for checking, **Tools for Checking Python Programs**, **Before You Debug**

prompt (>>>), **Interactive Sessions**, **Simple statements**

properties, **Properties**-**Properties and inheritance**

protocol adapters, **The Third-Party requests Package**

proxy (weakref module), **The weakref Module**

pseudorandom numbers, **Random and Pseudorandom Numbers**-**Crypto-Quality Random Numbers: The secrets Module**, **Unit tests dealing with large amounts of data**

pstats module, **The pstats module**-**The pstats module**

pudb debugger, **pudb**

py launcher (Windows), **The Windows py Launcher**

pydantic, **Other Type Checkers**, **FastAPI**

pyenv, **pyenv: Simple support for multiple versions**

pyflakes tool, Tools for Checking Python Programs
pyftpdlib module, HTTP and URL Clients
PyICU, More Internationalization Resources
Pyjion, Pyjion
Pylance, Other Type Checkers
PyPA (Python Packaging Authority), Workgroups, Packaging Programs and Extensions
PyPI (Python Package Index), Extension modules and Python sources, Client-Side Network Protocol Modules
PyPy, PyPy, Documentation, The PyPy Interpreter
Pyramid, A Few Popular Full-Stack Frameworks
Pyre, Other Type Checkers
Pyright, Other Type Checkers
PyScript, PyScript
pysnmp module, Other Network Protocols
pytest package, Testing with pytest
Pythex site, Regular Expressions and the re Module
Python
- chapter overviews, How To Use This Book
- development and versions, Python Development and Versions
- extending and embedding, Extending and Embedding Classic Python-Online Material
- implementations of, Python Implementations-Transcrypt: Convert your Python to JavaScript
- installation, Installation-Installing after the build
- introduction to, Introduction to Python-The Python Language
- licensing and price issues, Licensing and Price Issues
- resources, Python Resources-Social media
- significant changes through 3.11, Significant Changes in Python Through 3.11-Significant Changes in Python Through 3.11, New Features and Changes in Python 3.7 Through 3.11-Python 3.11
- standard library and extension modules, The Python Standard Library and Extension Modules
- upgrading, Planning a Python Version Upgrade-How Often Should You Upgrade?
- v3.7-v3.n migration, v3.7 to v3.n Migration-Summary

versions covered, Version Conventions

Python community, Community-Social media, Other Network Protocols

Python Database API 2.0 standard (DBAPI) (see Database API)

Python Developer's Guide, Python Development and Versions

Python Education workgroup, Workgroups

Python Enhancement Proposals (PEPs), Python Development and Versions

Python environments, Python Environments-Other Environment Management Solutions

other environment solutions, Other Environment Management Solutions

Python interpreter

development environments, Python Development Environments-Tools for Checking Python Programs

for multiple environments, Python Environments

purpose of, The Python Interpreter

python program, The python Program-Interactive Sessions

running programs, Running Python Programs

running Python in browsers, Running Python in the Browser

Python language (see also object-oriented Python)

control flow statements, Control Flow Statements-The with Statement

data types, Data Types

dictionary operations, Dictionary Operations-Dictionary Methods

documentation, The Python Language

expressions and operators, Expressions and Operators-:= in a list comprehension filter

functions, Functions-Recursion

lexical structure, Lexical Structure-Compound statements

numeric operations, Numeric Operations-Bitwise Operations on Integers

sequence operations, Sequence Operations-Sorting a list

set operations, Set Operations-Set Methods

style guide, The Python Language

- Python Package Index (PyPI), **Extension modules and Python sources**, **Client-Side Network Protocol Modules**
- Python Packaging Authority (PyPA), **Workgroups**, **Packaging Programs and Extensions**
- python program, **The python Program**-**Interactive Sessions**
  - command-line syntax and options, **Command-Line Syntax and Options**
  - environment variables, **Environment Variables**
  - interactive sessions, **Interactive Sessions**, **Identifiers**
  - PyPy interpreter, **The PyPy Interpreter**
  - Windows py launcher, **The Windows py Launcher**
- Python Software Foundation (PSF)
  - development and versions, **Python Development and Versions**
  - licensing and price issues, **Licensing and Price Issues**
  - resources available, **Python Software Foundation**
- Python wheels, **Distribution Utilities (distutils) and setuptools**
- PyTorch, **Additional numeric packages**
- pytype, **Other Type Checkers**
- pyupgrade package, **Upgrade Automation Using pyupgrade**

Q

- quadratic time performance (O(N^2)), **Large-Scale Optimization**
- question mark (?), **Pattern String Syntax**, **The fnmatch Module**
- questions and comments, **How to Contact Us**
- queue module, **The queue Module**-**The queue Module**
- quopri module, **The quopri Module**
- Quoted Printable (QP) encoding, **The quopri Module**
- quotes
  - double quote ("), **Delimiters**, **Strings**
  - double quote escaped (\"), **Strings**
  - single quote ('), **Delimiters**, **Strings**
  - single quote escaped (\"), **Strings**
  - triple double quotes ("""), **Strings**
  - triple single quotes ('''), **Strings**

R

radix indicator (#) (f-strings), **Radix indicator (#)**

RAII (resource acquisition is initialization), **The with Statement and Context Managers**

raise statement, **The try and raise Statements**, **The raise Statement**

raise to power operator (**), **Exponentiation**

randint (random module), **The random Module**

random (random module), **The random Module**

random access, **File and Text Operations**, **Sequential and nonsequential ("random") access**

random module, **Random and Pseudorandom Numbers**-**Crypto-Quality Random Numbers: The secrets Module**

randrange (random module), **The random Module**

range (built-in function), **range**, **Built-in Functions**

rapid prototyping, **Developing a Fast-Enough Python Application**

raw sockets, **The Berkeley Socket Interface**

raw string literals, **Strings**

RDBMS (relational DB management system), **Persistence and Databases**, **Cursor Objects**, **DBAPI-Compliant Modules**

re module, **Regular Expressions and the re Module**-**Groups** (see also regular expressions)

read (file method), **Attributes and Methods of File Objects**

readline (file method), **Attributes and Methods of File Objects**

readline module, **The readline Module**

readlines (file method), **Attributes and Methods of File Objects**

readlink (pathlib.Path class), **The pathlib Module**

read_bytes (pathlib.Path class), **The pathlib Module**

read_history_file (readline module), **The readline Module**

read_text (pathlib.Path class), **The pathlib Module**

realpath (os.path module), **The os.path Module**

rebinding, **Variables**

rect (cmath module), **The math and cmath Modules**

recursion, **Recursion**-**Recursion**

redirect_stderr (contextlib module), **The with Statement and Context Managers** redirect_stdout (contextlib module), The with Statement and Context Managers

reduce (functools module), The functools Module

reentrant locks, Lock and RLock objects

ref (weakref module), The weakref Module

reference counts, Garbage Collection

reference loops, The weakref Module

references (see also variables and other references)

- accessing nonexistent, Accessing nonexistent references
- creating new, Assignment Statements
- definition of term, Variables and Other References
- unbinding, del Statements

reflection, Debugging

regex module, The Third-Party regex Module

regex101 site, Regular Expressions and the re Module

regexps, Regular Expressions

regular expressions (REs)

- := operator, REs and the := Operator
- anchoring at string start and end, Anchoring at String Start and End
- functions of re module, Functions of the re Module
- match objects, Match Objects-Match Objects
- match versus search, Match Versus Search
- optional flags, Optional Flags-Optional Flags
- re module, Regular Expressions and the re Module-Groups
- regex module, The Third-Party regex Module
- regular expression objects, Regular Expression Objects-Regular Expression Objects

relational DB management system (RDBMS), Persistence and Databases, Cursor Objects, DBAPI-Compliant Modules

relative imports, Absolute Versus Relative Imports

relative paths, Packages

release (threading module), Lock and RLock objects, Condition objects, Semaphore and BoundedSemaphore objects

relpath (os.path module), The os.path Module

remainder (math module), The math and cmath Modules

remote endpoints, The Berkeley Socket Interface remove (os module), **File and directory functions of the os module**

remove method

- list type, **List methods**
- set type, **Set Methods**

removeprefix (string method), **Methods of String Objects**

removessuffic method (string method), **Methods of String Objects**

rename (os module), **File and directory functions of the os module**

repeat (itertools module), **The itertools Module**

repetition, **Concatenation and repetition**

replace (datetime.date class), **The date Class**

replace (datetime.datetime class), **The datetime Class**

replace (datetime.time class), **The time Class**

replace method (string method), **Methods of String Objects**

repr (built-in function), **Built-in Functions**

reprlib module, **The reprlib Module**

reproducibility, **Unit tests dealing with large amounts of data**

requests package, **The Third-Party requests Package**-**The Response class**

- Response class, **The Response class**

reserved words, **Keywords**

resolve (pathlib.Path class), **The pathlib Module**

resource acquisition is initialization (RAII), **The with Statement and Context Managers**

resources

- books, **Books**
- community, **Community**-**Social media**, **Other Network Protocols**
- documentation, **Documentation**-**Books**
- new features and changes v3.7 to 3.11, **Version Conventions**, **Significant Changes in Python Through 3.11**-**Significant Changes in Python Through 3.11**, **New Features and Changes in Python 3.7 Through 3.11**-**Python 3.11**

restricted execution, **Never exec or eval Untrusted Code**

return statement, **The return Statement**

reverse (list method), **List methods**

reversed (built-in function), Built-in Functions

rglob (glob module), The glob Module

rglob (pathlib.Path class), The pathlib Module

RHS (righthand side), Plain assignment

richer-text I/O, Richer-Text I/O-The msvcrt module

righthand side (RHS), Plain assignment

rjust method (string method), Methods of String Objects

rmdir (os module), File and directory functions of the os module

rmdir (pathlib.Path class), The pathlib Module

rmtree function (shutil module), The shutil Module

root privileges, The Berkeley Socket Interface

round (built-in function), Built-in Functions

rounding values, Precision specification

rstrip method (string method), Methods of String Objects

runtime errors, Accessing nonexistent references

RuntimeWarning class, Classes

runtime_checkable (typing module), Utilities and Decorators

RustPython, Python Implementations

S

sample (random module), The random Module

scandir (os module), File and directory functions of the os module

sched module, The sched Module

SciPy, SciPy

SciTE text editor, Free Text Editors with Python Support

scope

  nested scope, Nested functions and nested scopes

  variable scope, List comprehensions

scraping, Structured Text: HTML

scripts, definition of term, Running Python Programs

search (re instance methods), Regular Expression Objects

search methods, bs4 find… Methods (aka Search Methods)-Other named arguments

searches, optimization of, Searching and sorting

secrets module, Crypto-Quality Random Numbers: The secrets Module

Secure Shell (SSH) protocol, Other Network Protocols

Secure Sockets Layer (SSL), Transport Layer Security

seed (random module), The random Module, Unit tests dealing with large amounts of data

seek (file method), Attributes and Methods of File Objects

self parameter, Iterators, Function definitions in a class body, The factory function idiom, Bound and Unbound Methods

semantic errors, Accessing nonexistent references

semantic markup, Structured Text: HTML

Semaphore (threading module), Semaphore and BoundedSemaphore objects

Semaphore objects, Semaphore and BoundedSemaphore objects

semicolon (;)

- in simple statements, Simple statements
- lines and indentation, Lines and Indentation

sep (os module), Path-string attributes of the os module

sequence operations, Sequence Operations-Sorting a list, Sequences

- lists, Lists
- sequences in general, Sequences in General
- special methods for containers, Sequences
- strings, Strings
- tuples, Tuples

sequence patterns, Sequence patterns

sequences, Sequences-Lists

sequential file access, Sequential and nonsequential (“random”) access

serialization, Serialization-A shelving example

- csv module, The csv Module
- json module, The json Module
- pickle module, The pickle Module
- shelve module, The shelve Module

serialized calls (threading), Threaded Program Architecture

server-side web programming (see web programming)

serving HTTP (see web programming)

set (built-in type), Built-in Types

set (threading module), Event objects

set comprehensions, **Set comprehensions**

set operations

- optimizing, **Set operations**
- set membership, **Set Membership**
- set methods, **Set Methods**
- special methods for containers, **Sets**

setattr (built-in function), **Built-in Functions**

setdefault (dict method), **Dictionary Methods**

setdefaulttimeout (socket module), **The socket Module**

setrecursionlimit (sys module), **Recursion**, **The sys Module**

sets, **Sets**

sets of characters, **Sets of Characters**

setUp (unittest.TestCase class), **The TestCase class**

setuptools, **Distribution Utilities (distutils) and setuptools**-**Distribution Utilities (distutils) and setuptools**

set_completer (readline module), **The readline Module**

set_debug (gc module), **The gc Module**

set_threshold (gc module), **The gc Module**

SGML (Standard Generalized Markup Language), **Structured Text: HTML**

shallow copies, **The copy Module**

shebang (#!), **Running Python Programs**, **Character Sets**

shell environments, **Python Environments**

shell utilities, **The shutil Module**-**The shutil Module**

shelve module, **The shelve Module**

short-circuiting operators, **Short-Circuiting Operators**

shuffle (random module), **The random Module**

shutil module, **The shutil Module**-**The shutil Module**

sign indication, **Sign indication**

signatures (see function signatures)

significand, **Floating-Point Values**

significant changes, **Version Conventions**, **Significant Changes in Python Through 3.11**-**Significant Changes in Python Through 3.11**, **New Features and Changes in Python 3.7 Through 3.11**-**Python 3.11**

significant digits, **Precision specification**

Simple Mail Transport Protocol (SMTP), **Email Protocols**

Simple Network Management Protocol (SNMP), Other Network Protocols
simple statements, Simple statements
sin (math module), The math and cmath Modules
single quote ('), Delimiters, Strings
    escaped (\'), Strings
single-line loops, List comprehensions
singledispatch (functools module), The functools Module
singledispatchmethod (functools module), The functools Module
site customization, Per-Site Customization
site module, Per-Site Customization
site-packages directory, Python Environments, Working with Virtual Environments
site.py, Per-Site Customization
sitecustomize.py, Per-Site Customization
slash (/)
    in arithmetic operations, Division
    in file paths, Command-Line Syntax and Options
    in glob module, The glob Module
    pathlib.Path join operator, The pathlib Module
sleep (time module), The time Module
slice (built-in type), Built-in Types
slicing
    arrays, Shape, indexing, and slicing
    plain assignment and, Plain assignment
    sequences, Slicing a sequence
    special methods for containers, Container slicing
SMTP (Simple Mail Transport Protocol), Email Protocols
smtplib module, The smtplib Module
snake case, Identifiers
SNMP (Simple Network Management Protocol), Other Network Protocols
social media, Social media
socket module, The socket Module-The socket Module
socket objects, Socket Objects-Socket Objects
sockets
    internet and Unix sockets, The Berkeley Socket Interface

- networking using, **The Berkeley Socket Interface**
- raw sockets, **The Berkeley Socket Interface**
- socket addresses, **Socket Addresses**
- socket clients
    - connection-oriented, **A Connection-Oriented Socket Client**
    - connectionless, **A Connectionless Socket Client**
- socket servers
    - connection-oriented, **A Connection-Oriented Socket Server**
    - connectionless, **A Connectionless Socket Server**

soft keywords, **Keywords**

sort (list method), **List methods**

sorted (built-in function), **Built-in Functions**

sorting, optimization of, **Searching and sorting**

source distribution, **Extension modules and Python sources**, **Installing Python from Source Code**

spaces, **Lines and Indentation**

spawn function (os module), **Running Other Programs with the os Module**

spawnv function (os module), **Running Other Programs with the os Module**

special methods, **Special Methods**-**Special Methods for Numeric Objects**

- abstract base classes (ABCs), **Abstract Base Classes**
- basics of, **Special Methods**
- for containers, **Special Methods for Containers**
- general-purpose, **General-Purpose Special Methods**
- for numeric objects, **Special Methods for Numeric Objects**

spidering, **Structured Text: HTML**

split (os.path module), **The os.path Module**

split (string method), **Methods of String Objects**

splitlines (string method), **Methods of String Objects**

SpooledTemporaryFile class (tempfile module), **The tempfile Module**

spur module, **Other Network Protocols**

SQL injection attacks, **Parameter Style**

SQLite library (sqlite3 module), **SQLite**-**A sqlite3 example**
sqlite3.Connection class, **The sqlite3.Connection class**
SSH (Secure Shell) protocol, **Other Network Protocols**
SSL (Secure Sockets Layer), **Transport Layer Security**
ssl module, **SSLContext**
SSLContext class (ssl module), **SSLContext**
stack (inspect module), **The inspect Module**
standard error, **Standard Output and Standard Error**
standard exceptions
- classes raised by runtime errors, **Standard Exception Classes**
- hierarchy of, **The Hierarchy of Standard Exceptions**
- OSError subclasses, **OSError subclasses**

Standard Generalized Markup Language (SGML), **Structured Text: HTML**
standard input, **Standard Input**
standard library modules
- abc, **The abc module**
- argparse, **The argparse Module**-**The argparse Module**
- ast, **Standard Input**
- atexit, **Termination Functions**
- base64, **The base64 Module**
- calendar, **The calendar Module**
- codecs, **The codecs Module**
- collections, **The collections Module**-**deque**
- contextlib, **The with Statement and Context Managers**
- copy, **The copy Module**-**The copy Module**
- copyreg, **Pickling customization with the copyreg module**
- csv, **The csv Module**-**A csv example**
- ctypes, **Online Material**
- curses, **curses**
- data classes, **Data Classes**
- datetime, **The datetime Module**-**The timezone Class**
- dbm, **DBM Modules**
- decimal, **The decimal Module**
- doctest, **The doctest Module**
- email, **Functions in the email Package**
- enum, **Enumerated Types (Enums)**

errno, The errno Module

filecmp, The filecmp Module-The filecmp Module

fileinput, The fileinput Module-The fileinput Module

fnmatch, The fnmatch Module

fractions, The fractions Module

ftplib, HTTP and URL Clients

functools, The functools Module-The functools Module

gc, The gc Module-Instrumenting garbage collection

getpass, The getpass Module

gettext, The gettext Module-Essential gettext functions

glob, The glob Module

html.entities, The html.entities Module

html.parser, Which parser BeautifulSoup uses

http.server, http.server

imaplib, Email Protocols

inspect, The Frame Type, The inspect Module

io, The io Module-File-Like Objects and Polymorphism

itertools, The itertools Module-The itertools Module

json, The json Module-A json example

keyword, Keywords

locale, The locale Module-The locale Module

logging, The logging module

math and cmath, The math and cmath Modules

mmap, The mmap Module

multiprocessing, The multiprocessing Module

ntplib, Other Network Protocols

numbers, ABCs in the numbers module

operator, The operator Module

os, The os Module-File descriptor operations

os.path, The os.path Module-The os.path Module

pathlib, The pathlib Module-The pathlib Module

pdb, The pdb Module

pickle, The pickle Module-Pickling customization with the copyreg module

poplib, The poplib Module

pprint, The pprint Module

profile, The profile module

pstats, The pstats module

queue, The queue Module

quopri, The quopri Module

random, Random and Pseudorandom Numbers

re, Regular Expressions and the re Module-Groups

readline, The readline Module

reprlib , The reprlib Module

sched, The sched Module

secrets, Crypto-Quality Random Numbers: The secrets Module

shelve, The shelve Module

shutil, The shutil Module-The shutil Module

site, Per-Site Customization

smtplib, The smtplib Module

socket, The socket Module

sqlite3, SQLite-A sqlite3 example

ssl, SSLContext

stat, The stat Module

statistics, The statistics Module

string, The string Module

struct, The struct Module-The struct Module

subprocess, Using the Subprocess Module

sys, The sys Module-The sys Module, Standard Output and Standard Error

tarfile, File and Text Operations

telnetlib, Other Network Protocols

tempfile, The tempfile Module-The tempfile Module

textwrap, Text Wrapping and Filling

threading, The threading Module

time, The time Module-The time Module

timeit, The timeit module

tomllib, Packaging Programs and Extensions

traceback, The Frame Type

unicodedata, The unicodedata Module

unittest, The unittest Module

urlib, The urllib Package

uu, The uu Module

warnings, The warnings Module
weakref, The weakref Module
xml.etree.ElementTree, ElementTree
zipfile, The zipfile Module-The ZipInfo class
zoneinfo, The zoneinfo Module
__future__, Applying the Code Changes
standard output, Standard Output and Standard Error
Starlette, FastAPI
starmap (itertools module), The itertools Module
starred targets, Plain assignment
start (threading module), Timer objects
startswith (string method), Methods of String Objects
stat (os module), File and directory functions of the os module
stat (pathlib.Path class), The pathlib Module
stat module, The stat Module
statements, Statements
statistics module, The statistics Module
stderr (sys module), The sys Module, Standard Output and Standard Error
stdin (sys module), The sys Module, Standard Input
stdout (sys module), The sys Module, Standard Output and Standard Error
str (built-in type), Built-in Types, Strings and Things
strftime (datetime.datetime class), The datetime Class
strftime (time module), The time Module
string module, The string Module
StringIO class (io module), In-Memory Files: io.StringIO and io.BytesIO
strings
building from pieces, Building up a string from pieces
built-in string types, Strings-Strings
methods of string objects, Methods of String Objects-Methods of String Objects
optimizing string operations, String operations
pprint module, The pprint Module
reprlib module, The reprlib Module
sequence operations, Strings

- special methods for string representation, General-Purpose Special Methods
- string formatting, String Formatting-Format Specifier Syntax, Parameter Style
- string module, The string Module
- text wrapping and filing, Text Wrapping and Filling
- Unicode, Unicode-The unicodedata Module

strip (string method), Methods of String Objects

strongly private identifiers, Identifiers

strptime (datetime.datetime class), The datetime Class

strptime (time module), The time Module

Struct class (struct module), The struct Module

struct format strings, The struct Module

struct module, The struct Module-The struct Module

structural pattern matching, The match Statement

structured text

- eXtensible Markup Language (XML)
  - building an ElementTree from scratch, Building an ElementTree from Scratch
  - ElementTree, ElementTree-Functions in the ElementTree Module
  - parsing XML iteratively, Parsing XML Iteratively
  - parsing XML with ElementTree.parse, Parsing XML with ElementTree.parse-Editing an ElementTree
  - working with XML, Structured Text: XML
- HyperText Markup Language (HTML)
  - BeautifulSoup package, The BeautifulSoup Third-Party Package-An HTML Parsing Example with BeautifulSoup
  - generating HTML, Generating HTML-Building HTML with jinja2
  - html.entities module, The html.entities Module
  - working with HTML, Structured Text: HTML

stubs, Unit Testing and System Testing

sub (re instance methods), Regular Expression Objects

Sublime Text text editor, Free Text Editors with Python Support

subpackages, Packages

subprocess module, Using the Subprocess Module-Running Other Programs with the os Module

subtract (collections.Counter class), Counter

sum (built-in function), Built-in Functions

super (built-in type), Built-in Types

superclasses, The class Statement

supplemental material, obtaining, How to Contact Us, Packaging Programs and Extensions

SupportsAbs and other protocols (typing module), Protocols

suppress (contextlib module), The with Statement and Context Managers

swapcase (string method), Methods of String Objects

symlink (os module), File and directory functions of the os module

symlink_to (pathlib.Path class), The pathlib Module

symmetric_difference (set method), Set Methods

synchronization objects (threads), Thread Synchronization Objects-Lock and RLock objects

syntax (see command-line syntax and options)

syntax errors, Accessing nonexistent references

SyntaxWarning class, Classes

sys module, The sys Module-The sys Module, Standard Output and Standard Error

sys.stderr, Standard Output and Standard Error

sys.stdout, Standard Output and Standard Error

system (os module), Running Other Programs with the os Module

system analysis and design, Testing, Debugging, and Optimizing

system testing, Testing-Unit Testing and System Testing

T

tab (\t), Strings

tables, Packaging Programs and Extensions

tabs, Lines and Indentation

tags, Structured Text: HTML

tail-call elimination, Recursion

takewhile (itertools module), The itertools Module

tan (math module), The math and cmath Modules

TarFile class, The TarFile class

tarfile module, File and Text Operations, The tarfile Module-The TarInfo class

targets, Plain assignment

TarInfo class, The TarInfo class

tasks, Concurrency: Threads and Processes

TCP (Transmission Control Protocol), Networking Basics

TCP/IP protocols, The Berkeley Socket Interface

TDD (test-driven development), Unit Testing and System Testing

tearDown (unittest.TestCase class), The unittest Module, The TestCase class

tee (itertools module), The itertools Module

tell (file method), Attributes and Methods of File Objects

telnetlib module, Other Network Protocols

tempfile module, The tempfile Module-The tempfile Module

templating, Templating-Building HTML with jinja2

temporary files, The io Module

TemporaryDirectory class (tempfile module), The tempfile Module

TemporaryFile class (tempfile module), The tempfile Module

TensorFlow, Additional numeric packages

termination functions (see atexit module)

ternary operator, Expressions and Operators

- (see also if/else)

test case methods, The unittest Module

test fixtures, The unittest Module

test frameworks, The doctest Module

test-driven development (TDD), Unit Testing and System Testing

TestCase class (unittest module), The unittest Module, The TestCase class-The TestCase class

testing

- definition of term, Testing, Debugging, and Optimizing
- doctest module, The doctest Module-The doctest Module
- multiversion testing, Multiversion Testing
- nose2 utility and framework, Testing with nose2
- pytest package, Testing with pytest
- unit testing and system testing, Testing-Unit Testing and System Testing

- unittest module, The unittest Module-Unit tests dealing with large amounts of data

testing frameworks, Unit Testing and System Testing, Testing with nose2-Testing with pytest

text editors, Free Text Editors with Python Support

text input and output, Text Input and Output-The getpass Module

(see also file and text operations; structured text)

text mode, Binary and text modes

textwrap module, Text Wrapping and Filling

Thread class (threading module), Thread Objects

thread local storage, Thread Local Storage

thread pools, Threaded Program Architecture

thread safety, Thread Safety

threading module, The threading Module-Thread Local Storage

- functions of, The threading Module
- Thread class, Thread Objects
- thread local storage, Thread Local Storage
- thread synchronization objects, Thread Synchronization Objects

ThreadPool class (multiprocessing.pool module), The ThreadPool class

threads and processes

- basics of, Concurrency: Threads and Processes
- concurrent.futures module, The concurrent.futures Module-The concurrent.futures Module
- mmap module, The mmap Module-Using mmap Objects for IPC
- multiprocessing module, The multiprocessing Module-The ThreadPool class
- process environments, Process Environment
- queue module, The queue Module-The queue Module
- running other programs, Running Other Programs-Running Other Programs with the os Module
- selecting the best approach, Concurrency: Threads and Processes
- threaded program architecture, Threaded Program Architecture-Threaded Program Architecture

threading module, The threading Module-Thread Local Storage
threads in Python, Threads in Python
using for optimization, Using multiprocessing for heavy CPU work
tilde (~), Expressions and Operators
time class (datetime module), The time Class
time method (datetime.datetime class), The datetime Class
time module, The time Module-The time Module
time operations
calendar module, The calendar Module
datetime module, The datetime Module-The timezone Class
dateutil module, The dateutil Module
sched module, The sched Module
time module, The time Module-The time Module
zoneinfo module, The zoneinfo Module
time zone, The datetime Module, The time Class-The zoneinfo Module
in email, The email.utils Module
timedelta (datetime.datetime class), The date Class, The timedelta Class
timeit module, The timeit module
Timer (threading module), Timer objects
timestamp (datetime.datetime class), The datetime Class
timetuple (datetime.datetime class), The datetime Class
timetuples, The time Module
timetz (datetime.datetime class), The datetime Class
timezone (time module), The time Module
timezone class, The timezone Class
time–memory trade-off, Memoizing
timsort algorithm, Sorting a list
title (string method), Methods of String Objects
tkinter GUI toolkit, Threaded Program Architecture
TLS (transport layer security), Transport Layer Security
today (datetime.date class), The date Class
today (datetime.datetime class), The datetime Class
tokens, Tokens-Literals

identifiers, **Identifiers**
keywords, **Keywords**
types of, **Tokens**
tomllib module, **Packaging Programs and Extensions**
toordinal (datetime.date class), **The date Class**
toordinal (datetime.datetime class), **The datetime Class**
total (collections.Counter class), **Counter**
total_ordering (functools module), **The functools Module**
touch (pathlib.Path class), **The pathlib Module**
to_bytes (int method), **Integer numbers**
traceback module, **The Frame Type**, **The traceback Module**
transaction processing, **Handling Errors in Large Programs**
Transcrypt, **Transcrypt: Convert your Python to JavaScript**
translate (fnmatch module), **The fnmatch Module**
translate (string method), **Methods of String Objects**
Transmission Control Protocol (TCP), **Networking Basics**
transport layer protocol, **The Berkeley Socket Interface**
transport layer security (TLS), **Transport Layer Security**
triple double quotes ("""), **Strings**
triple single quotes ('''), **Strings**
True, **Data Types**, **Boolean Values**, **The if Statement**, **General-Purpose Special Methods**, **Built-in Types**
true division, **Division**
trunc (math module), **The math and cmath Modules**
truncation, **Precision specification**
truthy, **Boolean Values**
try statement, **The try and raise Statements**, **The try Statement-try/except/finally**
tuple (built-in type), **Built-in Types**
tuples, **Tuples**, **Tuples**, **The time Module**
TurboGears, **A Few Popular Full-Stack Frameworks**
tutors, **Mailing lists**
Two Function Rule, **Adding Type Annotations to New Code**
Two Pizza Rule, **Adding Type Annotations to New Code**
type (built-in type), **Dynamic class definition using the type built-in function**, **Built-in Types**
type annotations

- adding to code, How to Add Type Annotations to Your Code-Using .pyi Stub Files
- history of, History
- syntax for, Type Annotation Syntax-Type Annotation Syntax
- typing module, The typing Module-NewType
- using at runtime, Using Type Annotations at Runtime
- utilities for type checking, Type-Checking Utilities

type checking
- with isinstance, Built-in Types
- utilities for, Type-Checking Utilities

type equality checking, Built-in Types

type hints, Type Annotations

type inferencing, Other Type Checkers, Creating a NumPy array

type objects, Type Objects

type, definition of term, Python Classes

TypeAlias (typing module), Utilities and Decorators, TypeAlias

TypedDict (typing module), Defining Custom Types, TypedDict, Adding Type Annotations to New Code

types, Data Types

TypeVar (typing module), Utilities and Decorators, Generics and TypeVars-Restricting TypeVar to specific types

TYPE_CHECKING (typing module), Utilities and Decorators

typing module
- abstract base classes (ABCs), Abstract Base Classes
- defining custom types, Defining Custom Types-NewType
- protocols, Protocols
- type expression parameters, Type Expression Parameters
- types, Types-Types
- utilities and decorators, Utilities and Decorators-Utilities and Decorators

tzdata, The zoneinfo Module

tzinfo abstract class, The tzinfo Abstract Class

tzname (datetime.time class), The time Class

tzname (time module), The time Module

tzname (tzinfo class), The tzinfo Abstract Class

U

UDP (User Datagram Protocol), Networking Basics

umask (file), Permissions

unbinding, Variables, del Statements, General-Purpose Special Methods

unbound methods, Bound and Unbound Methods-Bound and Unbound Methods

unbounded iterators, Iterators

underscore (_)

- double (__ dunder), Python Classes, Class-private variables, Special Methods
- with gettext module, Using gettext for localization
- in identifiers, Identifiers, Class-private variables
- in interactive interpreter, Identifiers
- module-private variables, Module-private variables
- numeric literals, Underscores in numeric literals
- simple statements, Simple statements
- wildcard pattern, The wildcard pattern

unfinished tasks (in queues), The queue Module

unfreeze (gc module), The gc Module

ungetch (msvcrt module), The msvcrt module

Unicode characters

- avoiding homoglyphs, Identifiers
- BeautifulSoup package and, BeautifulSoup, Unicode, and encoding
- converting bytestrings into, Unicode-The unicodedata Module
- escaped Unicode character (\{name}), Strings
- normalization strategies, Identifiers
- in string literals, Strings
- \U (eight hex digits), Strings
- \u (four hex digits), Strings

unicodedata module, The unicodedata Module

uniform resource identifier (URI), URL Access

Uniform Resource Locator (URL), URL Access

union (set method), Set Methods

unit testing, Testing-Unit Testing and System Testing unittest module, The unittest Module-Unit tests dealing with large amounts of data

Universal Coordinated Time (UTC), The time Module

Unix sockets, The Berkeley Socket Interface

Unix-like platforms, Unix-Like Platforms

unlink (os module), File and directory functions of the os module

unlink (pathlib.Path class), The pathlib Module

unpack function (struct module), The struct Module

unpacking assignment, Plain assignment

unpack_from function (struct module), The struct Module

Unpickler function (pickle module), The pickle Module

untrusted code (eval, exec), Never exec or eval Untrusted Code

update (dict method), Dictionary Methods

updates, Version Conventions, Significant Changes in Python Through 3.11, New Features and Changes in Python 3.7 Through 3.11-Python 3.11

upgrading, Significant Changes in Python Through 3.11-Significant Changes in Python Through 3.11, Planning a Python Version Upgrade-How Often Should You Upgrade?

- breaking changes, New Features and Changes in Python 3.7 Through 3.11

upper (string method), Methods of String Objects

uppercase letters

- in class names, Identifiers
- in identifier names, Identifiers

URI (uniform resource identifier), URL Access

URL (Uniform Resource Locator), URL Access

urljoin (urllib.parse module), The urllib.parse module

urllib module, The urllib Package-The urllib.request module

urlopen (urllib.request module), The urllib.request module

urlretrieve (urllib.request module), The urllib.request module

urlsplit (urllib.parse module), The urllib.parse module

urlunsplit (urllib.parse module), The urllib.parse module

User Datagram Protocol (UDP), Networking Basics

user groups, User groups and organizations

user-coded classes, Formatting of User-Coded Classes

UserWarning class, Classes

UTC (Universal Coordinated Time), The time Module

utcfromtimestamp (datetime.datetime class), The datetime Class

utcnow (datetime.datetime class), The datetime Class

utcoffset (datetime.time class), The time Class

utcoffset (tzinfo class), The tzinfo Abstract Class

utctimetuple (datetime.datetime class), The datetime Class

UTF-8 encoding, Character Sets

utime (os module), File and directory functions of the os module

uu module, The uu Module

Uvicorn, FastAPI

uWSGI, WSGI Servers

V

Value class (multiprocessing module), The Value class

value conversion, Value Conversion

value formatting, Value Formatting: The Format Specifier

value patterns, Value patterns

ValueError, Sequences in General

values (dict method), Dictionary Methods

van Rossum, Guido, Python Development and Versions, History

variable scope, List comprehensions

variables and other references

- assignment statement, Assignment Statements, The copy Module
- class-private variables, Class-private variables
- del statement, del Statements
- free variables, Nested functions and nested scopes
- global variables, Namespaces
- module-private variables, Module-private variables
- variables, Variables

vars (built-in function), Built-in Functions

vectors, List operations

version (sys module), The sys Module

versions, Python Development and Versions

version_info (sys module), The sys Module

vertical bar (|), Command-Line Syntax and Options, Dictionaries

vertical tab (\v), Strings

very high-level language (VHLL), The Python Language

view objects, Dictionary Methods

Vim text editor, Free Text Editors with Python Support

virtual circuits, Client/Server Computing

virtual environments

- benefits of, Other Developments, Implementations, and Distributions, Enter the Virtual Environment
- best practices, Best Practices with Virtualenvs
- creating and deleting, Creating and Deleting Virtual Environments-Creating and Deleting Virtual Environments
- dependency requirements, Managing Dependency Requirements
- other environment solutions, Other Environment Management Solutions
- working with, Working with Virtual Environments

virtual subclasses, The abc module

virtualenv package, Other Environment Management Solutions

W

wait (threading module), Condition objects, Event objects, Barrier objects

Waitress, WSGI Servers

walrus operator (:=)

- assignment expressions, Assignment Expressions
- regular expressions and, REs and the := Operator
- simple statements, Simple statements

warn (warnings module), Functions

warnings module, The warnings Module

weak references, The weakref Module

WeakKeyDictionary (weakref module), The weakref Module

weakref module, The weakref Module-The weakref Module

WeakSet (weakref module), The weakref Module

WeakValueDictionary (weakref module), The weakref Module

web programming (server-side)

- http.server, http.server
- Python web frameworks, Python Web Frameworks-FastAPI
- serving HTTP, Serving HTTP

Web Server Gateway Interface (WSGI), WSGI-ASGI

web scraping, Structured Text: HTML

Web Server Gateway Interface (WSGI), WSGI

web2py, A Few Popular Full-Stack Frameworks

weekday (datetime.date class), The date Class

weekday (datetime.datetime class), The datetime Class

Werkzeug, Considerations When Using Lightweight Frameworks

while statement, := in a while statement, The while Statement

white-box testing, Unit Testing and System Testing

whitespace, Lines and Indentation, Delimiters, Fill and alignment

wildcard patterns, The match Statement

with statement, The with Statement, The with Statement and Context Managers-The with Statement and Context Managers, Creating a File Object with open, Socket Objects

with open, Creating a File Object with open

worker threads, Threaded Program Architecture

workgroups (PSF), Workgroups

wrap (textwrap module), Text Wrapping and Filling

wrapping, Text Wrapping and Filling, The io Module

wraps (functools module), The functools Module

write (file method), Attributes and Methods of File Objects

writelines (file method), Attributes and Methods of File Objects

write_bytes (pathlib.Path class), The pathlib Module

write_history_file (readline module), The readline Module

write_text (pathlib.Path class), The pathlib Module

WSGI (Web Server Gateway Interface), WSGI-ASGI

X

XHTML (Extensible HyperText Markup Language), Structured Text: HTML

XML (eXtensible Markup Language), Structured Text: HTML, Structured Text: XML

XML-RPC (Extensible Markup Language remote procedure call) protocol, Other Network Protocols

xml.etree.ElementTree module, ElementTree

Y

yield from expression, **yield from**

Z

zero normalization (z) (f-strings), **Zero normalization (z)**

zip (built-in function), **Built-in Functions**

ZipFile class (zipfile module), **The ZipFile class**

zipfile module, **The zipfile Module**-**The ZipInfo class**

ZipInfo class (zipfile module), **The ZipInfo class**

zip_longest (itertools module), **The itertools Module**

zoneinfo module, **The zoneinfo Module**