

## Python Glossary

<a href="#">Indentation</a>	Indentation refers to the spaces at the beginning of a code line
<a href="#">Comments</a>	Comments are code lines that will not be executed
<a href="#">Multi Line Comments</a>	How to insert comments on multiple lines
<a href="#">Creating Variables</a>	Variables are containers for storing data values
<a href="#">Variable Names</a>	How to name your variables
<a href="#">Assign Values to Multiple Variables</a>	How to assign values to multiple variables
<a href="#">Output Variables</a>	Use the print statement to output variables
<a href="#">String Concatenation</a>	How to combine strings
<a href="#">Global Variables</a>	Global variables are variables that belongs to the global scope
<a href="#">Built-In Data Types</a>	Python has a set of built-in data types
<a href="#">Getting Data Type</a>	How to get the data type of an object
<a href="#">Setting Data Type</a>	How to set the data type of an object
<a href="#">Numbers</a>	There are three numeric types in Python
<a href="#">Int</a>	The integer number type
<a href="#">Float</a>	The floating number type
<a href="#">Complex</a>	The complex number type
<a href="#">Type Conversion</a>	How to convert from one number type to another
<a href="#">Random Number</a>	How to create a random number
<a href="#">Specify a Variable Type</a>	How to specify a certain data type for a variable
<a href="#">String Literals</a>	How to create string literals
<a href="#">Assigning a String to a Variable</a>	How to assign a string value to a variable
<a href="#">Multiline Strings</a>	How to create a multi line string
<a href="#">Strings are Arrays</a>	Strings in Python are arrays of bytes representing Unicode characters
<a href="#">Slicing a String</a>	How to slice a string
<a href="#">Negative Indexing on a String</a>	How to use negative indexing when accessing a string
<a href="#">String Length</a>	How to get the length of a string
<a href="#">Check In String</a>	How to check if a string contains a specified phrase
<a href="#">Format String</a>	How to combine two strings
<a href="#">Escape Characters</a>	How to use escape characters
<a href="#">Boolean Values</a>	True or False
<a href="#">Evaluate Booleans</a>	Evaluate a value or statement and return either True or False
<a href="#">Return Boolean Value</a>	Functions that return a Boolean value
<a href="#">Operators</a>	Use operator to perform operations in Python
<a href="#">Arithmetic Operators</a>	Arithmetic operator are used to perform common mathematical operations
<a href="#">Assignment Operators</a>	Assignment operators are use to assign values to variables
<a href="#">Comparison Operators</a>	Comparison operators are used to compare two values
<a href="#">Logical Operators</a>	Logical operators are used to combine conditional statements
<a href="#">Identity Operators</a>	Identity operators are used to see if two objects are in fact the same object
<a href="#">Membership Operators</a>	Membership operators are used to test is a sequence is present in an object
<a href="#">Bitwise Operators</a>	Bitwise operators are used to compare (binary) numbers
<a href="#">Lists</a>	A list is an ordered, and changeable, collection
<a href="#">Access List Items</a>	How to access items in a list
<a href="#">Change List Item</a>	How to change the value of a list item
<a href="#">Loop Through List Items</a>	How to loop through the items in a list
<a href="#">List Comprehension</a>	How use a list comprehensive
<a href="#">Check if List Item Exists</a>	How to check if a specified item is present in a list

<a href="#">List Length</a>	How to determine the length of a list
<a href="#">Add List Items</a>	How to add items to a list
<a href="#">Remove List Items</a>	How to remove list items
<a href="#">Copy a List</a>	How to copy a list
<a href="#">Join Two Lists</a>	How to join two lists
<a href="#">Tuple</a>	A tuple is an ordered, and unchangeable, collection
<a href="#">Access Tuple Items</a>	How to access items in a tuple
<a href="#">Change Tuple Item</a>	How to change the value of a tuple item
<a href="#">Loop List Items</a>	How to loop through the items in a tuple
<a href="#">Check if Tuple Item Exists</a>	How to check if a specified item is present in a tuple
<a href="#">Tuple Length</a>	How to determine the length of a tuple
<a href="#">Tuple With One Item</a>	How to create a tuple with only one item
<a href="#">Remove Tuple Items</a>	How to remove tuple items
<a href="#">Join Two Tuples</a>	How to join two tuples
<a href="#">Set</a>	A set is an unordered, and unchangeable, collection
<a href="#">Access Set Items</a>	How to access items in a set
<a href="#">Add Set Items</a>	How to add items to a set
<a href="#">Loop Set Items</a>	How to loop through the items in a set
<a href="#">Check if Set Item Exists</a>	How to check if a item exists
<a href="#">Set Length</a>	How to determine the length of a set
<a href="#">Remove Set Items</a>	How to remove set items
<a href="#">Join Two Sets</a>	How to join two sets
<a href="#">Dictionary</a>	A dictionary is an unordered, and changeable, collection
<a href="#">Access Dictionary Items</a>	How to access items in a dictionary
<a href="#">Change Dictionary Item</a>	How to change the value of a dictionary item
<a href="#">Loop Dictionary Items</a>	How to loop through the items in a tuple
<a href="#">Check if Dictionary Item Exists</a>	How to check if a specified item is present in a dictionary
<a href="#">Dictionary Length</a>	How to determine the length of a dictionary
<a href="#">Add Dictionary Item</a>	How to add an item to a dictionary
<a href="#">Remove Dictionary Items</a>	How to remove dictionary items
<a href="#">Copy Dictionary</a>	How to copy a dictionary
<a href="#">Nested Dictionaries</a>	A dictionary within a dictionary
<a href="#">If Statement</a>	How to write an if statement
<a href="#">If Indentation</a>	If statemnts in Python relies on indentation (whitespace at the beginning of a line)
<a href="#">Elif</a>	elif is the same as "else if" in other programming languages
<a href="#">Else</a>	How to write an if...else statement
<a href="#">Shorthand If</a>	How to write an if statement in one line
<a href="#">Shorthand If Else</a>	How to write an if...else statement in one line
<a href="#">If AND</a>	Use the and keyword to combine if statements
<a href="#">If OR</a>	Use the or keyword to combine if statements
<a href="#">Nested If</a>	How to write an if statement inside an if statement
<a href="#">The pass Keyword in If</a>	Use the pass keyword inside empty if statements
<a href="#">While</a>	How to write a while loop
<a href="#">While Break</a>	How to break a while loop
<a href="#">While Continue</a>	How to stop the current iteration and continue wit the next
<a href="#">While Else</a>	How to use an else statement in a while loop
<a href="#">For</a>	How to write a for loop
<a href="#">Loop Through a String</a>	How to loop through a string
<a href="#">For Break</a>	How to break a for loop

<a href="#">For Continue</a>	How to stop the current iteration and continue with the next
<a href="#">Looping Through a range</a>	How to loop through a range of values
<a href="#">For Else</a>	How to use an else statement in a for loop
<a href="#">Nested Loops</a>	How to write a loop inside a loop
<a href="#">For pass</a>	Use the pass keyword inside empty for loops
<a href="#">Function</a>	How to create a function in Python
<a href="#">Call a Function</a>	How to call a function in Python
<a href="#">Function Arguments</a>	How to use arguments in a function
<a href="#">*args</a>	To deal with an unknown number of arguments in a function, use the * symbol before the parameter name
<a href="#">Keyword Arguments</a>	How to use keyword arguments in a function
<a href="#">**kwargs</a>	To deal with an unknown number of keyword arguments in a function, use the * symbol before the parameter name
<a href="#">Default Parameter Value</a>	How to use a default parameter value
<a href="#">Passing a List as an Argument</a>	How to pass a list as an argument
<a href="#">Function Return Value</a>	How to return a value from a function
<a href="#">The pass Statement in Functions</a>	Use the pass statement in empty functions
<a href="#">Function Recursion</a>	Functions that can call itself are called recursive functions
<a href="#">Lambda Function</a>	How to create anonymous functions in Python
<a href="#">Why Use Lambda Functions</a>	Learn when to use a lambda function or not
<a href="#">Array</a>	Lists can be used as Arrays
<a href="#">What is an Array</a>	Arrays are variables that can hold more than one value
<a href="#">Access Arrays</a>	How to access array items
<a href="#">Array Length</a>	How to get the length of an array
<a href="#">Looping Array Elements</a>	How to loop through array elements
<a href="#">Add Array Element</a>	How to add elements from an array
<a href="#">Remove Array Element</a>	How to remove elements from an array
<a href="#">Array Methods</a>	Python has a set of Array/Lists methods
<a href="#">Class</a>	A class is like an object constructor
<a href="#">Create Class</a>	How to create a class
<a href="#">The Class __init__() Function</a>	The __init__() function is executed when the class is initiated
<a href="#">Object Methods</a>	Methods in objects are functions that belong to the object
<a href="#">self</a>	The self parameter refers to the current instance of the class
<a href="#">Modify Object Properties</a>	How to modify properties of an object
<a href="#">Delete Object Properties</a>	How to modify properties of an object
<a href="#">Delete Object</a>	How to delete an object
<a href="#">Class pass Statement</a>	Use the pass statement in empty classes
<a href="#">Create Parent Class</a>	How to create a parent class
<a href="#">Create Child Class</a>	How to create a child class
<a href="#">Create the __init__() Function</a>	How to create the __init__() function
<a href="#">super Function</a>	The super() function makes the child class inherit the parent class
<a href="#">Add Class Properties</a>	How to add a property to a class
<a href="#">Add Class Methods</a>	How to add a method to a class
<a href="#">Iterators</a>	An iterator is an object that contains a countable number of values
<a href="#">Iterator vs Iterable</a>	What is the difference between an iterator and an iterable
<a href="#">Loop Through an Iterator</a>	How to loop through the elements of an iterator
<a href="#">Create an Iterator</a>	How to create an iterator
<a href="#">Stop Iteration</a>	How to stop an iterator
<a href="#">Global Scope</a>	When does a variable belong to the global scope?
<a href="#">Global Keyword</a>	The global keyword makes the variable global

<a href="#">Create a Module</a>	How to create a module
<a href="#">Variables in Modules</a>	How to use variables in a module
<a href="#">Renaming a Module</a>	How to rename a module
<a href="#">Built-in Modules</a>	How to import built-in modules
<a href="#">Using the dir() Function</a>	List all variable names and function names in a module
<a href="#">Import From Module</a>	How to import only parts from a module
<a href="#">Datetime Module</a>	How to work with dates in Python
<a href="#">Date Output</a>	How to output a date
<a href="#">Create a Date Object</a>	How to create a date object
<a href="#">The strftime Method</a>	How to format a date object into a readable string
<a href="#">Date Format Codes</a>	The datetime module has a set of legal format codes
<a href="#">JSON</a>	How to work with JSON in Python
<a href="#">Parse JSON</a>	How to parse JSON code in Python
<a href="#">Convert into JSON</a>	How to convert a Python object in to JSON
<a href="#">Format JSON</a>	How to format JSON output with indentations and line breaks
<a href="#">Sort JSON</a>	How to sort JSON
<a href="#">RegEx Module</a>	How to import the regex module
<a href="#">RegEx Functions</a>	The re module has a set of functions
<a href="#">Metacharacters in RegEx</a>	Metacharacters are characters with a special meaning
<a href="#">RegEx Special Sequences</a>	A backslash followed by a a character has a special meaning
<a href="#">RegEx Sets</a>	A set is a set of characters inside a pair of square brackets with a special meaning
<a href="#">RegEx Match Object</a>	The Match Object is an object containing information about the search and the result
<a href="#">Install PIP</a>	How to install PIP
<a href="#">PIP Packages</a>	How to download and install a package with PIP
<a href="#">PIP Remove Package</a>	How to remove a package with PIP
<a href="#">Error Handling</a>	How to handle errors in Python
<a href="#">Handle Many Exceptions</a>	How to handle more than one exception
<a href="#">Try Else</a>	How to use the else keyword in a try statement
<a href="#">Try Finally</a>	How to use the finally keyword in a try statement
<a href="#">raise</a>	How to raise an exception in Python

ArithmeticError	Raised when an error occurs in numeric calculations
AssertionError	Raised when an assert statement fails
AttributeError	Raised when attribute reference or assignment fails
Exception	Base class for all exceptions
EOFError	Raised when the input() method hits an "end of file" condition (EOF)
FloatingPointError	Raised when a floating point calculation fails
GeneratorExit	Raised when a generator is closed (with the close() method)
ImportError	Raised when an imported module does not exist
IndentationError	Raised when indentation is not correct
IndexError	Raised when an index of a sequence does not exist
KeyError	Raised when a key does not exist in a dictionary
KeyboardInterrupt	Raised when the user presses Ctrl+c, Ctrl+z or Delete
LookupError	Raised when errors raised cant be found
MemoryError	Raised when a program runs out of memory
NameError	Raised when a variable does not exist
NotImplementedError	Raised when an abstract method requires an inherited class to override the method
OSError	Raised when a system related operation causes an error
OverflowError	Raised when the result of a numeric calculation is too large
ReferenceError	Raised when a weak reference object does not exist
RuntimeError	Raised when an error occurs that do not belong to any specific expections
StopIteration	Raised when the next() method of an iterator has no further values
SyntaxError	Raised when a syntax error occurs
TabError	Raised when indentation consists of tabs or spaces
SystemError	Raised when a system error occurs
SystemExit	Raised when the sys.exit() function is called
TypeError	Raised when two different types are combined
UnboundLocalError	Raised when a local variable is referenced before assignment
UnicodeError	Raised when a unicode problem occurs
UnicodeEncodeError	Raised when a unicode encoding problem occurs
UnicodeDecodeError	Raised when a unicode decoding problem occurs
UnicodeTranslateError	Raised when a unicode translation problem occurs
ValueError	Raised when there is a wrong value in a specified data type
ZeroDivisionError	Raised when the second operator in a division is zero

Font: [https://www.w3schools.com/python/python\\_reference.asp](https://www.w3schools.com/python/python_reference.asp)