

Python Programming Language

- Python is a programming language.
- Python can be used on a server to create web applications.
- It was created in 1991 by Guido van Rossum.
- Python is used for: server side web development, software development, mathematics, system scripting.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping or for production ready software development.
- Python works on different platforms (Windows, Mac OS X, Linux, Raspberry PI).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write a program with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that the prototyping can be very quick.
- Python can be treated in a procedure way, an object-oriented way or a functional way.
- It is possible to write python in an integrated development environment such as Thonny, PyCharm, Netbeans or Eclipse, which are particularly useful when managing large collections of Python files.
- Python was designed to be for readability, and has some similarities to the English Language, with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons, or parentheses.
- Python relies on indentations, using white space, to define scope, of loops, functions, and classes. Other programming languages often use curly brackets for this purpose.
- Python is an interpreted programming language, this means that as a developer, you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.
- To test a short amount of code in python sometimes it is the quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.
- Whenever you are done in the python command line you can simply type the (exit()) function to quit the python command line interface.
- Python syntax can be executed by writing directly in the command line.
- In Python indentation is very important. Python uses indentation to indicate a block of code. Python will give an error if you skip the indentation.
- Python has commenting capability for the purpose of in-code documentation.
- Comment starts with a (#) Hash Symbol, and Python will render the rest of the line as a comment.
- Python also has extended documentation capability called doc-strings.
- Docstrings can be of one line or multiple lines.
- Python uses triple quotes at the beginning and at the ending of the doc-strings.
- Unlike other programming languages, Python has no command for declaring a variable.

Python Programming Language

- A variable is created at the moment you first assign a value to it.
- Variables do not need to be declared with any particular type and can even change type after they have been set.
- A variable can have a short name (like x or y) or a more descriptive name (age, carname, total_name).
- Rules for Python variables are:
 - A variable name must start with a letter or an underscore character.
 - A variable name cannot start with a number.
 - A variable name can only contain alpha-numeric characters and underscores, (A-Z, 0-9, _).
 - Variable name are case-sensitive (age, Age and AGE) are three different variables.
 - Remember that variables are case-sensitive.
 - The python (print) statement is often used to output variables.
 - To combine both text and variable, python uses (+) plus character.
 - For numbers, The Plus character works as mathematical operator.
 - If you try to combine a string and a number, python will give you an error.
 - There are three numeric types in Python.
 - INT, FLOAT, COMPLEX.
 - Variables of numeric types are created when you assign a value to them.
 - To verify the type of an object in python, use the (type()) function.
 - INT or Integer is a whole number positive or negative without decimals of unlimited length.
 - Float or Floating Point Number is a number positive or negative containing one or more decimals.
 - Float can also be scientific numbers with an (e) to indicate the power of (10).
 - Complex numbers are written with a “j” (Small Jey Alphabet) Character. As the imagery part.
 - There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-oriented language and as such it uses classed to define a data type. Including it primitive types.
 - Casting in python is therefore done using constructor function.
 - INT: Constructs and integer number from an integer literal or from a float literal by rounding down to the previous whole number, or from a string literal providing the string represents a whole number.
 - FLOAT: Construct a float number from integer literal, a float literal or a string literal, providing the string represents a float or an integer.
 - STR: Constructs a string from a wide variety of types, including strings, integer literals and float literals.
 - String literals in Python are surrounded by either single quotation marks, or double quotations mark.
 - Strings can be output on screens using the (print()) function.
 - Like many other programming languages. Strings in Python are arrays of bytes representing Unicode of characters. However, Python does not have a character datatype. A single character is simply a string with a length of one. Square brackets can be used to access elements of the string.

Python Programming Language

- Python allows for command line input. That means we are able to ask the user for input.
- Operators are used to perform operations on variables and values.
- Python divides the operators in the following groups:
- Arithmetic operators: are used on numeric values to perform common mathematical operations.
- Assignment Operators: are used to assign values to variables.
- Comparison Operators: are used to compare two values.
- Logical Operators: are used to combine conditional statements.
- Identity Operators: are used to compare objects, not if they are equal, but actually they are the same object with the same memory location.
- Membership Operators: are used to test if the sequence is presented in an object.
- Bit-wise Operators: are used to compare binary numbers.
- There are four collections data type in the python programming language.
- List: is the collection which is ordered and changeable. Allows duplicate members.
- Tuple: is a collection which is ordered and unchangeable. Allows duplicate members.
- Set: is a collection which is unordered and UN-indexed. No duplicate members.
- Dictionary is a collection which is unordered, changeable and indexed. No duplicate members.
- When choosing a collection type it is useful to understand the properties of that type. Choosing the right type for the particular data set could mean retention of meaning and it could mean an increase in efficiency and security.
- In Python, lists are written with square brackets.
- You can access the lists items by referring to the index number.
- To change the value of specific item refer to the index number.
- You can loop through the list items by using a for loop.
- Python has a set of built-in methods that you can use on lists.
- It is also possible to use the (list()) constructor to make a list.
- A tuple is a collection which is ordered and unchangeable. In python's tuples are written with round brackets.
- You can access tuples items by referring to the index number, inside square brackets.
- Once tuples are created, you cannot change its values. Tuples are unchangeable.
- You cannot remove items in a tuple.
- Tuples are unchangeable, so you cannot remove items from it. But you can delete the tuples completely.
- It is also possible to use the (tuple()) constructor to make a constructor.
- A set is a collection which is unordered and UN-indexed. In python sets are written with curly brackets.
- You cannot access the items in a set by referring to an index, since set are unordered the items has no index. But you can loop through the set item using the (FOR) loop, or asked (IF) specified value is present in a set by using the (IN) keyword.
- Once a set is created, you cannot change its items but you can add new items.
- To add one item to a set use the (add()) method. To add more than one item to a set use the (update()) method.

Python Programming Language

- A dictionary is a collection which is unordered, changeable and indexed. In Python, dictionaries are written with curly brackets, and they have keys and values. You can access the items of the dictionaries by referring to its key name inside square brackets. There is also a method which returns the same result.
- You can loop through a dictionary using (for) loop. When looping through a dictionary the return values are the keys of the dictionary but there are methods to return the values as well.
- You can also use the method (values()) to return values of a dictionary.
- Loop through both keys and values by using the items function.
- To check if specified key is present in a dictionary use the (in) keyword.
- To determine how many items (key-value pairs) a dictionary has use the (len()) method.
- Adding an item to the dictionary is done by using a new index key and assigning a value to it.
- There are several methods to remove items from a dictionary.
- The (pop()) method, removes the item with the specified key name.
- The (popitem()) method removes the last item. In versions prior than version (3.7) a random item is removed instead.
- The (del) keyword removes the item with the specified key name. The (del) keyword can also delete the dictionary completely.
- The (clear()) method empties the dictionary.
- It is also possible to use the (dict()) constructor to make a dictionary.
- Python has a set of built-in methods that you can use on dictionaries.
- Python supports the usual logical conditions from mathematics.
- Equal ($a == b$).
- Not Equal ($A != b$).
- Less than ($a < b$).
- Less than or equal ($a <= b$).
- Greater than ($a > b$).
- Greater than or equal to ($a >= b$).
- These conditions can be used in several ways. Most commonly in “if statements” and “loops”.
- An if statement is written by using the (if) keyword.
- Python relies on indentation, using white-space, to define scope in the code. Other programming languages often use curly brackets for this purpose.
- If statement, without indentation will raise an error.
- The (elif) keyword is used as for else if statement.
- The (else) keyword catches everything which isn't caught by the preceding condition.
- You can also have an (else) without the (elif).
- The (and) keyword is a logical operator and is used to combine conditional statements.
- The (or) keyword is the logical operator, and is used to combine conditional statements.
- Python has two primitive loop:
- While Loops.
- For Loops.
- With the while loop we can execute the set of statements as long as condition is true.
- Remember to increment the loop variable or else loop will continue forever.

Python Programming Language

- The while loop requires relevant variables to be ready.
- With the (break) statement, we can stop the loop even if the while loop is true.
- With the (continue) statement we can stop the current iteration and continue with the next.
- A (for) loop is used for iterating a sequence that is either a list, a tuple, a dictionary, a set, or a string. This is less like the (for) keyword in other programming languages and works more like a iterator method as found in other object oriented programming languages. With the (for) loop we can execute a set of statements once for each item in a list, tuple, sets etc. The (for) loop does not require an indexing variable to set before hand. Even strings are iterate-able objects they contain a sequence of characters.
- With the (break) statement we can stop the loop before it has looped through all the items.
- With the (continue) statement we can stop the current iteration of the loop and continue with the next statement.
- To loop through the set of code a specified number of times we can use the (range()) function
- The (range()) function returns the sequence of numbers starting from (0) by default and increment by (1) by default and ends at a specified number.
- The (range()) function defaults to (0) as a starting value, however it is possible to specify the starting value by adding a parameter like (range(2,6)) which means values from (2 to 6) but not including (6).
- The (range()) function defaults to increment the sequence by (1) however it is possible to specify the increment value by adding the third parameter like (range(2, 30, 3)).
- The (else) keyword in a (for) loop specifies a block of code to be executed when the loop returns (FALSE) value or have finished execution.
- Nested loop is a loop which is inside in a loop. The inner loop will be executed one time for each iteration of the outer loop.
- A function is a block of code which only executes when it is called. You can pass data, known as parameters, into a function. A function can return data as a result. In Python, function is defined using the keyword (def). Abbreviate as definition or define. To call a function use the function name followed by parentheses. Information can be passed to functions as parameter. Parameters are specified after the function name, inside the parentheses. You can add as many parameters as you like, just separate them with comma. When the function is called, the given parameter will be used inside the function. Function parameters can also have default values. If we cannot provide the value explicitly, the default value of the parameter will be used instead. To let a function return a value use the (return) statement.
- Python also accepts function recursion, which means a defined function can call itself. Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result. The developer should be very careful with recursion, as it can be quite easy to slip into writing a function which never terminates, or one that uses excess amount of memory or processor power. However, when writing correctly recursion can be very efficient and mathematically-elegant approach to programming.
- A lambda function is a small anonymous function. A lambda function can take any number of arguments, but can only have one expression. The expression is executed and the result is

Python Programming Language

returned. Lambda functions can take any number of arguments. The power of lambda is shown when you use them as an anonymous function inside another function. Use lambda function when an anonymous function is required for a short period of time.

- Python does not have built-in support for Arrays, but python list can be used instead.
- Arrays are used to store multiple values in one single variable. An array is a variable which can hold one value at a time. If you have a list of items, storing the list items in single variable can be very costly. An array can hold many values under a single name and you can access the values by referring to an index number. You refer to an array element by referring to an index number. The length of an array is always one more than the highest array index. You can use the (for in) loop to loop through all elements of an array. You can use the (append()) method to add an element to an array. You can use the (pop()) method to remove an element from an array. (pop()) method use index number of an array as parameter to the function to remove the value. You can also use the remove method to remove an element from the array. (remove()) method accepts value of an array as an argument to the function to remove the current element with the same value. The (remove()) method only removes the first occurrence of the specified value. Python has a set of built-in methods that you can use on lists/ arrays.
- Python does not have built-in support for Arrays, but Python lists can be used instead.
- An iterator is an object that contains a countable number of values. An iterator is an object that can be iterated upon meaning that you can transverse through all the values. Technically in Python, an iterator is an object which implements the iterator protocol which consist of the methods (`__iter__()` and `__next__()`).
- Lists, tuples, dictionaries and sets are all iterator-able objects. They are iterator-able containers from which you can get an iterator. All these objects have a (`iter()`) method which is used to get an iterator. Even strings are iterator-able objects and can return an iterator. We can also use (for) loop to iterate through an iterator-able object. The (for) loop actually creates an iterator object and executes the next method for each increment of the loop.
- To create an object of class as an iterator you have to implement the methods (`__iter__()` and `__next__()`) to your custom class.
- All the classes have the function called (`__init__()`) which allow you to do some initializing when the object is being created.
- The (`__iter__()`) method acts the same, you can perform operations, but must always return the iterator method itself.
- The (`__next__()`) method allows you to do operations, and must return the next item in the sequence. To prevent the iteration to go on forever, we can use the (StopIteration) statement.
- Consider a module to be the same as a code library. A file containing a set of functions you want to include in your application. To create a module, just save the code you want in a file with the file extension (*.py). We can use the freshly created module by using (import) statement. When using a function from a module use like this: (module_name.function_name). You can name a module file whatever you like but it must have the file extension (*.py). You can create and alias when you import a module by using the (as) keyword. The module can contain variables too of all type. There are several built-in modules in Python, which you can import whenever you like. There is a built-in function to list all the functions name or variables names in a

Python Programming Language

module called (`dir()`) function. The (`dir()`) function can be used on all modules also the ones you created yourself. You can choose to import only parts from the module by using the (`from`) keyword. When importing using (`from`) keyword, do not use the module name when referring to elements in the module.

- A date in Python is not a date type of its own, but we can import an module named (`datetime`) to work with dates as date objects. The date contains year, month, day, hour, minutes, second, and microseconds. The `datetime` module has many methods to return information about the date object. To create a date, we can use the (`datetime`) class constructor of `datetime` module. The `datetime` class requires three parameters to create a date: year, month and day. The `datetime` class constructor also accepts parameter for timezone and time. Hour, Minutes, Seconds, Microseconds, Tzone; but they are optional and has a default value 0. None for Time Zone. The `datetime` object has a method for formatting date object into readable strings. The method is called (`strftime()`) and accepts one parameter, (`format`), to specify the format of the returned string.
- JSON is a syntax for storing and exchanging data. JSON is text, written with JavaScript Object Notation.
- Python has a built-in package called (`json`) which can be used to work with JSON data.
- If you have a JSON string you can parse it by using (`json.loads()`) method. The result will be a python dictionary.
- If you have a Python object, you can convert it into JSON string by using (`json.dumps()`) method. You can convert Python objects of the following types into JSON string:
 - Dictionaries, Lists, Tuples, string, int, float, true, false and none.
 - When you convert from Python to JSON, Python objects are converted into JavaScript equivalent.
- The (`json.dumps()`) method has parameters to make it easier to read the result. Use the (`indent`) parameter to define the numbers of indents. You can also define the separators, default value is ("`,`", "") which means using a comma and a space to separate each object and a colon and a space to separate keys from values. The (`json.dumps()`) method has parameters to order the keys in the result. Use the (`sort_keys`) parameter to specify if the result should be sorted or not.
- A Regex or Regular Expression, is a sequence of characters that forms a search pattern. Regex can be used to check if the string contains the specified search pattern. Python has a built-in package called (`re`), which can be used to work with Regular Expression. The (`re`) module offers a set of functions that allows us to search a string for a match. Meta-characters are the characters, with a special meaning. A special sequence is a (`\`) followed by one of the characters from the list in documentation, and has a special meaning. A set is a set of characters inside a pair of square brackets. With a special meaning. A Match object is an object containing the information about the search and the result. If there are no match, the value `None` will be returned instead of the Match Object. The Match object has properties and methods used to retrieve the information about the search and the result.
- PIP is a package manager for Python Packages, or modules if you like.
- If you have Python version (3.4) or later, PIP is included by default.
- A package contains all the files you need for a module.

Python Programming Language

- Modules are Python Code Libraries that you can include in your project.
- Downloading a package is very easy. Open the command line interface and instruct PIP to download the package you want.
- Once a package is install, it is ready to use.
- Import the downloaded package in your project.
- Use the (uninstall) command to remove a package. The PIP package manager will ask you to confirm that you want to remove the <desired> package. Press (y) key and the package will be removed.
- Use the (list) command to list all the packages installed on your system.
- The (try) block lets you test a block of code for errors.
- The (except) block lets you handle the error.
- The (finally) block lets you execute code, regardless of the result of the (try) and (except) blocks.
- When an error occurs, or exception as we call it, Python will generally stops and generate an error message. These exceptions can be handled, using the (try) statement. Since the (try) block raises an error then the (except) block will executed. Without the (try) block, the program will crash and raise an error. You can define many exception blocks as you want, if you want to execute a special block of code for a special kind of error.
- You can also use the (else) keyword to define a block of code to be executed if no errors were raised.
- The (finally) block, if specified, will be executed regardless if the (try) block raises an error or not. This can be useful to close objects and clean up resources. The program can continue without leaving the object open.
- File handling is an important part of any computer application.
- Python has several functions for, creating, reading, updating and deleting files. Commonly known as (CURD) operations.
- The key function for working with files in Python is the (open()) function. The (open()) function accepts two arguments, (filename and mode). There are four (4) different methods (modes) for opening a file.
- Mode (r): Read, default value. Opens a file for reading, error if the file does not exist.
- Mode (a): Appends, opens a file for appending, creates the file if it does not exist.
- Mode (w): Write, Opens a file for writing, creates the file if it does not exist.
- Mode (x): Create, creates the specified file, returns an error if the file exist.
- Mode (t): Text, default value. Text mode.
- Mode (b): Binary, Binary mode. For example: images.
- To open a file for reading it is enough to specify the name of the file.
- The (open()) function returns a file object, which has a (read()) method for reading the content of the file. By default the (read()) method returns the whole text, but you can also specify how many characters you want to return.
- You can return one line by using the (readline()) method. By looping through the lines of the file, you can read the whole file, line by line.

Python Programming Language

- To write to an existing file, you must add a parameter to the (open()) function. ("a") appends to the end of the file. ("w") writes, to the file, will overwrite any existing content.
- The ("w") option will overwrite the file.
- To create a new file in Python, use the (open()) method with one of the following modes.
- ("x") - Create, will create a file, returns an error if the file exist.
- ("a") - Append, will create a file, if the specified file does not exist.
- ("w") - Write, will create a file, if the specified file does not exist.
- To delete a file, you must import the (os) module and run its (os.remove()) method.
- To avoid getting an error, you might want to check if the file exist before you try to delete it.
- To delete an entire folder, use the (os.rmdir()) method. You can remove empty folder.
- Python can be used in database applications.
- One of the most popular databases is MySQL.
- Python needs a MySQL driver to access the MySQL Database.
- You use PIP to install "MySQL Connector". PIP is most already likely installed in your Python Environment.
-

Python Programming Language

Python Classes and Objects:

- Python is an object oriented programming language.
- Almost everything in Python is an object, with its properties and methods.
- A class is like an object constructor, or a blue print for creating objects.
- We usually use the class name to create an object.
- All classes have a function called (`__init__()`), which is always executed when the class is being initiated.
- Use the (`__init__()`) function to assign values to object properties or other operations that are necessary to do when the object is being created.
- The (`__init__()`) function is called automatically every time the class is being used to create a new object.
- Objects can also contain methods, methods in objects are functions that belongs to the object.
- The (`self`) parameter is a reference to the class reference itself, like (`this`) keyword in other programming languages. It is used to access variables that belongs to the class. It does not have to be named (`self`) you can call it whatever you like, but it has to be the first parameter of any function in the class.
- You can modify properties on objects too.
- You can also delete properties on objects by using (`del`) keyword.
- You can delete the object by using the (`del`) keyword.

Python Programming Language

Class:

A class is a collection of functions and variables. It defines the behavior and the data of that type.

Constructor:

A constructor is a special method of a class, that is used for the initialization of an object for that class. Constructors can also be used for the declaration of the values for the class members variables. Constructors have no return type. Constructors also have same name as the class name. A class can have a different set of constructors for the deployment of variety of object of the same class, with the difference of parameters.

Object:

Object is the physical existence of a class.

Inheritance:

Inheritance is a feature of Object Oriented Programming that allows code reusability when a class includes property of another class.

Abstraction:

Abstraction is the process of accessing only what is needed. Detailed implementation and the complexity is hidden from the world. The concept of abstraction focuses on what the object does.

Encapsulation:

Encapsulation is the idea of hiding the unnecessary details from the outside world. Access modifiers are used for that purpose.

Polymorphism:

In general Polymorphism means “have many shapes”. Polymorphism in Object Oriented Programming focuses on modifying and redefining the implementation of a function. Differ by the terms method or constructor overloading and method or constructor overriding.

Python Programming Language

The Python Tutorials: <https://docs.python.org/3/tutorial/index.html>

Python Officials: <https://www.python.org/>

Python File Operations: https://www.w3schools.com/python/python_file_handling.asp

Python Database Operations: https://www.w3schools.com/python/python_mongodb_getstarted.asp

Python References: https://www.w3schools.com/python/python_reference.asp

Python Info: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

Continue From: https://www.w3schools.com/python/python_mysql_getstarted.asp

Python Programming