

How do you write comments in python?

Ans: Comments in Python start with a # character. However, alternatively at times, commenting is done using docstrings(strings enclosed within triple quotes).**Example:**

```
1<span data-mce-type="bookmark" style="display: inline-block; width: 0px; overflow: hidden; lin
2<pre><span>#Comments in Python start like this
3print("Comments in Python start with a #")
```

How will you capitalize the first letter of string?

In Python, the `capitalize()` method capitalizes the first letter of a string. If the string already consists of a capital letter at the beginning, then, it returns the original string

What is the purpose of 'is', 'not' and 'in' operators?

Ans: Operators are special functions. They take one or more values and produce a corresponding result.

is: returns true when 2 operands are true (Example: "a" is 'a')

not: returns the inverse of the boolean value

in: checks if some element is present in some sequence

What is the usage of help() and dir() function in Python?

Ans: Help() and dir() both functions are accessible from the Python interpreter and used for viewing a consolidated dump of built-in functions.

1. **Help() function:** The help() function is used to display the documentation string and also facilitates you to see the help related to modules, keywords, attributes, etc.
2. **Dir() function:** The dir() function is used to display the defined symbols.

Explain split(), sub(), subn() methods of "re" module in Python.

Ans: To modify the strings, Python's "re" module is providing 3 methods. They are:

- **split()** – uses a regex pattern to "split" a given string into a list.
- **sub()** – finds all substrings where the regex pattern matches and then replace them with a different string

- **subn()** – it is similar to sub() and also returns the new string along with the no. of replacements

Whenever Python exits, why isn't all the memory de-allocated?

Ans:

1. Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces are not always de-allocated or freed.
2. It is impossible to de-allocate those portions of memory that are reserved by the C library.
3. On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object

How can the ternary operators be used in python?

Ans: The Ternary operator is the operator that is used to show the conditional statements. This consists of the true or false values with a statement that has to be evaluated for it.

Syntax:

The Ternary operator will be given as:
`[on_true] if [expression] else [on_false]`
 x, y = 25, 50
 big = x if x < y else y

Example:

The expression gets evaluated like if x<y else y, in this case if x<y is true then the value is returned as big=x and if it is incorrect then big=y will be sent as a result.

What does this mean: *args, **kwargs? And why would we use it?

Ans: We use *args when we aren't sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. **kwargs is used when we don't know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The identifiers args and kwargs are a convention, you could also use *bob and **billy but that would not be wise.

What are negative indexes and why are they used?

Ans: The sequences in Python are indexed and it consists of the positive as well as negative numbers. The numbers that are positive uses '0' that is uses as first index and '1' as the second index and the process goes on like that.

The index for the negative number starts from '-1' that represents the last index in the sequence and '-2' as the penultimate index and the sequence carries forward like the positive number.

The negative index is used to remove any new-line spaces from the string and allow the string to except the last character that is given as `S[:-1]`. The negative index is also used to show the index to represent the string in correct order.

Shallow Copy	Deep Copy
Shallow Copy stores the references of objects to the original memory address.	Deep copy stores copies of the object's value.
Shallow Copy reflects changes made to the new/copied object in the original object.	Deep copy doesn't reflect changes made to the new/copied object in the original object.
Shallow Copy stores the copy of the original object and points the references to the objects.	Deep copy stores the copy of the original object and recursively copies the objects as well.
A shallow copy is faster.	Deep copy is comparatively slower.

What is the difference between a Mutable datatype and an Immutable data type?

Mutable data types can be edited i.e., they can change at runtime. Eg – List, Dictionary, etc.

Immutable data types can not be edited i.e., they can not change at runtime. Eg – String, Tuple, etc.

How are arguments passed by value or by reference in Python?

Everything in Python is an object and all variables hold references to the objects. The reference values are according to the functions; as a result, you cannot change the value of the references. However, you can change the objects if it is mutable.

What is the difference between / and // in Python?

// represents floor division whereas / represents precise division. For Example:

$5//2 = 2$

$5/2 = 2.5$

How is Exceptional handling done in Python?

There are 3 main keywords i.e. try, except, and finally which are used to catch exceptions and handle the recovering mechanism accordingly. Try is the block of a code that is monitored for errors. Except block gets executed when an error occurs.

The beauty of the final block is to execute the code after trying for an error. This block gets executed irrespective of whether an error occurred or not. Finally, block is used to do the required cleanup activities of objects/variables.

Can we Pass a function as an argument in Python?

Yes, Several arguments can be passed to a function, including objects, variables (of the same or distinct data types), and functions. Functions can be passed as parameters to other functions because they are objects. Higher-order functions are functions that can take other functions as arguments

Difference between for loop and while loop in Python

The “for” Loop is generally used to iterate through the elements of various collection types such as [List](#), [Tuple](#), [Set](#), and [Dictionary](#). Developers use a “for” loop where they have both the conditions start and the end. Whereas, the “while” loop is the actual looping feature that is used in any other programming language. Programmers use a Python while loop where they just have the end conditions.

How do you floor a number in Python?

The Python math module includes a method that can be used to calculate the floor of a number.

- [floor\(\)](#) method in Python returns the floor of x i.e., the largest integer not greater than x.
- Also, The method `ceil(x)` in Python returns a ceiling value of x i.e., the smallest integer greater than or equal to x.

What is PIP?

PIP is an acronym for Python Installer Package which provides a seamless interface to install various Python modules. It is a command-line tool that can search for packages over the internet and install them without any user interaction.

What is a zip function?

Python `zip()` function returns a zip object, which maps a similar index of multiple containers. It takes an iterable, converts it into an iterator and aggregates the elements based on iterables passed. It returns an iterator of tuples

What is monkey patching in Python?

In Python, the term monkey patch only refers to dynamic modifications of a class or module at run-time.

What are Access Specifiers in Python?

Python uses the ‘_’ symbol to determine the access control for a specific data member or a member function of a class. A Class in Python has three types of [Python access modifiers](#): **Public Access Modifier**: The members of a class that are declared public are easily accessible from any part of the program. All data members and member functions of a class are public by default. **Protected Access Modifier**: The members of a class that are declared protected are only accessible to a class derived from it. All data members of a class are declared protected by adding a single underscore ‘_’ symbol before the data members of that class. **Private Access Modifier**: The members of a class that are declared private are accessible within the class only, the private access modifier is the most secure access modifier. Data members of a class are declared private by adding a double underscore ‘__’ symbol before the data member of that class

What is Python Switch Statement

From version 3.10 upward, Python has implemented a switch case feature called “structural pattern matching”. You can implement this feature with the match and case keywords. Note that the underscore symbol is what you use to define a default case for the switch statement in Python.

Note: Before Python 3.10 Python doesn't support match Statements.

- Python3

```
match term:
    case pattern-1:
        action-1
    case pattern-2:
        action-2
    case pattern-3:
        action-3
    case _:
        action-default
```

What are Exception Groups in Python?

The latest feature of Python 3.11, . The ExceptionGroup can be handled using a new except* syntax. The * symbol indicates that multiple exceptions can be handled by each except* clause.

ExceptionGroup is a collection/group of different kinds of Exception. Without creating Multiple Exceptions we can group together different Exceptions which we can later fetch one by one whenever necessary, the order in which the Exceptions are stored in the Exception Group doesn't matter while calling them.

What are Function Annotations in Python?

It is a feature that allows you to add metadata to function parameters and return values. This way you can specify the input type of the function parameters and the return type of the value the function returns.

Function annotations are arbitrary Python expressions that are associated with various parts of functions. These expressions are evaluated at compile time and have no life in Python's runtime environment. Python does not attach any meaning to these annotations. They take life when interpreted by third-party libraries

Global Interpreter Lock (GIL)?

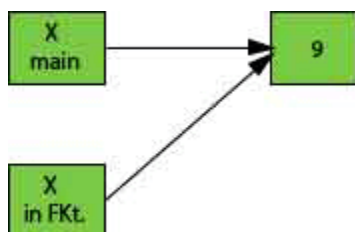
Python Global Interpreter (GIL) is a type of process lock that is used by Python whenever it deals with processes. Generally, Python only uses only one thread to execute the set of written statements. The performance of the single-threaded process and the multi-threaded process will be the same in Python and this is because of GIL in Python. We can not achieve multithreading in Python because we have a global interpreter lock that restricts the threads and works as a single thread

What is Python's parameter passing mechanism?

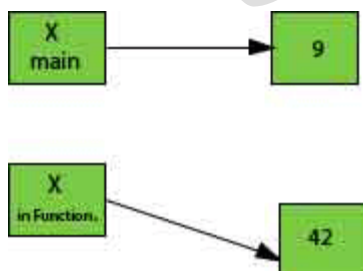
There are two parameters passing mechanism in Python: 1)Pass by references

2) Pass by value

By default, all the parameters (arguments) are passed "by reference" to the functions. Thus, if you change the value of the parameter within a function, the change is reflected in the calling function as well. It indicates the original variable. For example, if a variable is declared as `a = 10`, and passed to a function where it's value is modified to `a = 20`. Both the variables denote to the same value.



The pass by value is that whenever we pass the arguments to the function only values pass to the function, no reference passes to the function. It makes it immutable that means not changeable. Both variables hold the different values, and original value persists even after modifying in the function.



Python has a default argument concept which helps to call a method using an arbitrary number of arguments.

What is the difference between remove() function and del statement?

The user can use the remove() function to delete a specific object in the list.

Example:

1. `list_1 = [3, 5, 7, 3, 9, 3]`
2. `print(list_1)`
3. `list_1.remove(3)`
4. `print("After removal: ", list_1)`

Output:

```
[3, 5, 7, 3, 9, 3]  
After removal: [5, 7, 3, 9, 3]
```

If you want to delete an object at a specific location (index) in the list, you can either use **del** or **pop**.

1. `list_1 = [3, 5, 7, 3, 9, 3]`
2. `print(list_1)`
3. `del list_1[2]`
4. `print("After deleting: ", list_1)`

Output:

```
[3, 5, 7, 3, 9, 3]  
After deleting: [3, 5, 3, 9, 3]
```


Shabbbeer