1. What is PEP 8?

PEP 8 stands for **Python Enhancement Proposal**, it can be defined as a document that helps us to provide the guidelines on how to write the Python code. It is basically a set of rules that specify how to format Python code for maximum readability. It was written by Guido van Rossum, Barry Warsaw and Nick Coghlan in 2001.

1. What do you mean by Python literals?

Literals can be defined as a data which is given in a variable or constant. Python supports the following literals:

**String Literals**

String literals are formed by enclosing text in the single or double quotes. For example, string literals are string values.

**Example:**

1. # in single quotes
2. single = 'JavaTpoint'
3. # in double quotes
4. double = "JavaTpoint"
5. # multi-line String
6. multi = '''''Java
7. T
8. point'''
10. **print**(single)
11. **print**(double)
12. **print**(multi)

**Output:**

JavaTpoint

JavaTpoint

Java

T

point

**Numeric Literals**

Python supports three types of numeric literals integer, float and complex.

**Example:**

1. # Integer literal
2. a = 10
3. #Float Literal
4. b = 12.3
5. #Complex Literal
6. x = 3.14j
7. **print**(a)
8. **print**(b)
9. **print**(x)

**Output:**

10

12.3

3.14j

**Boolean Literals**

Boolean literals are used to denote Boolean values. It contains either True or False.

**Example:**

1. p = (1 == True)
2. q = (1 == False)
3. r = True + 3
4. s = False + 7
6. **print**("p is", p)
7. **print**("q is", q)
8. **print**("r:", r)
9. **print**("s:", s)

**Output:**

p is True

q is False

r: 4

s: 7

**Special literals**

Python contains one special literal, that is, **'None'**. This special literal is used for defining a null variable. If 'None' is compared with anything else other than a 'None', it will return false.

**Example:**

1. word = None
2. **print**(word)

**Output:**

None

1. **What is zip() function in Python?**

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

**Signature**

1. zip(iterator1, iterator2, iterator3 ...)

**Parameters**

**iterator1, iterator2, iterator3:** These are iterator objects that are joined together.

**Return**

It returns an iterator from two or more iterators.

#### Note: If the given lists are of different lengths, zip stops generating tuples when the first list ends. It means two lists are having 3, and 5 lengths will create a 3-tuple.

1. How to overload constructors or methods in Python?

Python's constructor: \_init\_\_ () is the first method of a class. Whenever we try to instantiate an object \_\_init\_\_() is automatically invoked by python to initialize members of an object. We can't overload constructors or methods in Python. It shows an error if we try to overload.

**Example:**

1. **class** student:
2. **def** \_\_init\_\_(self, name):
3. self.name = name
4. **def** \_\_init\_\_(self, name, email):
5. self.name = name
6. self.email = email
8. # This line will generate an error
9. #st = student("rahul")
11. # This line will call the second constructor
12. st = student("rahul", "rahul@gmail.com")
13. **print**("Name: ", st.name)
14. **print**("Email id: ", st.email)

**Output:**

Name: rahul

Email id: rahul@gmail.com

1. **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**.

**Example:**

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]

#### Note: You don't need to import any extra module to use these functions for removing an element from the list.

We cannot use these methods with a tuple because the tuple is different from the list.

1. What is the use of break statement?

The break statement is used to terminate the execution of the current loop. Break always breaks the current execution and transfer control to outside the current block. If the block is in a loop, it exits from the loop, and if the break is in a nested loop, it exits from the innermost loop.

**Example:**

1. list\_1 = ['X', 'Y', 'Z']
2. list\_2 = [11, 22, 33]
3. **for** i **in** list\_1:
4. **for** j **in** list\_2:
5. **print**(i, j)
6. **if** i == 'Y' **and** j == 33:
7. **print**('BREAK')
8. **break**
9. **else**:
10. **continue**
11. **break**

**Output:**

2

X 11

X 22

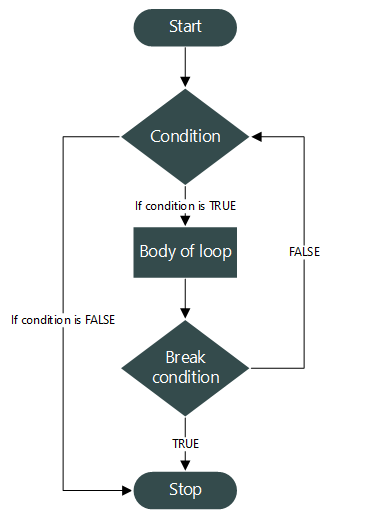
X 33

Y 11

Y 22

Y 33

BREAK



Python Break statement flowchart.

1. How is memory managed in Python?

Memory is managed in Python in the following ways:

* Memory management in python is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have access to this private heap. The python interpreter takes care of this instead.
* The allocation of heap space for Python objects is done by Python's memory manager. The core API gives access to some tools for the programmer to code.
* Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space.

1. **What is a generator in Python?**

In Python, the generator is a way that specifies how to implement iterators. It is a normal function except that it yields expression in the function. It does not implements \_\_itr\_\_ and next() method and reduce other overheads as well.

If a function contains at least a yield statement, it becomes a generator. The yield keyword pauses the current execution by saving its states and then resume from the same when required.

1. What is Pass in Python?

Pass specifies a Python statement without operations. It is a placeholder in a compound statement. If we want to create an empty class or functions, the pass keyword helps to pass the control without error.

**Example:**

1. **class** Student:
2. **pass** # Passing class
3. **class** Student:
4. **def** info():
5. **pass** # Passing function
6. **What is pickling and unpickling in Python?**

The Python pickle is defined as a module which accepts any Python object and converts it into a string representation. It dumps the Python object into a file using the dump function; this process is called **Pickling**.

The process of retrieving the original Python objects from the stored string representation is called as **Unpickling**.