



# [COMMON PYTHON INTERVIEW QUESTIONS]

*[By: Ojasi Kesari]*

## **1.Explain differences b/w List vs Tuples.**

While both Lists and Tuples are used to store a collection of heterogenous data, Lists are mutable while Tuples are immutable.

As a results, Lists offers more built-in operations and more flexibility in usage. On the other hand, Tuples are faster in iterations and consume less memory. Tuples can be used in areas that need an immutable data type (like dictionary keys), while Lists cannot.

## **2. What is a Decorator? Explain with example.**

Decorators are a powerful Python tool that allow to modify the functionality of other functions. They do so by extending the functionality of a function without changing the original function code. Since functions in Python are first-class objects, they can be treated like objects – including returning, passed as arguments, and assigned to other objects. Decorators utilize this property.

```
def decoratorFunc (func):  
    print("This is a decorator")  
    func()  
    print("Decoration complete")  
  
def originalFunc():  
    print("This is a function")  
  
originalFunc = decoratorFunc(originalFunc)
```

In the above example, (decoratorFunc) is a decorator that has its own functionality, as well as a wrapper function. When we assign this decorator function to (originalFunc), the func() in the decorator function now refers to (originalFunc). Thus, we get to use the functionality of (originalFunc), as well as the added functionality of (decoratorFunc).

### **3. Difference b/w List Comprehension, Dict Comprehension and Generator?**

In Python, Comprehension is simply a shorthand technique of creating collection of values.

List Comprehension involves iterating over a range of values and creating a list from those values that satisfy certain criteria. Example:

*[ n\*\*2 for n in range(10)] -> Creates list of squares of the first 10 numbers.*

Dictionary comprehension is similar, but instead of working on single values, it creates key-value pairs.

*{ i:j for (i,j) in zip(num, num\_cube)} -> Uses list of numbers and their cubes respectively, and creates a dictionary in the (n: n\*\*3) format.*

Generators are also similar to list comprehension in the sense that they too create a collection of certain values. However, unlike list comprehension, their results are not stored in the memory. This offers limited functionality with the values thus generated. However, generators are used

when memory is a constraint and the generated values are only needed for a single-use.

*(num\*\*2 for n in range(10)) -> Generator of the squares of first 10 numbers.*

## **4. How is memory managed in Python?**

In Python, memory is allocated primarily in two types of blocks – Stack and Heap.

The Stack memory store all the function calls and references in a program. Even if a function is declared earlier, the memory allocation only happens when a function call is made. All the objects declared within the function are stored in the Stack memory. When the function returns, its corresponding memory is de-allocated.

Heap memory is used to store the values of objects created in the main program body. This memory is accessible to all other objects and functions throughout the program.

In Python, values are stored in the memory and variables merely reference to this memory. When the value of a variable is changed, it references to another location in the memory where that value is stored. Through Reference Counting, Python keeps tracks of the number of references for any block of memory. When the count reaches zero, the memory is deallocated. This is known as Garbage Collection, which means that users do not have to manually de-allocate memory after program execution.

## **5. What is the difference b/w range and xrange?**

Both the Range and xrange functions are used to create a range of values. Both receive the same arguments – Start, Stop, Step. However, the Range function returns a list of values, while xrange returns a Generator object. Range offers more flexibility to work on the values, while xrange was faster and memory-efficient.

xrange cannot be used since Python 3.0.

## **6.What is a ternary operator?**

In simple terms, a ternary operator in Python is a shorthand way of writing conditional statements. It has three components – the condition, the if-true expression, and the if-false expression.

*[x if x > 5 else y] -> Ternary Operator*

## **7. What is Inheritance? How is it achieved in Python?**

Inheritance is the principle through which a child class inherits the properties of its parent class. Inheritance helps in maintain hierarchy between classes.

Like most OOP language, Python too supports Inheritance. This is done by passing the parent class while declaring a child class.

*def ChildClass(ParentClass):*

*pass*

The child class can be empty, in which case it will inherit all the properties of the parent class (including the `__init__` method). However, it is free to declare its own `__init__` method. Furthermore, using the `super()` method, the child class can access the methods and objects of the parent class (as opposed to the “self” keyword, which allows a class to refer to its own objects). In case of methods/objects with the same name in the child class, the parent’s inheritance will be overridden.

## **8. What is global and local variable in Python?**

Local variables are the variable declared within a local scope, like a function definition. Local variables can only be used within the scope they were declared.

Global variables are the variable declared outside any scope. As such, they are available to access throughout the program, including a local scope. If a local and a global variable existed with the same name, the local variable would be used by default within the local scope. To modify a global variable inside a local scope, it must be declared within it with the “global” variable.

## **9. Explain Break, Continue and Pass keywords.**

The Break keyword is used to halt the iteration of a loop. The Continue keyword is used to skip the remaining lines of code

in the loop block and immediately move to the next iteration of the loop.

The Pass keyword is used to define an empty method or class.

```
for num in range(10):
```

```
    if num%5 == 0:
```

```
        continue
```

```
    else:
```

```
        break
```

```
Class Sample():
```

```
    pass
```

## **10. Explain type conversion in Python.**

Type Conversion is a process in which the data type of an object is converted to another. It could be of two types – implicit and explicit.

In Implicit type conversion, Python automatically converts an object to another data type to better represent the results or for compatibility reasons. For example, the addition of an integer and a floating-point number results in a floating-point number.

In Explicit type conversion (type casting), the user can manually change the data type of the object. This is achieved using methods like `int()`, `float()`, `str()`, `list()`, `set()` etc.