Part 2

# PYTHON INTERVIEW QUESTIONS FOR EXPERIENCE HOLDERS

SWIPE

## Q1: What is Scope Resolution in Python?

Sometimes objects within the same scope have the same name but **function differently**. Here we use scope resolution.

Example:

Python modules namely 'math' and 'cmath' have common functions - log10(), acos(), exp() etc. To resolve, it is necessary to prefix them with their respective module, like math.exp() and cmath.exp().

## Q2: What are decorators in Python?

Decorators in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function. They are represented the **@decorator_name** in Python and are called in a bottom-up fashion.

The beauty of the decorators lies in the fact that besides adding functionality to the output of the method, they can even **accept arguments** for functions and can further modify those arguments before passing them to the function itself. The inner nested function, i.e. 'wrapper' function, plays a significant role here.

## Q3: What are Dict and List comprehensions?

Python comprehensions, like decorators, are **syntactic** sugar constructs that help build altered and filtered lists, dictionaries, or sets from a given list, dictionary, or set.

Using comprehensions **saves a lot of time** and code that might be considerably more verbose (containing more lines of code).

- Performing mathematical operations on the entire list
- Performing conditional filtering operations on the entire list
- Combining multiple lists into one
- Flattening a multi-dimensional list

## Q4: What is lambda in Python? Why is it used?

Lambda is an **anonymous function** in Python, that can accept any number of arguments, but can only have a single expression.

It is generally used in situations requiring an anonymous function for a short time period.

Lambda functions can be used in either of the two ways:

- Assigning lambda functions to a variable
- Wrapping lambda functions inside another function

## Q5: How do you copy an object in Python?

We use the copy module.

There are two ways of creating copies for the given object using the copy module -

- **Shallow Copy** is a bit-wise copy of an object. The copied object created has an exact copy of the values in the original object. If either of the values is a reference to other objects, just the reference addresses for the same are copied.
- **Deep Copy** copies all values recursively from source to target object, i.e. it even duplicates the objects referenced by the source object.

## Q6: What is the use of help() and dir() functions?

**help() function** is used to display the documentation of modules, classes, functions, keywords, etc.

**dir() function** tries to return a valid list of attributes and methods of the object it is called upon.

- For Modules/Library objects, it returns a list of all attributes, contained in that module.
- For Class Objects, it returns a list of all valid attributes and base attributes.
- With no arguments passed, it returns a list of attributes in the current scope.

## Q7: What is the difference between .py and .pyc files?

Major differences:

- .py files contain the **source code** of a program. Whereas, .pyc file contains the **bytecode** of your program.
- Before executing a python program python interpreter checks for the compiled files. If the file is present, the virtual machine executes it. If not found, it checks for .py file. If found, compiles it to .pyc file and then python virtual machine executes it.
- Having .pyc file saves you the compilation time.

## Q8: How Python is interpreted?

- Python as a language is not interpreted or compiled. Interpreted or compiled is the property of the implementation. Python is a bytecode(set of interpreter readable instructions) interpreted generally.
- Source code is a file with .py extension.
- Python compiles the source code to a set of instructions for a virtual machine. The Python interpreter is an implementation of that virtual machine. This intermediate format is called “bytecode”.
- .py source code is first compiled to give .pyc which is bytecode. This bytecode can be then interpreted by the official CPython or JIT(Just in Time compiler) compiled by PyPy.

Data-Driven
Science

**FOUND IT HELPFUL?**

Let us know in the comments