
Python Developer Questions

1. Basic Python Knowledge:

Explain the difference between Python 2 and Python 3.

Answer:

Python 2 and Python 3 are the different versions of Python programming language. Python 3 is the release to address the common issues of Python 2 and it enhances the coding flow when the use of Python 3. Also, Python 3 introduces new features to make the Python programs more effective and efficient.

Describe Python's data types, such as integers, strings, lists, dictionaries, and sets.

Answer:

Python supports for various type of data types such as integers, strings, lists, dictionaries and sets. Integers are numeric data types and python supports for other type of numeric data types also such as floats and complex numbers. Strings and lists are sequential data types. Lists are mutable and strings are immutable. Dictionaries are used for the mapping purposes and it's mutable. Sets are unordered collections and it's also mutable.

Describe your understanding of variables, data assignment, and variable scope.

Answer:

Variables can be changed after the initialization with value. Data assignment is assigning a value to the variable with the use of equal (=) symbol. Variable scope refers the accessibility of the variable where it can be accessed globally or locally within the Python program.

2. Control Structures:

Write a simple if statement to check a condition.

Answer:

```
If 5 > 3:  
    print("Greater")
```

Advice / write a code that uses a for loop to iterate over a list or range.

Answer:

```
For number in range(10):  
    print(number)
```

Tell us some example of using while loops

Answer:

When the need of iteration, while loops come into the action.

Examples,

- Looping through the list items or over the range of numbers.
- If a program should run infinitely then while loop will help with the use of simple while True statement.

3. Functions:

Define a function that takes parameters and returns a value.

Answer:

```
def sum(firstNum, secondNum):  
    return firstNum + secondNum
```

Describe about the usage of keyword arguments and default parameter values.

Answer:

Keyword arguments are used to obtain values as function arguments in order to increase the code readability and minimize the occurrences of semantic errors within the program. Default parameter values are used to assign default values to parameters those are taken by the function to perform any actions. when the user doesn't pass any values when the function calls, default parameter values will be used to perform any actions within the function. If the user wants to pass a value, then the default parameter values will be overwritten.

Request an example of a function that uses the return statement

Answer:

When the need of a value which should be derived from the execution of statements within a function. For an example, if the sum takes two or more values as arguments to return the calculated total value.

4. Data Structures

Tell us about your knowledge of lists and their methods (e.g., append, pop, index).

Answer:

Lists are used to store any data type values on it and it stores the data in the contiguous memory locations. It's mutable. Data values can be accessed and changed using indexes randomly. Lists have numerous amount of methods in order to perform operations such as insertion, deletion, update, searching and sorting. To insert values, append() method can be used and it stores the provided value at the end of the list. If a value needs to added to specific position of the list then insert() method will be useful. pop() method is used to pop out a last value in the list. It returns the last value of the list and removes it also. index() method is used to find the position of the value in the list.

Advice about work with dictionaries, including adding, modifying, and accessing keys and values.

Answer:

Dictionaries are the useful data type of Python. When the need of mapping, dictionaries play a crucial role in any python program. A dictionary contains keys and values which are referred by their keys. It's mutable and once the value is set with the key, it can be changed in the future. It's much easier to add and modify the values of dictionaries. Values can be added or modified using a key and its value even after the initialization of the dictionary. Keys can be accessed with the use of the function called keys().

5. Exception Handling

Write a code that handles exceptions using try and except blocks.

Answer:

try:

```
    print("It's about exception handling!")
```

except Exception as e:

```
    print(e)
```

Tell us about the purpose of the finally block

Answer:

finally block is used to execute some statements at the end of the try and except blocks. It executes always if exception occurs or not in the program also after the successful try block execution.

6. File Handling

Provide a code to read from and write to a text file.

Answer:

To perform read operation,

with `open("text.txt", "r")` as file:

```
data = file.read()
```

To perform write operation,

with `open("text.txt", "w")` as file:

```
file.write("New Data is inserted!")
```

Explain the difference between reading modes ('r', 'w', 'a').

Answer:

Reading modes are used to define the permissions those can be used to perform operations on the file. "r" is used to perform read operations on the file and if the file doesn't exist then it will throw an error. "w" is used to perform write operations on the file and if the file doesn't exist then it will create a new file or it will truncate the already existing file. "a" is used to perform appending operations like "w" but it appends the data at the end of the file and it doesn't truncate the already existing file.

7. Object-Oriented Programming (OOP)

Tell us about your understanding about the basics of classes and objects in Python.

Answer:

Classes are like blue prints to create objects. In python, classes are defined with using class keyword and all classes have functions to perform actions.

Create a simple class with attributes and methods.

Answer:

```
class Car:
    def __init__(self, model, color):
        self.model = model
        self.color = color

    def getModel(self):
        return self.model

    def setColor(self, color):
        self.color = color
```


8. Modules and Libraries

Tell us about the importing and using external modules (e.g., math, random).

Answer:

Modules are imported and used with the help of “from” and “import” keywords. External modules will reduce and save the significant amount of time while developing Python programs because, we don’t need to reinvent the wheel again and again. External modules can be large and those contains various kind of useful operations in order to perform actions effectively and efficiently with the importing and make simple function calls.

Tell us about the purpose of commonly used libraries like os, sys, or datetime

Answer:

Commonly used libraries like os, sys or datetime are helpful in most of the times. These have most of the common operations which can be performed in an OS and System level. Most of the common libraries are helpful to solve most problems easily in Python. For an example, to perform OS level operations then “os” module is useful. “sys” module is useful when we need command line arguments list. “datetime” is useful to deal with date, time and timestamps.

9. Basic Algorithms and Problem Solving

Present a coding problem that involves iterating over data and performing a simple operation (e.g., finding the sum of all even numbers in a list).

```
def findSumOfAllEvenNumbers(list):
```

```
    sum = 0
```

```
    for number in list:
```

```
        sum += number
```

```
    return sum
```

10. Coding Exercises

Write a Python code that could solve a problem by include tasks like reversing a string, calculating Fibonacci numbers, or implementing a simple data structure

Reversing a string:

```
def reverseString(string):  
    return string.reverse()
```

calculating Fibonacci numbers:

```
def iterateForFibonacciNumber(num):  
    if num <= 1:  
        return num  
  
    fibo = [0, 1]  
    for i in range(2, num + 1):  
        fibo.append(fibo[ i - 1 ] + fibo[ i - 2])  
  
    return fibo[num]  
  
for num in range(25):  
    print(iterateForFibonacciNumber(num))
```

Simple Data Structure – Stack:

```
class Stack:  
    def __init__(self):  
        self.items = []
```

```
def push(self, item):
    self.items.append(item)

def isEmpty(self):
    return len(self.items) == 0

def pop(self):
    if not self.isEmpty():
        return self.items.pop()
    else:
        print("Stack is empty!")

def peek(self):
    if not self.isEmpty():
        return self.items[-1]
    else:
        print("Stack is empty!")

def size(self):
    return len(self.items)
```

11. Version Control

Tell us about your understanding of basic Git commands

Answer:

Git command are used to perform actions based on their use case. I have a good understanding with Git commands in order to perform most important tasks with local Git and remote GitHub repositories. "git log" command is useful to view the commit history and "git status" command is useful to see the local repository's changes. Git command are useful to initialize a repository locally, adding files to it, make commits and overriding the commits can be done with the use of "git init", "git add" and "git commit". To deal with the branch to view and switch to another branch then "git branch", "git checkout" are used. To set and change the origin of the local repository then "git remote" command is used. To push and pull the code to remote repository which is in GitHub centralized server then "git push" and "git pull" are used. "git config" command can be used to set configurations. "git reset" command is used to move to last commit and if there any modification on files which are added to commit that will be removed but untracked files will not be removed. "git clean" command is used to remove untracked files by git which are not added by git. "git rebase" is used to rebase to specific commits to edit.