iNeuron

Python

Interview Questions

Q1. How will you perform static analysis in a Python application or find bugs?
Ans- PyChecker can be helpful as a static analyser to identify the bugs in the Python project. This also helps to find out the complexity-related bugs. Pylint is another tool that helps check if the Python module is at par with the coding standards.

Q2. What are the tools required to unit test your code?
Ans. To test units or classes, we can use the “unittest” python standard library. It is the easiest way to test code, and the features required are similar to the other unit testing tools like TestNG, JUnit.

Q3. How to get indices of N maximum values in a NumPy array?
Ans. With the help of below code, we can get the maximum values in a NumPy array :

```
Import numpy as nm

arr=nm.array([1, 6, 2, 4, 7])

print (arr.argsort() [-3:] [::-1])
```

Output:

```
[ 4 6 1]
```

Q4. How can you use ternary operators (Ternary) in Python?
Ans. Ternary operators are used to display conditional statements. This consists of the true or false values. Syntax :

The ternary operator is indicated as:

[on_true] if [expression] else [on_false] x, y = 25, 50big = x if x <y else y

Example: The expression is evaluated as if x <and else and, 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 returned as a result.

Q5. What does this mean? * args, ** kwargs? Why would we use it?
Ans. * Args is used when you are not sure how many arguments to pass to a function, or if you want to pass a list or tuple of stored arguments to a function.

** kwargs is used when we don't know how many keyword arguments to pass to a function, or it is used to pass the values from a dictionary as the keyword argument.

The args and kwargs identifiers are a convention, you can also use * bob and ** billy but that would not be wise

Q6. Does Python have OOps concepts?
Ans. Python is an object-oriented programming language. This means that any program can be solved in Python, creating an object model. However, Python can also be treated as a procedural and structural language.

Q7. How do I save an image locally using Python whose URL I already know?
Ans. We will use the following code to store an image locally from a URL

```
import urllib.request

urllib.request.urlretrieve (“URL”, “file-name.jpg”)
```

Q8. How are percentages calculated with Python / NumPy?
Ans. Percentages can be calculated using the following code:

```
import numpy as np

a = np.array ([1,2,3,4,5])

p = np.percentile (a, 50) #Returns 50%.

print (p)
```

Q9. When does Abnormal Termination occur?
Ans. First of all, I should mention that abend or abnormal termination is bad. You don’t want it to happen during your programming experience. However, it is practically unavoidable, in one way or another especially when you are a beginner.

Abend is an error in your program during its execution, while the main tasks continue to perform processes. This is caused by a code error or some software problem.

Q10. Explain the bytes() function in Python.
Ans. The bytes() function in Python returns a bytes object. It converts objects into bytes objects. It also creates empty bytes objects of the specified size.

Q11. Explain the ‘with statement’.
Ans. In Python, the ‘with statement’ is used for exception handling and resource management. It makes the code cleaner and readable as it allows a file to be opened and closed while executing a block of code containing the ‘with statement’.

Q12. What are Pandas Data Frames?
Ans. Pandas DataFrame is a two-dimensional tabular data structure with labelled axes. The data is aligned in a tabular manner in rows and columns. Data Frames are widely used in data science, machine learning, scientific computing, etc.

Here are some features of Dataframes:

2-dimensional
Labelled axes (rows and columns)
Size-mutable
Arithmetic operations can be performed on rows and columns.

Q13. How to combine Data Frames in Pandas?
Ans. We can combine Data Frames using the following functions:

Concat() function: It is used for vertical stacking.
pd.concat([data_frame1, data_frame2])

append(): It is used for horizontal stacking of DataFrames.
data_frame1.append(data_frame2)

join(): It is used to extract data from different DataFrames which have one or more columns common.

Q14. How to access the top n rows of a dataframe?
Ans. To access the top n rows of a data frame, we will use df.head(n).

Q15. How to access the last n rows of a dataframe?
Ans. We will use df.tail(n) to access the last n rows of a dataframe.

Q16. What are Python namespaces?
Ans. A namespace is a mapping from names to objects. It is a system that ensures that all the object names in a program are unique and can be used without any conflict. Python maintains a namespace in the form of a Python dictionary. These namespaces are implemented as dictionaries with 'name as key' mapped to its respective 'object as value'. Namespaces have different lifetimes as they are often created at different points in time.

Some of the namespaces in a Python program are:

Local Namespace – it contains local names inside a function. The local namespace is created for a function call and lasts until the function returns.
Global Namespace – It consists of the names from various imported modules that are being used in the ongoing project. This namespace is created when the package is imported into the script and lasts until the script ends.
Built-In Namespace – This namespace contains built-in functions and built-in exception names.

Q17. Write a python program to check if the number given is a palindrome or not

Ans. Below is the code to check if the given number is palindrome or not:

```
Num =input("Enter a number:")
if num==num[::-1]
        print ("It is a Palindrome!")
else:
        print("It is not a Palindrome!")
```

Q18. What is Scope Resolution in Python?

Ans: In some cases, objects within the same scope have the same name. However, they work differently. In such cases, scope resolution help in Python automatically.

Q19. How is data abstraction done in Python?

Ans**.** It can be achieved in Python using abstract classes and interfaces. Data abstraction only supplies the necessary details and hides the implementation.

Abstraction is selecting data from a larger pool to show only the relevant details to the object.