20 interview questions for Data Types, Operators, Conditional Statements, Looping Statements, Functions.

1. **Python is an interpreted language. Explain.**

   - An interpreted language is any programming language which executes its statements line by line. Programs written in Python run directly from the source code, with no intermediary compilation step.

2. **What are functions in Python?**

   - A function is a block of code which is executed only when a call is made to the function. def keyword is used to define a particular function as shown below:

     ```
     def function ():

     print ("Hi, Welcome to Edyoda")

     function (); # call to the function
     ```

     Output:

     ```
     Hi, Welcome to Edyoda
     ```

3. **What are Python namespaces? Why are they used?**

   - A namespace in Python ensures that object names in a program are unique and can be used without any conflict. Python implements these namespaces as dictionaries with 'name as key' mapped to a corresponding 'object as value'. This allows for multiple namespaces to use the same name and map it to a separate object. A few examples of namespaces are as follows:
     - **Local Namespace** includes local names inside a function. the namespace is temporarily created for a function call and gets cleared when the function returns.
     - **Global Namespace** includes names from various imported packages/ modules that are being used in the current project. This namespace is created when the package is imported in the script and lasts until the execution of the script.
     - **Built-in Namespace** includes built-in functions of core Python and built-in names for various types of exceptions.

     The **lifecycle of a namespace** depends upon the scope of objects they are mapped to. If the scope of an object ends, the lifecycle of that namespace comes to an end. Hence, it isn't possible to access inner namespace objects from an outer namespace.

4. **What are the applications of Python?**

   - Python is used in various software domains some application areas are given below.
     - Web and Internet Development
     - Games
     - Scientific and computational applications
     - Language development
     - Image processing and graphic design applications
     - Enterprise and business applications development
     - Operating systems
     - GUI based desktop applications
   - Python provides various web frameworks to develop web applications. The popular python web frameworks are **Django, Pyramid, Flask**.
   - Python's standard library supports for E-mail processing, FTP, IMAP, and other Internet protocols.

- Python's **SciPy** and **NumPy** helps in scientific and computational application development.
- Python's **Tkinter** library supports to create a desktop-based GUI application.

5. 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.

6. What is swap case() function in the Python?

- It is a string's function which converts all uppercase characters into lowercase and vice versa. It is used to alter the existing case of the string. This method creates a copy of the string which contains all the characters in the swap case. If the string is in lowercase, it generates a small case string and vice versa. It automatically ignores all the non-alphabetic characters. See an example below.

- **Example:**

```
string = "IT IS IN LOWERCASE."
print(string.swapcase())


string = "it is in uppercase."
print(string.swapcase())
```

- **Output:**

```
it is in lowercase.
IT IS IN UPPERCASE.
```

7. How to remove whitespaces from a string in Python?

- To remove the whitespaces and trailing spaces from the string, Python providies strip([str]) built-in function. This function returns a copy of the string after removing whitespaces if present. Otherwise returns original string.

- **Example:**

```
string = "  javatpoint "
string2 = "    javatpoint        "
string3 = "       javatpoint"
print(string)
print(string2)
print(string3)
print("After stripping all have placed in a sequence:")
print(string.strip())
print(string2.strip())
print(string3.strip())
```

Output:

```
javatpoint
   javatpoint
      javatpoint
After stripping all have placed in a sequence:
Javatpoint
javatpoint
javatpoint
```

8. What are the different file processing modes supported by Python?

- Python provides **four** modes to open files. The read-only (r), write-only (w), read-write (rw) and append mode (a). 'r' is used to open a file in read-only mode, 'w' is used to open a file in write-only mode, 'rw' is used to open in reading and write mode, 'a' is used to open a file in append mode. If the mode is not specified, by default file opens in read-only mode.

  - Read-only mode (r): Open a file for reading. It is the default mode.
  - Write-only mode (w): Open a file for writing. If the file contains data, data would be lost. Other a new file is created.
  - Read-Write mode (rw): Open a file for reading, write mode. It means updating mode.
  - Append mode (a): Open for writing, append to the end of the file, if the file exists.

9. What is pass in Python?

- The pass keyword represents a null operation in Python. It is generally used for the purpose of filling up empty blocks of code which may execute during runtime but has yet to be written. Without the **pass** statement in the following code, we may run into some errors during code execution.

```
def myEmptyFunc():
  # do nothing
  pass
myEmptyFunc()    # nothing happens
## Without the pass keyword
# File "<stdin>", line 3
# IndentationError: expected an indented block
```

10. What are modules and packages in Python?

- Python packages and Python modules are two mechanisms that allow for **modular programming** in Python. Modularizing has several advantages -

  - **Simplicity**: Working on a single module helps you focus on a relatively small portion of the problem at hand. This makes development easier and less error-prone.
  - **Maintainability**: Modules are designed to enforce logical boundaries between different problem domains. If they are written in a manner that reduces interdependency, it is less likely that modifications in a module might impact other parts of the program.
  - **Reusability**: Functions defined in a module can be easily reused by other parts of the application.
  - **Scoping**: Modules typically define a separate namespace, which helps avoid confusion between identifiers from other parts of the program.

Modules, in general, are simply Python files with a .py extension and can have a set of functions, classes, or variables defined and implemented. They can be imported and initialized once using the import statement. If partial functionality is needed, import the requisite classes or functions using from foo import bar.

11. What are global, protected and private attributes in Python?

- **Global** variables are public variables that are defined in the global scope. To use the variable in the global scope inside a function, we use the global keyword.
- **Protected** attributes are attributes defined with an underscore prefixed to their identifier eg. _sara. They can still be accessed and modified from outside the class they are defined in but a responsible developer should refrain from doing so.
- **Private** attributes are attributes with double underscore prefixed to their identifier eg. __ansh. They cannot be accessed or modified from the outside directly and will result in an AttributeError if such an attempt is made.

12. What is self in Python?

- **Self** is a keyword in Python used to define an instance of an object of a class. In Python, it is explicitly used as the first parameter, unlike in Java where it is optional. It helps in distinguishing between the methods and attributes of a class from its local variables.

13. What is __init__?

- __init__ is a contructor method in Python and is automatically called to allocate memory when a new object/instance is created. All classes have a **__init__** method associated with them. It helps in distinguishing methods and attributes of a class from local variables.

```
# class definition
class Student:
  def __init__(self, fname, lname, age, section):
      self.firstname = fname
      self.lastname = lname
      self.age = age
      self.section = section
# creating a new object
stu1 = Student("Sara", "Ansh", 22, "A2")
```

14. What is docstring in Python?

- Documentation string or docstring is a multiline string used to document a specific code segment.
- The docstring should describe what the function or method does.

15. What is slicing in Python?

- As the name suggests, ‘slicing’ is taking parts of.
- Syntax for slicing is **[start : stop : step]**
- **start** is the starting index from where to slice a list or tuple
- **stop** is the ending index or where to sop.
- **step** is the number of steps to jump.

- Default value for **start** is 0, **stop** is number of items, **step** is 1.
- Slicing can be done on **strings, arrays, lists**, and **tuples**.

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(numbers[1 : : 2])  #output : [2, 4, 6, 8, 10]
```

16. What is the difference between Python Arrays and lists?

- Arrays in python can only contain elements of same data types i.e., data type of array should be homogeneous. It is a thin wrapper around C language arrays and consumes far less memory than lists.
- Lists in python can contain elements of different data types i.e., data type of lists can be heterogeneous. It has the disadvantage of consuming large memory.

```
import array
a = array.array('i', [1, 2, 3])
for i in a:
   print(i, end=' ')    #OUTPUT: 1 2 3
a = array.array('i', [1, 2, 'string'])    #OUTPUT: TypeError: an integer is required (got type str)
a = [1, 2, 'string']
for i in a:
  print(i, end=' ')    #OUTPUT: 1 2 string
```

17. **What are the built-in types available in Python?**

- One of the most common python interview question, There are mutable and immutable built-in types.
- The mutable ones include:
  - List
  - Sets
  - Dictionaries
- The immutable types include:
  - Strings
  - Tuples
  - Numbers

18. **What is lambda function in Python?**

- It is often used as an inline function and is a single expression anonymous function. It is used to make a new function object and return them at runtime.

  Lambda is an anonymous function in Python that can accept any number of arguments and can have any number of parameters. However, the lambda function can have only a single expression or statement. Usually, it is used in situations that require an anonymous function for a short time period. Lambda functions can be used in either of the two ways:

  Here's an example of the lambda function:

  ```
  a = lambda x,y : x+y
  print(a(5, 6))
  Output: 11
  ```

19. **Difference between generators and iterators?**

- In Python, iterators are used to iterate over a group of elements (in a list, for example). The way of implementing these iterators is known as generators. It yields an expression in the function, but otherwise behaves like a normal function.

20. **How do you change the data type of a list?**

- To change a list into a tuple, we use the tuple() function
- To change it into a set, we use the set() function
- To change it into a dictionary, we use the dict() function
- To change it into a string, we use the .join() method