1. Why is Python called a platform independent language?

Ans: Python is called a platform independent language because it can be run in any operating system without any sort of modification at any level which also makes it a widely used programming language around the globe.

2. Can you mention some advantages of Python language?

Ans: Python has a lot of advantages. Such as:

1. One can write a python code once and can run it anywhere.
2. It has a vast library
3. It is object oriented
4. Python is object oriented
5. This language is compatible with multiple languages
6. Easy to read and write
7. It is also easy to learn

3. What do you understand by variable?

Ans: A variable in Python is a symbolic name that acts as a reference or container for storing data. It is actually a name for referring to an object.
For example: a = 5, here 'a' is a variable

4. What are the main data types in Python? Write brief note about each of them.

Ans: Main data type includes:

- **Integers (int):** Represents whole numbers.
  Example: x = 10

- **Strings (str):** Represents sequences of characters. It is enclosed within quotes ("").
  Example: name = "Python"

- **List (list):** Mutable sequences of elements.
  Example: fruits = ["apple", "banana", "cherry"]

- **Tuple (tuple):** Immutable sequences of elements.
  Example: coordinates = (10, 20, 30)
- **Dictionary (dict):** Data structure that stores the value and key and written as key:value
  Example: student = {"name": "John", "age": 21}

- **Floating-point (float):** Represents decimal numbers.
  Example: y = 3.14

- **Sets (set):** Unordered collection of unique elements.
  Example: unique_numbers = {1, 2, 3, 4}

- **Booleans (bool):** Represents logical values: True or False.
  Example: is active = True

5. Why do we need tuples even after having list?

Ans: The prime reason behind using tuples even after having lists is the immutability of tuples. Tuples cannot be changed so it is useful for cases where data are fixed and should not be changes which ensures security of the data. Also, tuple is faster than lists in terms of access time.

6. How many conditional statements are there? What do you understand by nested conditional statement?

Ans: There are three conditional statements:

- **<u>if Statement:</u>** If there is only one condition to match.

  if *condition*:
  executes if the *condition* is TRUE.

- **<u>if-else Statement:</u>** Matches one condition if it is TRUE, otherwise executes another if it is False.

  if *condition*:
  executes if the *condition* is TRUE

else:
executes if the *condition* is FALSE

- **<u>if-elif-else Statement:</u>** Matches multiple conditions if it is TRUE, executes the first TRUE condition

  if *condition1*:
  executes if the *condition1* is TRUE otherwise move to the next one

  elif *condtion2*:
  executes if the *condition2* is TRUE otherwise move to the next one

  else:
  executes if none of the conditions are TRUE

Nested Condition: A conditional statement inside another conditional statement is basically a nested condition.

```
if condition1:
    if condition2:
        print("Both conditions are true")
    else:
        print("Condition1 is true, but Condition2 is false")
else:
    print("Condition1 is false")
```

7. What do you understand by looping? What kind of loops are there? What do you understand by nested loop?

Ans: Looping means executing a block of statement over and over until a condition is satisfied. Looping is of two types:

- **For Loop:** A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

- **While Loop:** With the while loop we can execute a set of statements as long as a condition is true.

```
i = 1
while i < 6:
  print(i)
  i += 1
```

Nested Loop: A nested loop is a loop inside a loop. The "inner loop" will be executed one time for each iteration of the "outer loop"

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]
for x in adj:
  for y in fruits:
    print(x, y)
```

8. Which functions are used to take input and give output?

Ans:

Input Function:

- input(): Used to take input from the user as a string.

```
name = input("Enter your name: ")
print("Hello, " + name)
```

Output Function:

- print(): Used to display output on the screen.

```
print("Hello, World!")
```