**Interview Questions**

**Data Type:**

**Q1. What are data Types?**

Ans: Data types are set of values and set of operations defined on some specific data. Implementation of data type is an expression of data and operations in terms of specific programming languages such as Python, Java and C++.

**Q2. Name the data types in Python?**

Ans: There are basically 10 data types in python. These are, string, integer, float, Boolean, complex, tuple, list, dictionary, set and None.

str = "Suraj"/'Hello' # String: Stored in single or double quates

num = 25    # int:  Stores integers

flo = 2.5   # float: Stores decimal

boo = True/False # boolean: Stores True/False only

comp = 5+10j # Complex: sores data in this format

tup = (1,2,3,4,5) # Tuple: Stores data in () braces, can store heterogeneous data

lst = [1,True,"Suraj",2.5] # list: Stores data in [] braces, can store heterogeneous data

set1 = {1,2,3,4,5,6} #Set: Stores data in {} braces, no duplicates are allowed

dic = {1:"One",2:"Two",3:"Three"} #Dict: Stores data in {} braces, {key: value} pair

n = None

**Q3. Why these data types known as Python’s core data types?**

Ans: They are known as the core data types because they are part of the Python language itself and are always available to create other objects, you usually need to call functions in imported modules.

**Q4. What is an immutable data type and what are the immutable data types in python?**

Ans: An immutable data type is a data type which can not be modified once it is created. The immutable data types in python are int, float, decimal, Boolean, string, tuple.

**Q5. What is the major difference between Tuple and List?**

Ans: The major difference between Tuple and List in python is Tuple is an immutable data type and List in the other hand is a mutable data type. As we already know that in python except for List, Set and Dictionary data type every other data type is immutable.

**Q6. What does sequence mean, and which data types fall into this category?**

Ans: A sequence data type is a collection of elements by a specific position. In python string, list and tuple are data types based on sequence. The data types which are sequenced support indexing and slicing.

Ex**:** In python string, list and tuple are sequenced data type.

**Operators:**

**Q1. What are operators?**

Ans: Operators are defined as specific symbols or keyword used to perform some specific operations on values and variables.

**Q2. What are different types of operators in python?**

Ans: There are total 7 types of operators in python, these are given below-

1. Arithmetic operator **(+, -, \*, %, /, //, \*\*)**
2. Assignment operator **(+=, -=, \*=, %=, /=, //=, \*\*=)**
3. Comparison operator **(==, !=, <, >, <=, >=)**
4. Logical operator **(and, or, not)**
5. Identity operator **(is, is not)**
6. Membership operator **(in, not in)**
7. Bitwise operator **(&, |, ^, ~, <<, >>)**

**Q3. Show by example the execution of arithmetic operator.**

Ans:

a = 5

b = 2

add = a + b #Addition

sub = a - b #Substraction

mul = a \* b #Multiplication

div = a / b #Division (float)

div1 = a // b #Division (floor)

mod = a % b #Modulus

pow = a \*\* b #Power

# Output

print(add) # 7

print(sub) # 3

print(mul) # 10

print(div) # 2.5

print(div1) # 2

print(mod) # 1

print(pow) # 25

**Q3. What are comparison operators used for?**

Ans: Comparison operator compares two values and always returns boolean (True or False) value as output based on the condition provided.

x = 20

y = 15

#Output

print(x == y) # False

print(x != y) # True

print(x < y) # False

print(x > y) # True

print(x <= y) # False

print(x >= y) #True

**Q4. How are the logical operators used in python?**

Ans: These operators are used for checking multiple conditions. The output is always in boolean (True or False).

**and –** True if both the conditions are True

**or –**  True if at-least one condition is True

**not –** Reverse the output (from True to False and from False to True.)

x = 20

y = 15

# Output

print(x==y and x!=y) # False

print(x<y or x>y) # True

print(not x>=y) # False

**Q5. How are the identity operators used in python?**

Ans: There are two identity operators is and is not.

**“is” –** Returns True if both the variables are referring to the same object.

**“is not”** **–** Returns True if both the variables are not referring to the same object.

a = 10

b = 20

c = a

# Output

print(a is not b) #True

print(a is c) #True

**Conditional Statements:**

**Q1. What is a condition statement in Python?**

Ans: Conditional statements are also called decision making statements. In python we use these statements if we want to execute a block of code when the given condition is True or False.

**Q2. What are the different conditional statements in python?**

Ans: There are 3 types of conditional statements in python, these are- If, elif, else.

If – It is used to check the first condition in a program.

Elif – It is used after if statement to use another comparison.

Else - The elif keyword in python is a way of saying "if the previous conditions were not true, then try this condition.

**Q3. What is nested if statement?**

Ans: Nested if statement is one if statement inside another if statement. This allows you to test multiple criteria and increases the number of possible outcomes.

**Q4. By an example show the if else statement in a program.**

a = 20

b = 30

if a > b:

  print("a is greater than b")

else:

  print("b is greater than a") # Else part will get printed

**Q5. Use and (Logical operator) in a conditional statement.**

a = 60

b = 30

c = 70

if a > b and c > a:

  print("Both conditions are True") # If part will get printed

else:

    print("Both the conditions are not True")

**Q6. Use or (Logical operator) in a conditional statement.**

a = 60

b = 30

c = 50

if a > b or c > a:

  print("At least one of the condition is True") # If part will get printed

else:

    print("Both the conditions are not True")

**Looping Statements:**

**Q1. Why do we use loops in python and state the types of loops?**

Ans: In python or any other programming language loops are used to iterate(traverse) over a sequence of data. There are two types of loops in python- 1. For Loop, 2. While Loop.

**Q2. Discuss about for loop in python.**

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

With the for loop, we can execute a set of statements, once for each item in a list, tuple, set etc.

#For loop for iterating a list

lst = ["Suraj", "Rakesh", "Priyanka"]

for x in lst:

    print(x)

#Output:

        #Suraj

        #Rakesh

        #Priyanka

#For loop for iterating a string

name = "Suraj"

for x in name:

    print(x, end = " ") #Output: S u r a j

**Q3. Give an example of loop with range function.**

Ans:

for x in range(10): #starts from 0, ends at 9

    print(x, end=" ")#Output: 0 1 2 3 4 5 6 7 8 9 10

for x in range(1,11):  #starts from 1, ends at 10

    print(x, end = " ") #Output: 1 2 3 4 5 6 7 8 9 10

for x in range(1,11,2):  #starts from 1, ends at 10 with increment 2

    print(x, end = " ") #Output: 1 3 5 7 9

**Q4. What is the need of an else statement in a for loop?**

Ans: If we want to print some specific line of code after the execution of the for loop, then we use the else statement after the for loop.

The else keyword in a for loop specifies a block of code to be executed when the loop is finished.

**Q5. What is a nested loop discuss with the help of an example?**

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

for x in range(1,11):

    for y in range(1,11):

        print(x,"\*",y,"=",x\*y)

    else:

        print("Multiplication table of: ",y)

# Output: Multiplication table from 1 to 10

**Q6. Discuss about while loop in python.**

Ans: With the while loop we can execute a set of statements as long as a condition is true.

x = 1

while x <= 10: # print x as long as x<=10

    print(x, end=" ")

    x+=1 #output: 1 2 3 4 5 6 7 8 9 10

**Q7. What are different transfer statements in python?**

Ans: There are 3 transfer statements in python, these are-

Break- With the break statement we can stop the loop even if the while condition is true.

Continue - With the continue statement we can stop the current iteration and continue with the next.

Pass - With the continue statement we can stop the current iteration, and continue with the next

**Functions:**

**Q1. What is the use of function in python discuss in brief?**

Ans: We use functions in programming to bundle a set of instructions that we want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task.

def <function\_name>:

    function\_body

    write some action

    return value

**Q2. How many types of functions do we have in python?**

* Python Built-in Functions.
* Python Recursion Functions.
* Python Lambda Functions.
* Python User-defined Functions.

**Q3. What are arguments?**

Ans: Information can be passed into functions as arguments. Arguments are specified after the function name, inside the parentheses. we can add as many arguments as we want, separated by a comma.

def add(a,b,c): # Here a,b,c are arguments or parameters

    print(a+b+c)

# Calling the function name and giving value to the parameters

add(1,2,3) #output: 6

**Q4. How many types of arguments are there in python?**

Ans: There are 4 types of arguments in python-

* Required/Positional arguments
* Default arguments
* Keyword arguments
* Variable length arguments – This is again divided into 2 types-

1. Arbitrary positional argument (\*args)
2. Arbitrary keyword argument (\*\*kwargs)

**Q5. Discuss recursion function in brief with an example.**

Ans: Python also accepts function recursion, which means a defined function can call itself. Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

#Factorial of a number using recursion function

def factorial(n):

    if n == 1:

        return n

    else:

        return n\*factorial(n-1) #Use of recursion

res = factorial(5) #Calling the function

print(res) #Output will be 120