iNeuron


# Python

## Interview Questions

Q1. What are identity operators?
Ans- this is used to verify whether two values are on the same part of the memory or not. There are two types of identity operators:

a) Is: return true if two operands are identical
b) is not: returns true if two operands are not identical

Q2. What is the difference between a=10 and a==10?

Ans- The expression a=10 assigns the value 10 to variable a, whereas a==10 checks if the value of is equal to 10 or not. If yes then it returns 'True' else it will return 'False'.

Q3. What is an expression?
Ans- Logical line of code that we write while programming, are called expressions. An expression can be broken into operators and operands . It is therefore said that an expression is a combination of one or more operands and zero or more operators that are together used to compute a value.

Q4. What are the basic rules of operator precedence in python?

Ans- The basic rule of operator precedence in python is as follows:

1. Expressions must be evaluated from left to right
2. Expressions of parenthesis are performed first.
3. In python the operation procedure follows as per the acronym PEMDAS:
   a) Parenthesis

b) Exponent
c) Multiplication
d) Division
e) Addition
f) Subtraction

4. Mathematical operators are of higher precedence and the Boolean operators are of lower precedence. Hence, mathematical operations are performed before Boolean operations.

Q5. Arrange the following operators from high to low precedence.
a) Assignment
b) Exponent
c) Addition and Subtraction
d) Relational Operators
e) Equality operators
f) Logical operators
g) Multiplication, division, floor division and modulus

Ans- 1. Exponent
2. Multiplication, division, floor division and modulus
3. Addition and Subtraction
4. Relational Operators
5. Equality operators
6. Assignment
7. Logical operators

Q6. Is it possible to change the order of evaluation in an expression.

Ans- Yes, it is possible to change order of evaluation of an expression. Suppose you want to perform addition before multiplication in an expression, then you can simply put the addition expression in parenthesis.

Q7. What is the difference between implicit and explicit expression?
Ans- Conversion is the process of converting one data type into another. Two types of conversion in Python as follows:

1. Implicit type conversion
2. Explicit type conversion

Q8. What is a statement?
Ans- A complete unit of code that Python interpreter can execute is called statement.

Q9. What is input statement?
Ans- The input statement is used to get user input from the keyboard. The syntax for input () function is as follows:

```
In [2]: x=int(input("eter any no. "))
        print(x)

        eter any no. 982733
        982733
```

Q10. Look at the following code and find the solution.

Ans-

```
In [3]: #find the solution

        num1=int(input("enter the first no. "))
        num2=int(input("enter the second no. "))

        print(num1+num2)
```

```
enter the first no. 976351000
enter the second no. 1283320098
2259671098
```

Q11. What is Associativity of Python operators? What are non-associative operators?
Ans- Associativity defines the order in which an expression will be evaluated if it has more than one operator having same precedence. In such a case generally left to right associativity is followed.

Operators like assignment or comparison operators have no associativity and known as Non associative operators.

Q12. What are control statements?
Ans- They are used to control the follow of program execution. They help in deciding the next steps under specific conditions also allow repetitions of program for certain number of times.

Two types of control statements are as follows:

1. Conditional branching
   - If

```
In [6]: a = 33
        b = 200
        if b > a:
            print("b is greater than a")
```

```
b is greater than a
```

- If.... else

```
In [11]: a = 200
         b = 33
         if b > a:
           print("b is greater than a")
         else:
           print("b is not greater than a")

         b is not greater than a
```

- Nested if statements

```
In [12]: x=784839285721
         y=982392392
         z=872362837

         if z>x:
             print('wow')
         elif x==y==z:
             print('yeah')
         else:
             print('this is right')

         this is right
```

2.Loops

While: repeat a block of statements as long as a given condition is true

For: repeat a block of statements for certain number of times.

Q13. Write a code to print the star pattern.

Ans-

```
In [10]: for i in range(1,5):
             print("*"*i)
```

```
*
**
***
****
```

```
In [11]: #2nd method

count=1
while count < 5:
    print("*"*count)
    count = count + 1
```

```
*
**
***
****
```

Q14. Write code to produce the following pattern:

1
22
333
4444

Ans-

```
In [12]: count = 1
while count < 5:
    print(str(count)*count)
    count = count+1
```

```
1
22
333
4444
```

Q15. Write a code to generate the following pattern.

1

12
123
1234
Ans-

```
In [13]: count = 1
         string1 =''
         while count < 5:
             for i in range(1, count+1):
                 string1 = string1+str(i)

             count = count +1
             print(string1)
             string1 =''
```

```
1
12
123
1234
```

Q16. Write code to spell a word entered by the user.
Ans-

```
In [15]: word = input("please enter any word ")
         for i in word:
             print(i)
```

```
please enter any word ineuron
i
n
e
u
r
o
n
```

Q17. What are the statements to control a loop?
Ans – The following three statements can be used to control as loop:

a) break: breaks the execution of the loop and jumps to next statement after the loop.

b) Continue: takes the control back to the top of the loop without executing the remaining statements.

c) pass: does nothing

Q18. What is the meaning of conditional branching?
Ans- Deciding whether certain sets of instructions must be executed or not based on the value of an expression is called conditional branching.

Q19. What is the difference between the continue and pass statement?
Ans- Pass does nothing whereas continue starts the next iterations of the loop.

Q20. What is Self-used for in Python?

Ans- It is used to represent the instance of the class. This is because in Python, the '@' syntax is not used to refer to the instance attributes.