Module 1: If-Else Assignment

Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: You work in XYZ Corporation as a Data Analyst. Your company has told you to work with the if-else condition. Tasks To Be Performed:

1. Input the values of a and b as 10 and 20 respectively. Now check if a is greater or b is greater using if condition. Think about all the edge cases, and print the statements accordingly.

In [1]:

```
##1,
a=10
b=20
if a>b:
    print("hello")
elif b>a:
    print("hi")
else:
    print("hey")
```

```
hi
```

Module 1: Conditional Statements

Assignment Problem Statement: You work in XYZ Corporation as a Data Analyst. Your company has told you to work with the if-else condition. Tasks To Be Performed:

1. Take three user inputs and print the greatest number from those inputs using if-else condition. Edge cases, if any, should also be handled

In [2]:

```
a=10
b=20
c=30
if a>b and a>c:
  print("a is greater")
elif b>a and b>c:
    print("b is greater")
elif c>a and c>b:
    print("c is greater")
else:
    print("nothing")
```

```
c is greater
```

Module 1: Loops

Assignment - 1 Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: You work in XYZ Corporation as a Data Analyst. Your company has told you to work with the looping statements. Tasks To Be Performed:

1. Print the numbers from 1 to 10 using while loop.

In [3]:

```
i=1
while(i<=10):
    print(i)
    i+=1
```

```
1
2
3
4
5
6
7
8
9
10
```

Module 1: Loops

Assignment-2 Python for Data Science Certification Course Problem Statement: You work in XYZ Corporation as a Data Analyst. Your company has told you to work with the looping statements. Tasks To Be Performed:

1. Create a list that has 10, 23, 4, 26, 4, 75, 24, 54 values and with the help of while loop fetch the even numbers and print the numbers.

In [8]:

```
list = [10, 23, 4, 26, 4, 75, 24, 54]
i = 0
while i < len(list):
    if list[i] % 2 == 0:
        print(list[i])
    i += 1
```

```
10
4
26
4
24
54
```

Module 1: Case Study - 1

Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python Certification Course Problem Statement: Consider yourself to be Sam who is a data scientist. He has been invited as a guest lecturer at a college to take an introductory session on Python. Tasks To Be Performed:

1. Create a list containing squares of numbers from 1 to 10 (HINT: use List Comprehension).
2. Write a function to check if the year number is a leap year.
3. Write a function to take an array and return another array that contains the members of the first array that are even.
4. Write a function that takes 2 arrays and prints the members of the first array that are present in the second array. (HINT: use Membership Comprehension)

In [9]:

```
##1,
num=[1,2,3,4,5,6,7,8,9,10]
squares=[]
nums=[squr*squr for squr in num]
print(nums)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

In [ ]:

```
##2,
def leap(year):
        if (year % 400 == 0):
            print(year,'is leap year')
        elif(year % 4 == 0 and year % 100 != 0):
            print(year,"is year leap year")
        else:
            print(year,"is year not leap year")

year = int(input("enter the year:"))
leap(year)
```

In [ ]:

```
##3,
array=[1,2,3,4,5,6]
def even(array):
    new_array = [array]
    for i in array:
        if i%2==0:
            new_array.append
    return new_array
print(even(array))
```

In [ ]:

```
##4,
x=np.array([1,2,3,6,7])
y=np.array([1,2,3,8,9])
common = []
for i in range(0,len(x)):
   if x[i] in y:
       common.append(x[i])
print(common)
```

Module 1: Case Study - 2

Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python Certification Course Problem Statement: Consider yourself to be Sam who is a data scientist. He has been invited as a guest lecturer at a college to take an introductory session on Python. Tasks To Be Performed:

1. Create 1st tuple with values -> (10, 20, 30), 2nd tuple with values -> (40, 50, 60): a. Concatenate the two tuples and store it in "t_combine" b. Repeat the elements of "t_combine" 3 times c. Access the 3rd element from "t_combine" d. Access the first three elements from "t_combine" e. Access the last three elements from "t_combine"
2. Create a list 'my_list' with these elements: a. First element is a tuple with values 1, 2, 3 b. Second element is a tuple with values "a", "b", "c" c. Third element is a tuple with values True, False
3. Append a new tuple – (1, 'a', True) to 'my_list': a. Append a new list – *"sparta", 123+ to my_list
4. Create a dictionary 'fruit' where: a. The first key is 'Fruit' and the values are ("Apple", "Banana", "Mango", "Guava") b. The second key is 'Cost' and the values are (85, 54, 120, 70) c. Extract all the keys from 'fruit' d. Extract all the values from 'fruit'
5. Create a set named 'my_set' with values (1, 1, "a", "a", True, True) and print the result.

In [ ]:

```
##1,
t1=(10,20,30)
t2=(40,50,60)
print("The original tuple 1 : " + str(t1))
print("The original tuple 2 : " + str(t2))
t_combine = t1 + t2
N=3

res = [t_combine for i in range(N)]
res = tuple(res)
# printing result
print("The duplicated tuple elements are : " + str(res))
print(t_combine)
t_combine[3]
t_combine[:3]
t_combine[3:]
```

In [ ]:

```
##2,
my_list = [(1, 2, 3), ("a", "b", "c"), (True, False)]
print(my_list)
```

In [ ]:

```
##3,
a_tuple = (1, 'a', True)
a_tuple = a_tuple + (*'sparta', 123,)

print(a_tuple)
```

In [ ]:

```
##4,
fruit = {'Fruit': ["Apple", "Banana", "Mango", "Guava"], 'Cost': [85, 54, 120, 70]}
print(fruit)
type(fruit)
len(fruit)
```

In [ ]:

```
##5,
my_set = {1, 1, 'a', 'a', True, True}
print(my_set)
```