Module 02. Control Structures

September 26, 2018

1 Control Structures

Programming often involves examining a set of conditions and deciding which action to take based on those conditions. Python's if statement allows you to examine the current state of a program and respond appropriately to that state.

1.1 Decision Making - (if - else) Statement

Ex. WAP to check entered number is even or odd (if - else)

```
In [4]: x = int(input("Enter a no: "))
        if (x % 2) == 0:
            print(x, " is an even no")
        else:
            print(x, " is an odd no")
```

```
Enter a no: 5
5  is an odd no
```

Ex. Grade system example (if - elseif - else (Ladder))

```
In [5]: marks = int(input("Enter your marks:"))
        if (marks >= 75):
            print("You are passed in distinction")
        elif (marks < 75 and marks >= 60):  # use and or &
            print("You are passed in First Class")
        elif (marks < 60 and marks >= 40):
            print("You are passed in Second class")
        else:
            print("You are failed in your exam")
```

```
Enter your marks:60
You are passed in First Class
```

Ex. Print largest of three numbers – nested if..else

```
In [6]: print("Enter three numbers")
        a, b, c = int(input()), int(input()), int(input())

        if(a==b and a==c):
            print("all r equal")

        elif(a>b):

            if(a>c):
                print(a, "is largest")
            else:
                print(c, "is largest")

        elif(b>c):
            print(b, "is largest")

        else:
            print(c, "is largest")
```

```
Enter three numbers
3
5
7
7 is largest
```

Ex. check if a is present in string "Mumbai" – Membership Operator (in)

```
In [7]: if "a" in "Mumbai":
            print("Present")
        else:
            print("Not present")
```

```
Present
```

1.2 Iteration - while

Ex. Find number of digits in a number and then find sum of digits of a number

```
In [12]: num = int(input("Enter a number"))
         sum = 0
         count = 0

         while num > 0:
             count += 1
             digit = num % 10
```

```
            sum += digit
            num = num // 10

        print("No. of digits =", count, "and sum of digits=", sum)
```

```
Enter a number1234
No. of digits = 4 and sum of digits= 10
```

1.2.1 - Continue and break

Ex. WAP that keeps accpeting character from user till user enters 'q'

```
In [8]: while True:
            ch = input("enter a character")
            if ch == 'q':
                break
        print("end of program!")
```

```
enter a charactere
enter a characterw
enter a character5
enter a characterq
end of program!
```

Ex. WAP that prints all number from 1 to 50 skipping multipes of 3

```
In [11]: x = 0
         while x <= 50:
             x += 1
             if x % 3:
                 print(x, end=" ")
             else:
                 continue
```

```
1 2 4 5 7 8 10 11 13 14 16 17 19 20 22 23 25 26 28 29 31 32 34 35 37 38 40 41 43 44 46 47 49 5
```

1.3 Iteration - for

1.3.1 - range()

range(start, stop, step) - Return a virtual sequence of numbers from start to stop by step.

```
In [16]: for i in range(10): # default start = 0, end = 9, default step = 1
             print(i, end= " ")
```

```
0 1 2 3 4 5 6 7 8 9
```

```
In [17]: for i in range(1, 10): # start = 1, end = 9, default step = 1
             print(i, end= " ")

1 2 3 4 5 6 7 8 9

In [18]: for i in range(1, 10, 2): # start = 1, end = 9, step = 2
             print(i, end= " ")

1 3 5 7 9

In [19]: # Reverse numbering

         for i in range(10, 0, -1): # start = 10, end = 1, step = -1
             print(i, end= " ")

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

Ex. Find factorial of a number

```
In [14]: n=int(input("Enter a number : "))
         fact=1
         for i in range(1,n+1,1):
             fact *= i
         print ("Factorial of",n,"is",fact)

Enter a number : 5
Factorial of 5 is 120
```