

# Problem Solving and Programming

**Date 12 June 2019**

## **Day Objectives**

*String Slicing*

*Functions in Python*

*Basic Problems related to conditional statements using functions.*

*Iterations in Python*

In [ ]:

## **String Slicing**

```

In [1]: st="Python"
        l=len(st)
        st[0] # accessing the first character
        st[-1] # accesing the last character
        st[l-1] # accessing the last character
        st[-2] # accessing the penultimate character
        st[0:2] # using the range we can extract particular characters in string:upper
                #bound is inclusive, lower bound is exclusive
        st[-2:] #accessing the last two characters

        # Access all characters except first and last character
        s="Python"
        print(s[1:-1])

        # Access middle character of even length string
        # s1="Python"
        # print(s1[len(s1)//2])

        # Access middle character of odd length string
        s1='Pythons'
        print(s1[len(s1)//2])

        s2="Python"
        print(s2[-1::-1]) #Reverse of string
        print(s2[-1:-3:-1]) #Last 2 characters in reverse order

        #Reverse middle two characters in even length string
        #s3="Python"
        #s3[len()]

        s4="Python"
        s4[0:len(s4)+1:2] #Accessing alternate characters in the string
        s4[0:len(s4)+1:-2] #Accessing the alternate characters in the string in reverse
                           #order.

```

```

ytho
h
nohtyP
no

```

```
Out[1]: ''
```

```
In [ ]:
```

## Functions

```

In [2]: #Function to reverse a string
        def reverseString(n):
            return n[-1::-1]
        reverseString('Python')

```

```
Out[2]: 'nohtyP'
```

```
In [3]: #Create a function to check if a string is palindrome
def Palindrome(p):
    c=p
    if(c==p[-1::-1]):
        return True
    else:
        return False
Palindrome('BOB')
```

Out[3]: True

```
In [4]: #Function to check if a given year is Leap year.
def LeapYear(year):
    if(year % 400 == 0 or (year % 100 != 0 and year % 4==0)):
        return True
    return False
LeapYear(2000)
```

Out[4]: True

```
In [5]: #Function to count number of digits in a given number
def num(m):
    r=1
    c=0
    while m>0:
        r=m%10
        c+=1
        m=m//10
    return c
num(123)
```

Out[5]: 3

```
In [6]: # Function to check greatest of four numbers
def greatest(n1,n2,n3,n4):
    if n1>n2 and n1>n3 and n1>n4:
        return n1
    elif n2>n3 and n2>n4:
        return n2
    elif n3>n4:
        return n3
    else:
        return n4
greatest(10,90,150,200)
```

Out[6]: 200

In [ ]:

## Iteration

- for
- while

```
In [7]: # Function to print n numbers
def printNaturalnumbers(N):
    for i in range(1,N+1):
        print(i, end=" ")
    return
printNaturalnumbers(10)
```

1 2 3 4 5 6 7 8 9 10

```
In [8]: # Function to print n numbers using while Loop
def Naturalwhile(n):
    c=1
    while c<=n:
        print(c, end=" ")
        c+=1
    return
Naturalwhile(25)
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

```
In [9]: # Function to print all numbers divisible by 6 and not a factor of
# 100 in a given range (lb,ub) inclusive

def divFactor(lb,ub):
    for i in range(lb,ub+1):
        if(i%6==0 and 100%i!=0):
            print(i,end=" ")
    return
divFactor(10,100)
```

12 18 24 30 36 42 48 54 60 66 72 78 84 90 96

```
In [10]: # Function to find average of cubes of all even numbers in given range(lb,ub)
# inclusive
def average(lb,ub):
    c=0
    s=0
    cubes=1
    avg=1
    for i in range(lb,ub+1):
        if(i%2==0):
            c+=1
            cubes=i*i*i
            s=s+cubes
    avg=s/c
    print(avg)
    return
average(1,5)
```

36.0

```
In [11]: # Function to generate list of factors for a given number
def Factors(n):
    for i in range(1,n+1):
        if(n%i==0):
            print(i,end=' ')
    return
Factors(12)
```

1 2 3 4 6 12

```
In [12]: # Function to calculate factorial of a number
def Factorial(number):
    m=1
    while(number>=1):
        m=m*number
        number=number-1
    return m
Factorial(5)
```

Out[12]: 120

```
In [32]: # Function to check if a number is prime.
def Prime(p):
    c=0
    for i in range(2,p):
        if(p%i==0):
            c+=1
    if(c==0):
        return "Prime"
    else:
        return "Not Prime"
Prime(12)
```

Out[32]: 'Not Prime'

```
In [36]: # Function to calculate the average of first n prime numbers
def avgNprimes(n):
    sum=0
    primecount=0
    seqcount=2
    while(primecount<n):
        if Prime(seqcount):
            primecount+=1
            sum+=seqcount
            seqcount+=1
    return sum/n
avgNprimes(10)
```

Out[36]: 6.5

```
In [15]: # Function to generate all perfect numbers in a given range(lb,ub)inclusive.
def Perfectnumbers(lb,ub):
    d=1
    s=0
    for i in range(lb,ub+1):
        if(i%d==0):
            s=s+d
            d+=1
            if(s==i):
                return s
    Perfectnumbers(10,25)
```

## Function to print reverse of a given range in same line

```
In [ ]: def reverseRange
```