

# Problem Solving And Programming

Date 12 June 2019

## Day Objectives

- String Slicing
- Function in Python
- Basic Problems related to conditional statements using functions
- Iteration in python
- Python Data Structures-lists,tuples and dictionaries
- Basic operations on data structures
- Applying Data Structures to solve problems

In [ ]:

1

## String Slicing

In [22]:

```
1 s1="python"
2
3 s1[0] ###Accessing the first character in a string
4
5 s1[1] ###Accessing the second character in a string
6
7 s1[len(s1)-1] ### Accessing the last character in a string
8
9 s1[-1] ### Accessing the last character in a string
10
11 s1[-2] ### Accessing the last penultimate character in a string
12
13 s1[0:2] ### Accessing the first two characters in a string
14
15 s1[-2:] ### Accessing the last two characters in a string
16
17 s1[4:] ### Accessing the character 5th character to end of a string
18
19 ### Accessing all characters except first and last character
20
21 s1[1:-1]
22
23 #### Accessing the middle character in a string
24
25 s1[len(s1)//2]
26
27 s1[len(s1)//3]
28
29 ###Reverse of a string
30 s1[-1::-1]
31
32 s1[-1:-3:-1] ## Accessing the last two characters in a reverse string
33
34 #reverse the middle two characters in an even length string
35
36 s1[len(s1)//2:(len(s1)//2)-2:-1]
37
38 ##Alternative characters of a string same order
39 ### "python" - "pto"
40
41 s1[::-2]
42
43 ##Accessing alternate character of a string in reverse order
44 ###"python" - "nhy"
45
46 s1[::-2]
47
48
49
50
51
```

Out[22]: 'nhy'

```
In [15]: 1 s1[0]
          2
```

```
Out[15]: 'p'
```

```
In [11]: 1 s1[-1]
```

```
Out[11]: 'v'
```

```
In [13]: 1 s1[len(s1)-2]
```

```
Out[13]: 'c'
```

```
In [ ]: 1
```

## functions

```
In [28]: 1 ###Function to reverse of a string
          2 def reverseString(s1):
          3     return s1[::-1]
          4
          5 reverseString("Python")
```

```
Out[28]: 'nohtyP'
```

```
In [38]: 1 ###Function to check if a string is Palindrome or not
          2 s=str(input("enter the string: "))
          3 def palindrome(s):
          4     if s==s[::-1]:
          5         return True
          6     else:
          7         return False
          8 palindrome(s)
```

enter the string: 101

```
Out[38]: True
```

```
In [43]: 1 ###Function to check if a given year is Leap year
          2 y=int(input("Enter the year:"))
          3 def year(y):
          4     if(y%400==0 or (y%100!=0 and y%4==0)):
          5         return True
          6     else:
          7         return False
          8
          9 year(y)
         10
```

Enter the year:1996

```
Out[43]: True
```

```
In [11]: 1  ### count no of digits
2  n=int(input("Enter the number:"))
3  def num(n):
4      c=0
5      while n!=0:
6          c=c+1
7          n=n//10
8
9
10     return c
11 num(n)
12
```

Enter the number:123

Out[11]: 3

```
In [1]: 1  def number(n):
2      return len(n)
3  n=(input("Enter the string:"))
4  number(n)
```

Enter the string:123

Out[1]: 3

```
In [17]: 1  def greatest(n1,n2,n3,n4):
2      if n1 > n2 and n1 > n3 and n1 > n4:
3          return n1
4      elif n2 > n3 and n2 > n4:
5          return n2
6      elif n3 > n4:
7          return n3
8      return n4
9  greatest(1,12,13,14)
```

Out[17]: 14

```
In [ ]: 1  ### Iterations
2  - for
3  - while
4  #### for loop in python
5  [lb,ub]
6      for number in range (lb,ub+1)
7
```

In [29]:

```

1  ###Function to print n natural nos
2
3  def printnNaturalNumbers(n):
4      for i in range(1,n+1):
5          print(i,end=" ")
6      return
7
8  printnNaturalNumbers(30)
9  printnNaturalNumbers(10)
10
11
12

```

```

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 26 27 28 29 3
0 1 2 3 4 5 6 7 8 9 10

```

In [32]:

```

1  ###Function to print N natural numbers using while loop
2
3  def natural(n):
4      i=1
5      while i <= n:
6          print(i,end=" ")
7          i=i+1
8      return
9  natural(20)

```

```

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

```

In [11]:

```

1  #Function to print the alternative values in a range
2  #[500,550] ->500 502 504.....550
3   #(500,550) ->501 503 505 .....549
4  #range(500,550) ->500 501 502 ..... 549
5  #All set based functions in python have start value inclusive end value is e
6
7
8
9  ###write a program to print alternate values
10 def alternatevalue(start,end):
11     for i in range(start,end+1,):
12         print(i,end=" ")
13     return
14 alternatevalue(500,525)
15

```

```

500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519
520 521 522 523 524 525

```

In [14]:

```

1  ###function to print reverse of a given range in the same
2  def reverserange(start,end):
3      for i in range(end,start,-1):
4          print(i,end=" ")
5      return
6  reverserange(500,525)
7

```

525 524 523 522 521 520 519 518 517 516 515 514 513 512 511 510 509 508 507 506  
505 504 503 502 501

In [20]:

```

1  #function to print the odd nos in a range
2
3  def odd(start,end):
4      for i in range(end,start,-1):
5          if(i%2!=0):
6              print(i,end=" ")
7      return
8
9  odd(1,10)

```

9 7 5 3

In [38]:

```

1  #Function to print sumof numbers in a range
2
3  def sumnumbers(start,end):
4      sum=0
5      for i in range(start,end+1):
6          sum=sum+i
7      return sum
8
9  sumnumbers(1,10)
10
11
12
13

```

Out[38]: 55

In [41]:

```

1  #Function to calculate the average of a given range
2
3  def average(start,end):
4      sum=0
5      c=0
6      for i in range(start,end+1):
7          sum=sum+i
8          c=c+1
9      return sum//c
10 average(1,5)
11
12
13
14

```

Out[41]: 3

```
In [10]: 1 ##### Function to print all numbers divisible by 6 and not a factor of 100 in
2 a=int(input("Enter the first range:"))
3 b=int(input("Enter the second range:"))
4 def divisible(a,b):
5     for i in range(a,b+1):
6         if i%6==0 and i%100!=0:
7             print(i,end=" ")
8     return
9 divisible(a,b)
```

Enter the first range:1

Enter the second range:100

6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96

```
In [17]: 1 ## Function to find the average of cubes of all even numbers in a given range
2 a=int(input("Enter the first range:"))
3 b=int(input("Enter the second range:"))
4 def avg(a,b):
5     sum=0
6     count=0
7     for i in range(a,b+1):
8         if i%2==0:
9             sum +=i ** 3
10            count +=1
11    return(sum/count)
12
13
14 avg(a,b)
```

Enter the first range:1

Enter the second range:3

Out[17]: 8.0

```
In [ ]: 1 ### Function to generate the List of factors of a given number
2 n=int(input("Enter the number:"))
3 def fact(n):
4     for i in range(1,n+1):
5         if(n%i==0):
6             print(i,end=" ")
7
8 fact(n)
9
```

```
In [12]: 1  ### Function to calculate of a given number
2  n=int(input("Enter the number:"))
3  def factorial(n):
4      f=1
5      for i in range(1,n+1):
6          f=f*i;
7      return f
8
9  factorial(n)
10
```

Enter the number:12

Out[12]: 479001600

```
In [2]: 1  ###Function to check if a given no is prime
2  n=int(input("Enter the number:"))
3  def prime(n):
4      c=0
5      for i in range(1,n+1):
6          if(n%i==0):
7              c=c+1
8      if(c==2):
9          print("prime")
10     else:
11         print("Not a prime")
12 prime(n)
13
```

Enter the number:5  
prime

```
In [8]: 1  i=1
2  x = int(input("Enter the number:"))
3  for k in range (1, (x+1), 1):
4      c=0
5      for j in range (1, (i+1), 1):
6          a = i%j
7          if (a==0):
8              c = c+1
9
10     if (c==2):
11         print (i)
12     else:
13         k = k-1
14
15     i=i+1
```

Enter the number:7  
2  
3  
5  
7



```

In [46]: 1  #####average of the given prime number
          2  n=int(input("Enter the first  number:"))
          3  n1=int(input("Enter the second number"))
          4  def prime(n,n1):
          5      sum=0
          6      c=0
          7      for i in range(n,n1):
          8
          9          for j in range(1,i):
10              if(i%j==0):
11                  c=c+1
12
13              if(c==2):
14                  sum=sum+j
15
16              return sum//c
17  prime(n,n1)
18

```

Enter the first number:1  
Enter the second number7

Out[46]: 0

```

In [3]: 1  #####average of the factorials
          2  n1=int(input("enter the first range:"))
          3  n2=int(input("enter the second range:"))
          4  def factorials(n1,n2):
          5      t=0
          6      count=0
          7      for i in range(n1,n2+1):
          8          sum=1
          9          for j in range(1,i+1):
10              sum=sum*j
11              count=count+1
12              t=t+sum
13          print(t/count)
14
15  factorials(n1,n2)
16
17

```

enter the first range:1  
enter the second range:5  
30.6

```

In [ ]: 1 ##### calculate number of days ina given time peroid
2 y1=int(input("Enter the first year:"))
3 y2=int(input("Enter the second year:"))
4 def leapyear(y1,y2):
5     for i in range(y1,y2+1):
6         if( y%400==0 or (y%100!=0 and y%4==0)):
7             print(366 * 24 * 60 * 60 end="")
8         else:
9             print(365* 24 * 60 * 60 end=" ")
10    return
11    leapyear(y1,y2)
12
13

```

```

In [51]: 1 #Function to generate all Leap years in a given time period
2 # 2000-2020 ->2000 2004 2008 2012 2016 2020
3
4 def isLeapYear(y):
5     if y%400==0 or (y%100!=0 and y%4==0):
6         return True
7     return False
8
9 isLeapYear(2016)
10
11
12
13 def generateLeapYears(startyear,endyear):
14     for year in range(startyear,endyear+1):
15         if isLeapYear(year):
16             print(year,end=" ")
17     return
18 generateLeapYears(1919,2019)
19

```

```

1920 1924 1928 1932 1936 1940 1944 1948 1952 1956 1960 1964 1968 1972 1976 1980
1984 1988 1992 1996 2000 2004 2008 2012 2016

```

In [2]:

```
1  # Function to a table
2  n=int(input("Enter the number:"))
3  lb=int(input("Enter the lower bound:"))
4  ub=int(input("Enter the upper bound:"))
5  def table(n,lb,ub):
6      for i in range(lb,ub+1):
7
8
9          print(n,"x",i,"=",n*i)
10
11      return
12
13
14  table(n,lb,ub)
```

Enter the number:20

Enter the lower bound:1

Enter the upper bound:20

```
20 x 1 = 20
20 x 2 = 40
20 x 3 = 60
20 x 4 = 80
20 x 5 = 100
20 x 6 = 120
20 x 7 = 140
20 x 8 = 160
20 x 9 = 180
20 x 10 = 200
20 x 11 = 220
20 x 12 = 240
20 x 13 = 260
20 x 14 = 280
20 x 15 = 300
20 x 16 = 320
20 x 17 = 340
20 x 18 = 360
20 x 19 = 380
20 x 20 = 400
```

In [ ]:

```
1  #Function to calculate number of hours for a given period
2  #number of hours(11,1975,3,1999)->2014504 or 2015248
3  #number of hours(5,2019,6,2019)->1464
4
```

In [ ]:

```
1  #Function to print all numbers
```