

Problem solving and Programming

Date 12th June 2019

- ObjectivesString Slicing
- Functions in python
- Basic Problem related to conditional statements using functions
- Python Data Structure - lists, Tuples and Dictionaries

```
### - Basic Operation on Data Structures  
### - Applying Data Structures to solve problems
```

In []:

1

String Slicing

```
In [48]: 1 s1 = "Python"
2 s1 # accessing the second character of a string
3 s1[3] # accessing the third character of a string
4 s1[len(s1)-1] # Accessing the length of the string and getting last character
5 s1[-1] # another way to accessing the last character that when we dont know
6 s1[-2] # accessing the penultimate character of a string
7
8 s1[0:2] # accessing the first two character of a string
9
10 s1[-2:] # accessing the last two character
11
12 s1[1:-1] # accessing the first and last character of a string
13
14 s1[len(s1)//2] #Accessing the middle character
15
16 s1[-1: :-1] # printing a string in reserve direction .
17
18 s1[::-2] # accessnig alternate character of a string in reverse of a string
19
20 s1[::2] # accessing alternate character from start to end of string
21
22 # print the even character
23
24 #s1[-1:-3:-1] # printing the last 3 character in recerse order
25
26
27
```

Out[48]: 'nhy'

```
In [ ]: 1
```

Functions

```
In [50]: 1 # Function to reverse a string
2 def reverseString(s):
3     return s[::-1]
4
5 reverseString("python")
```

Out[50]: 'nohtyp'

```
In [62]: 1 # Function to check if a string is a palidrome
2 # palidrome means checking that a string is same from frist letter to end an
3 def palidrome(s):
4     if s == s[::-1]:
5         return True
6     else:
7         return False
8
9 palidrome("xyx")
10
11
```

Out[62]: True

```
In [ ]: 1
```

```
In [87]: 1 # Function to check if a given year is a Leap year
2 def isLeapYear(year):
3     if year % 400 == 0 or (year % 100 != 0 and year % 4 == 0):
4         return True
5     return False
6
7 isLeapYear(2020)
8
```

Out[87]: True

```
In [ ]: 1
```

```
In [88]: 1 # Function to count the number of digits in a given number
2 def countDigits(n):
3     return len(str(n))
4
5 countDigits(347834)
```

Out[88]: 6

```
In [89]: 1 #Function to identify the greatest of 4 numbers
2
3 def greatest4(n1, n2, n3, n4):
4     if(n1 > n2 and n1 > n3 and n1 > n4):
5         return n1
6     elif n2 > n3 and n2 > n4
7
```

File "<ipython-input-89-10f004959653>", line 6

elif n2 > n3 and n2 > n4

^

SyntaxError: invalid syntax

```
In [ ]: 1
```

iteration

- for
- while

In [96]:

```

1  # function to print n natural number
2  def printNNaturalNumber(n):
3      for counter in range(1,n+1):
4          print(counter, end=" ")
5      return
6
7  printNNaturalNumber(30)
8

```

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 30

In [97]:

```

1  # function tp print count number using while loop
2  Number = int(input("Please Enter any Number: "))
3  Count = 0
4  while(Number > 0):
5      Number = Number // 10
6      Count = Count + 1
7
8  print("\n Number of Digits in a Given Number = %d" %Count)

```

Please Enter any Number: 12

Number of Digits in a Given Number = 2

In [98]:

```

1  # function to print N Natural numbers using while loop
2
3  def nNaturalNumbers(n):
4      counter = 1
5      while counter <= n:
6          print(counter, end = " ")
7          counter = counter + 1
8      return
9
10 nNaturalNumbers(9)

```

1 2 3 4 5 6 7 8 9

In [109]:

```

1  #function to print all numbers divisible by 6 and
2  # not a factor of 100 in a given range(lb,ub) inclusive
3
4  def divisible(n):
5      for i in range(1,n+1):
6          if(i % 6 == 0 and 100 % i != 0):
7              print(i, end = " ")
8
9  divisible(120)
10

```

6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120

```
In [122]: 1 # function to generate the list of factors for given numner
          2 # 12 -> 1,2,3,4,6,12
          3 def factors(n):
          4     print("the factors of given number ")
          5     for i in range(1,n+1):
          6         if n % i == 0:
          7             print(i)
          8
          9 factors(50)
```

the factors of given number

1
2
5
10
25
50

```
In [130]: 1 # function to calculate the factorial of a given number
          2 def fact(n):
          3     fact = 1
          4     # print(" the factorial number are", end = " ")
          5     for i in range(2,n+1):
          6         fact = fact * i
          7     return fact
          8 fact(10)
          9
         10
```

Out[130]: 1

```
In [2]: 1 # function to check if a given number is prime
        2 def prime(n):
        3     flag = True
        4     for i in range(2,n+1):
        5         if n % i == 0:
        6             flag = False
        7         return flag
        8     return flag
        9 prime(35)
```

Out[2]: False

```
In [113]: 1 # function to calculate the aaverage first N prime numbers
          2
```

```
In [1]: 1 # funtion to genreate all perfect numbers in given numbers
2 def perfect(n):
3     if fact(n) == n:
4         return True
5     return False
6 def generateperfect(lb,ub):
7     for i in range(lb,ub+1):
8         if perfect(i):
9             print(i,end=" ")
10    return
11 generateperfect(1, 1000)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-2a1915c17abd> in <module>
      9         print(i,end=" ")
     10     return
----> 11 generateperfect(1, 1000)

<ipython-input-1-2a1915c17abd> in generateperfect(lb, ub)
      6 def generateperfect(lb,ub):
      7     for i in range(lb,ub+1):
----> 8         if perfect(i):
      9             print(i,end=" ")
     10     return

<ipython-input-1-2a1915c17abd> in perfect(n)
      1 # funtion to genreate all perfect numbers in given numbers
      2 def perfect(n):
----> 3     if fact(n) == n:
      4         return True
      5     return False
```

NameError: name 'fact' is not defined

```
In [148]: 1 #function to find the avg of cubes of all even number in a given range(ld,ub)
2 def avgCubeEven(lb,ub):
3     sum = 0
4     count = 0
5     for i in range(lb,ub + 1):
6         if i % 2 == 0:
7             sum += i** 3
8             count += 1
9         return sum/count
10 avgCubeEven(1,3)
```

Out[148]: 8.0

```
In [ ]: 1
```

