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]:
```
s1 = "Python"
s1   # accessing the second character of a string
s1[3]  # accessing the  third charater of a string
s1[len(s1)-1] # Accessing the length of the string and getting last characte
s1[-1] # another way to accessing the last character that when we dont know
s1[-2] # accessing the penultimate character of a string

s1[0:2] # accessing the first two character  of  a string

s1[-2:] # accessing the last two character

s1[1:-1] # accessing the first and last character of a string

s1[len(s1)//2] #Accessing the middle character

s1[-1: :-1] # printing a string in reserve direction .

s1[::-2] # accessnig alternate character of a string in reverse of a string

s1[::2] # accessing alternate character from start to end of string

# print the even character

#s1[-1:-3:-1] # printing the last 3 character in recerse order


```

Out[48]: 'nhy'

In [ ]:
```
1
```

Functions

In [50]:
```
# Function to reverse a string
def reverseString(s):
    return s[::-1]

reverseString("python")
```

Out[50]: 'nohtyp'

In [62]:
```
# Function to check if a string is a palidrome
# palidrome means checking that a string is same from frist letter to end an
def palidrome(s):
    if s == s[::-1]:
        return True
    else:
        return False

palidrome("xyx")


```

Out[62]: True

In [ ]:
```

```

In [87]:
```
# Function to check if a given year is a leap year
def isLeapYear(year):
    if year % 400 == 0 or (year % 100 != 0 and year % 4 == 0):
        return True
    return False

isLeapYear(2020)

```

Out[87]: True

In [ ]:
```

```

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

countDigits(347834)
```

Out[88]: 6

In [89]:
```
#Function to identify the greatest of 4 numbers

def greatest4(n1, n2, n3, n4):
    if(n1 > n2 and n1 > n3 and n1 > n4):
        return n1
    elif n2 > n3 and n2 > n4

```

```
  File "<ipython-input-89-10f004959653>", line 6
    elif n2 > n3 and n2 > n4
                            ^
SyntaxError: invalid syntax
```

In [ ]:
```

```

iteration

- for
- while

In [96]:
```
# function to print  n  natural number
def printNNaturalNumber(n):
    for counter in range(1,n+1):
        print(counter, end=" ")
    return

printNNaturalNumber(30)

```

```
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
```

In [97]:
```
# function tp print count number using while loop
Number = int(input("Please Enter any Number: "))
Count = 0
while(Number > 0):
    Number = Number // 10
    Count = Count + 1

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]:
```
# function to print N Natural numbers using while loop

def nNaturalNumbers(n):
    counter = 1
    while counter <= n:
        print(counter, end = " ")
        counter =  counter + 1
    return

nNaturalNumbers(9)
```

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

In [109]:
```
#function to print all numbers divisible by 6 and
# not a factor of 100 in a given range(lb,ub) inclusive

def  divisible(n):
    for i in range(1,n+1):
        if(i % 6 == 0 and 100 % i != 0):
            print(i, end =" ")

divisible(120)

```

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

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

factors(50)
```

```
the factors of given number
1
2
5
10
25
50
```

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


```

Out[130]:
```
1
```

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

Out[2]:
```
False
```

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

```

In [1]:
```
# funtion to genreate all perfect numbers in given numbers
def perfect(n):
    if fact(n) == n:
        return True
    return False
def generateperfect(lb,ub):
    for  i in range(lb,ub+1):
        if perfect(i):
            print(i,end=" ")
    return
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]:
```
#function to find the avg of cubes of all even number in a given range(ld,ub
def avgCubeEven(lb,ub):
    sum = 0
    count = 0
    for i in range(lb,ub + 1):
        if i % 2 == 0:
            sum += i** 3
            count += 1
            return sum/count
avgCubeEven(1,3)
```

Out[148]: 8.0

In [ ]:
```

```