Functions conti..

- Arguments in functions
  - here we are able to call a function by these arguments
- Required Arguments
  - Arguments passes to a fucntion in correct positional order
- Keyword Arguments
  - identifies the arguments by parameter name
- Default Arguments
  - Assumes default value if value is not specified in function call
- Varaible length arguments
  - Process of a function for more arguments than you specified while defining the function

In [16]:
```
### required arguments

def hii(name):
    print('welcome to python online workshop by',name)
hii('APSSDC')
```

```
welcome to python online workshop by APSSDC
```

In [17]:
```
## Keyword Arguments

def details(number,branch):
    print(number)
    print(branch)

details(branch = 'ECE',number=663)
```

```
663
ECE
```

In [18]:
```
## Default arguments

def details(number,branch='ECE'):
    print(number)
    print(branch)
details(123,)
```

```
123
ECE
```

In [19]:
```
## Varaible length arguments

def funcv(n,*args): ##asterisk
    print(n)
    for i in args:
        print(i,end=' ')
funcv(2,'hi','welcome','to','python','online',123,456)
```

```
2
hi welcome to python online 123 456
```

In [20]:
```
def abcd(num):
    output = num*num
    return output
num = 10
res = abcd(num)
res
```

Out[20]: 100

In [24]:
```
## leap year check
def is_leap_year(year): ## 2000
    if year % 400 == 0 or (year % 100 != 0 and year % 4 == 0):
        return True
    return False
is_leap_year(2016)
```

Out[24]: True

In [25]:
```
## leap year in the given range

def leapyearsrange(startyear,endyear):  ## (2000, 2020)
    for year in range(startyear,endyear+1): ## for year in range(2000,2020)
        if is_leap_year(year): ## is_leap_year(2000)
            print(year,end=' ')

startyear = int(input('enter start year: '))
endyear = int(input('enter end year: '))
leapyearsrange(startyear,endyear)
```

```
enter start year: 2000
enter end year: 2020
2000 2004 2008 2012 2016 2020
```

In [ ]:
```

```

In [26]:
```
## Strings
s = 'python'
s[::-1]
```

Out[26]: 'nohtyp'

In [ ]:
```
[start:end:step]
[0::1]  - forward indexing -- positive indexing -- left to right

0 ---- end of the string
-1 -- end of the stings
```

In [27]:
```
s = 'python'
s[0]
```

Out[27]: 'p'

In [28]:
```
s[::2]
```

Out[28]: 'pto'

In [29]:
```
def count(string):
    a = n = s = 0
    for i in string:
        if(i.isdigit()):
            n = n+1
        elif(i.isalpha()):
            a = a+1
        else:
            s = s+1
    print('No of character: ',a)
    print('No of digits: ',n)
    print('No of sc: ',s)

string = input()
count(string)
```

```
abcd123@#
No of character:  4
No of digits:  3
No of sc:  2
```

In [32]:
```
s = "A P S S D C"
a = s.split()
print(a)
print("".join(a).capitalize())
```

```
['A', 'P', 'S', 'S', 'D', 'C']
Apssdc
```

Data Structures

- Data structures are the way of organizing or storing a data
- Data is nothing but information
- The main use data strucutes is we can access access data and we can workout on the data efficienly
- Types of DS
  - List
  - Tuple
  - Dictoinary
  - Set

- List
  - Collection of Hetrogenous data
  - Ordered and mutable(changeable)
  - Allows duplicate elements or items
  - represented by square brackets []

In [41]:
```
## declaration list

li = [123,'Nandini',9.8]
li
```

Out[41]: [123, 'Nandini', 9.8]

In [43]:
```
## ordered

print(li[0])
print(li[-1])
```

```
123
9.8
```

In [36]:
```
li[0]
```

Out[36]: 123

In [37]:
```
li[0] = 'Shyam'
```

In [38]:
```
li
```

Out[38]: ['Shyam', 'Nandini', 9.8]

In [39]:
```
li[0]
```

Out[39]: 'Shyam'

In [40]:
```
li1 = [1,2,2,4,6,3,6,7]
li1
```

Out[40]: [1, 2, 2, 4, 6, 3, 6, 7]

In [ ]: