# Functions

[ ]:
```
# There are two types of functions
#1. In-Built Functions
#2. user defined functions
```

[ ]:
```
# function should start with DEF
# function should have name associated withat
# funcion may or may not have parameters/arguments within paranthesis ()
```

[3]:
```
a = [25,34,24,33,46]
```

[5]:
```
sum(a)
```

[5]:
```
162
```

[11]:
```
a.count(34)
```

[11]:
```
1
```

[13]:
```
max(a)
```

[13]:
```
46
```

[15]:
```
b = "mahesh"
```

[19]:
```
b.upper(), b.capitalize(), b.strip()
```

[19]:
```
('MAHESH', 'Mahesh', 'mahesh')
```

[23]:
```
#2. user defined functions

def add_num(a,b):
    c=a+b
    print(c)
```

[25]:
```
add_num(20,30)
```

```
50
```

[27]:
```
add_num(35,45)
```

```
80
```

[29]:
```
add_num(44,64)
```

```
108
```

[31]:
```
print(20+30)
print(35+45)
print(44+64)
```

```
50
80
108
```

[2]:
```
def calculator(a,b):
    c= a+b
    d= a*b
    e= b/a
    print(c,d,e)
```

[4]:
```
calculator(35,45)
```

```
80 1575 1.2857142857142858
```

[37]:
```
def billing(price, qty):
    amount = price*qty
    print(amount)
```

[39]:
```
billing(25, 100)
```

```
2500
```

[41]:
```
billing(250,5)
```

```
1250
```

[43]:
```
def greeting():
    print("Good Morning")
```

[45]:
```
greeting()
```

```
Good Morning
```

[53]:
```
def greeting1(person_name):
    print("Good Morning", person_name)
    print("Had coffee", person_name)
```

[59]:
```
greeting1("Ashish")
```

```
Good Morning Ashish
Had coffee Ashish
```

[61]:
```
greeting1("Devadatta")
```

```
Good Morning Devadatta
Had coffee Devadatta
```

[63]:
```
greeting("Rohit")
```

```
Good Morning Rohit
Had coffee Rohit
```

[65]:
```
def greeting2(person1,person2,person3):
```

```
    print("Good Morning", person1,person2,person3)
    print("Had coffee", person1,person2,person3)
```

[69]:
```
greeting2("Ashish","Devadataa","Rohit")
```

```
Good Morning Ashish Devadataa Rohit
Had coffee Ashish Devadataa Rohit
```

[83]:
```
# local varialbe is a variable which is assinged inside the function
# Global variable is a variable which assinged outside the function

a=30 # Global variable
b=20 # Global Variable

def add_num(a,b):
    c=a+b # local varible
    print(c)
```

[87]:
```
a,b
```

[87]:
```
(30, 20)
```

[89]:
```
# There are four types of function arguments
#1.positonal arguments
#2.keyword arguments
#3.default agrugments
#4.variable length arguments
```

[95]:
```

#1.positonal arguments
def billing(price, qty):
    amount = price*qty
    print(amount)
```

[99]:
```
billing(25,100)
```

```
2500
```

[105]:
```
#2.keyword arguments
billing(qty=100,price=25)
```

```
2500
```

[111]:
```
#3.default agrugments

def student_admission(st_name,institue="Imax"):
    print(st_name,institue)


```

[115]:
```
student_admission("Ravi")
```

```
Ravi Imax
```

[117]:
```
student_admission("Ramesh")
```

```
Ramesh Imax
```

[119]:
```
student_admission("Rohan","raju academy")
```

```
Rohan raju academy
```

[137]:
```
#4.variable length arguments:

def imax_inst(*courses):
    for i in courses:
        print(i)


```

[139]:
```
imax_inst("python","data analytics","AI","Machine Learning","SAP","Data Engineering","Cloud Computing")
```

```
python
data analytics
AI
Machine Learning
SAP
Data Engineering
Cloud Computing
```

[141]:
```
#anaonymus fucntion(lambda function)
```

[143]:
```
def add_num(a,b):
    c=a+b
    print(c)
```

[151]:
```
x = lambda a,b:a+b
```

[153]:
```
x(20,30)
```

[153]:
```
50
```

[155]:
```
y = lambda a,b:a*b
```

[157]:
```
y(20,30)
```

[157]:
```
600
```

[159]:
```
# LIST Comprahension
```

[165]:
```
sq = []

for i in range(1,11):
    sq.append(i*2)
```

[167]:
```
sq
```

[167]:
```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

[169]:
```
type(sq)
```

[169]:
```
list
```

[173]:
```
sum(sq) , max(sq)
```

[173]:
```
(110, 20)
```

[175]:
```
i=1
s= [i*2 while i in range(1,11]
```

[177]:
```
s
```

[177]:
```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

[179]:
```
type(s)
```

[179]:
```
list
```

[181]:
```
sum(s),max(s),len(s),min(s)
```

[181]:
```
(110, 20, 10, 2)
```

[ ]:
```
#Numpy
#pandas
#matplotlib, seaborn
```

[ ]:
```

```