In [ ]:
```
# Function

- Function is a organized block of code

- It help on Resuable
```

In [ ]:
```
Syntax:

def <name_of_function>(parameter1, parameter2.... N-parameter):
    <cond block>
    return value
```

In [ ]:
```
Types Function:

    1- Built in function
    2- user defined function
    3- Anonymous Function
```

In [ ]:
```
# Built Function

print()
len()
input()
int()
float()
```

In [11]:
```
# User defined function

    - default argument
    - Keyword argument
    - variable length arugment
    - Required arugment
```

In [ ]:
```
# ANonymous function

# A function has no name we called as ANonymous function

eg. lambda function
```

In [1]:
```
def greet():

    print("welcome to python")
```

In [2]:
```
# calling the function
greet()
```

```
welcome to python
```

In [6]:
```
def add_two_numbers():
    val1 =10
    val2 = 20
    total = val1 + val2
    return total
```

In [7]:
```
add_two_numbers()
```

Out[7]:
```
30
```

In [8]:
```
# Required argument
def greet(name):

    return name
```

In [12]:
```
#
greet("John")
```

Out[12]:
```
'John'
```

In [13]:
```
def add_two_numbers(num1,num2):
    total = num1 + num2
    return total
```

In [17]:
```
add_two_numbers(10,20)
```

Out[17]:
```
30
```

In [24]:
```
def square(n):
    return n * n
```

In [25]:
```
square(5)
```

Out[25]:
```
25
```

In [33]:
```
### default argument

def add_two_names(name1, name2="bob"):
    return "name 1 is " + name1 + "============ name2 is "+name2
```

In [34]:
```
add_two_names("john")
```

Out[34]:
```
'name 1 is john============ name2 is bob'
```

In [35]:
```
add_two_names("john","Amit")
```

Out[35]:
```
'name 1 is john============ name2 is Amit'
```

In [40]:
```
## Keyword argument


def find_the_age(name,age):
    return f"name {name} ========== age {age}"
```

In [43]:
```
find_the_age(name="Henary",age = 20)
```

Out[43]:
```
'name Henary ========== age 20'
```

In [50]:
```
# Variable length argument

def names(*names):
    for name in names:
        print(name)
```

In [51]:
```
names("john","Bob","Ken","william","lords","God","tom")
```

```
john
Bob
Ken
william
lords
God
tom
```

In [ ]:
```
# what is the difference between *args vs **kwargs?
```

In [70]:
```

def names(**names):
    for name in names.items():
        print(name)
```

In [71]:
```
names(name1 = "john",name2="Bob",name3="william",name4="tom")
```

```
('name1', 'john')
('name2', 'Bob')
('name3', 'william')
('name4', 'tom')
```

In [ ]:
```
## Anonymous Function
```

In [72]:
```
def play():
    print("i can play cricket")
```

In [73]:
```
play()
```

```
i can play cricket
```

In [76]:
```
def square(x):
    print(x * x)
```

In [77]:
```
square(10)
```

```
100
```

In [83]:
```
s = lambda x : x * x
```

In [85]:
```
s(10)
```

Out[85]:
```
100
```

In [94]:
```
x = lambda x,y,z : x + y+z
```

In [95]:
```
x(10,20,30)
```

Out[95]:
```
60
```

In [96]:
```
l2 =lambda :"hello world"
```

In [98]:
```
l2()
```

Out[98]:
```
'hello world'
```

In [ ]:
```
## Excercise

- what is difference bwteen local vs global
- what is the *args vs **args
- Find the even and odd number using function?
```