function in python

In [1]:
```
def greet(): # function name is greet()
    print('Hello')
    print('Good Morning Team')
```

In [ ]:
```
# no output becoz, we didn't call the function
```

In [5]:
```
def greet(): # function name is greet()
    print('Hello')
    print('Good Morning Team')
greet()    #function calling
```

```
Hello
Good Morning Team
```

In [1]:
```
def greet():
    print('hello')
    print('good morning team')
greet()

def greet():
    print('hello')
    print('good morning team')
greet()
```

```
hello
good morning team
hello
good morning team
```

In [3]:
```
def greet(): # function name is greet()
    print('Hello')
    print('Good Morning Team')
greet()

print()

def greet(): # function name is greet()
    print('Hello')
    print('Good Morning Team')
greet()
```

```
Hello
Good Morning Team

Hello
Good Morning Team
```

In [ ]:
```
#above code requires more memory, so memory gets wasted
```

In [4]:
```
def greet(): # function name is greet()
    print('Hello')
    print('Good Morning Team')
greet()
```

```
print('******************')
greet()
```

```
Hello
Good Morning Team
******************
Hello
Good Morning Team
```

In [2]:

```
def greet():  # declare function without argument
    print('hello')
    print('good morning team')
greet()
print('***********')
greet()
print('***********')
greet() # function calling  with out argument)
```

```
hello
good morning team
***********
hello
good morning team
***********
hello
good morning team
```

In [3]:

```
# function without argument

def greet():
    print('hello')
    print('good morning team')
greet()
```

```
hello
good morning team
```

In [4]:

```
# function with argument

def add(x,y):
    c=x+y
    print(c)

add(2,7)
```

```
9
```

In [5]:

```
# function with argument

def add(x,y):
    c=x+y
    return c

add(2,7)
```

Out[5]:

```
9
```

In [6]:

```
# function with argument

def add(x,y):
    c=x+y
    return c

add(5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[6], line 7
      4     c=x+y
      5     return c
----> 7 add(5)

TypeError: add() missing 1 required positional argument: 'y'
```

In [7]:

```
# function with argument

def add(x,y):
    c=x+y
    return c

add(5,6,7)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 7
      4     c=x+y
      5     return c
----> 7 add(5,6,7)

TypeError: add() takes 2 positional arguments but 3 were given
```

In [8]:

```
# function with argument

def add(x,y,z):
    c=x+y
    return c

add(5,6,7)
```

Out[8]:

```
11
```

In [9]:

```
# function with argument

def add(x,y,z):
    c=x+y+z+m
    return c

add(5,6,7)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 7
      4     c=x+y+z+m
      5     return c
----> 7 add(5,6,7)

Cell In[9], line 4, in add(x, y, z)
      3 def add(x,y,z):
----> 4     c=x+y+z+m
      5     return c

NameError: name 'm' is not defined
```

In [10]:

```
# function with argument

def add(x,y,z,n):
    c = x+y+z+m
    return c

add(2,4,5,8)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[10], line 7
      4     c = x+y+z+m
      5     return c
----> 7 add(2,4,5,8)

Cell In[10], line 4, in add(x, y, z, n)
      3 def add(x,y,z,n):
----> 4     c = x+y+z+m
      5     return c

NameError: name 'm' is not defined
```

In [11]:

```
# function with argument

def add(x,y,z,n):
    c = x+y+z+n
    return c

add(1,2,3,4)
```

Out[11]:

```
10
```

In [12]:

```
def greet():
    print('hello')
    print('good morning team')
greet()

def add(x,y):
    c = x+y
    return c

add(2,3)
```

```
hello
good morning team
```

Out[12]: 5

In [13]:

```
def greet():
    print('hello')
    print('good morning team')

def add(x,y):
    c = x+y
    return c

def sub(x,y):
    d = x-y
    return d


greet()
print(add(2,3))
print(sub(2,3))
```

```
hello
good morning team
5
-1
```

In [14]:

```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result = add_sub(2,4)
print(result)
print(type(result))
```

```
(6, -2)
<class 'tuple'>
```

In [15]:

```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result, result1 = add_sub(1,2)

print(result)
print(result1)
print(type(result))
```

```
3
-1
<class 'int'>
```

In [16]:

```
def add_sub_mul(x,y):
    c= x+y
    d= x-y
    e= x*y
    return c, d, e

add, sub, mul = add_sub_mul(2,3)

add
```

```
sub
mul
```

Out[16]:

```
6
```

update

In [17]:

```
def update():
    x = 2
    print(x)
update()
```

```
2
```

In [18]:

```
def update(): #update function take the value from the user
    x = 2
    print(x)

update(2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 5
      2     x = 2
      3     print(x)
----> 5 update(2)

TypeError: update() takes 0 positional arguments but 1 was given
```

In [19]:

```
def update(x): #update function take the value from the user
    x = 2
    return x

update(10)
```

Out[19]:

```
2
```

In [20]:

```
def update(x):
    x = 2
    return x

a = 10
update(a)
print(a)
```

```
10
```

In [21]:

```
def update(x):
    x = 2
    return x

x = 10
update(x)
print(x)
```

```
10
```

In [31]:

```
def update(x):
    x = 2
    return x
x=10
print(x)
```

```
10
```

In [32]:

```
def update(x):
    x = 2
    return x
x=10
print(x)
update(x)
```

```
10
```

Out[32]:

```
2
```

In [33]:

```
def update(x):
    x = 2
    return x
x=10
update(x)
print(x)
```

```
10
```

In [ ]: