In [1]:
```
def ratna():
    print("Hello,Welcome")
ratna()
```

```
Hello,Welcome
```

In [3]:
```
# No parameter , no return
def add():
    a=10
    b=20
    print(a+b)
add()
```

```
30
```

In [5]:
```
# parameter and no return
def add(a,b):
    print(a+b)
add(5,7)
```

```
12
```

In [7]:
```
# no parameter but return
def get_value():
    return 100
x = get_value()
print(x)
```

```
100
```

In [9]:
```
# parameter and return
def multiply(a,b):
    return a*b
result = multiply(4,5)
print(result)
```

```
20
```

In [13]:
```
# function default parameter
def ratna(name="Student"):

    print("Hello",name)
```

```
ratna()
ratna("Smruti")
```

```
Hello Student
Hello Smruti
```

In [15]:
```
x = 10 #global
def demo():
    y = 20  #local
    print(x)
    print(y)
demo()
```

```
10
20
```

In [17]:
```
def ratna():
    print('hello')
    print('good morning')
```

In [19]:
```
def ratna():
    print('hello')
    print('good morning')
ratna() # im just calling fuc
```

```
hello
good morning
```

In [21]:
```
def ratna():
    print('hello')
    print('good morning')
ratna()  # call multiple times
def ratna():
    print('hello')
    print('good morning')
ratna()
```

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

In [23]:
```
def ratna():
    print('hello')
    print("good")
ratna()
ratna()
```

```
hello
good
hello
good
```

In [25]:
```
def ratna():
    print("hey")
    print("how are you")
ratna()
print()
ratna()
```

```
hey
how are you

hey
how are you
```

In [27]:
```
def add(x,y):
    c= x+y
    print(c)
add(5,6)
```

```
11
```

In [29]:
```
def add(x,y,z):
    c=x+y+z
    print(c)
add(1,4,5)
```

```
10
```

In [33]:
```
def ratna():
    print("hello")
    print("how have u been")
ratna()
def add(x,y):
```

```
    c= x+y
    print(c)
add(5,4)
# you can create  multiple fuc and call many times
```

```
hello
how have u been
9
```

In [35]:
```
def ratna():
    print("hello")
    print("how have u been")

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

def sub(x,y,z):
    d= x-y-z
    print(d)
ratna()
add(5,4)
sub(10,2,5)
```

```
hello
how have u been
9
3
```

In [37]:
```
def ratna():
    print("hello")
    print("ratna calling u")
ratna()
def add(x,y):
    c=x+y
    return c
result = add(4,5)
print(result)
```

```
hello
ratna calling u
9
```

In [41]:
```
def add_sub(x,y):
    c = x+y
    d= x-y
    return c,d
add_sub(4,5)
```

Out[41]: (9, -1)

In [43]:
```
def add_sub(x,y): # what if i want to return 2 values add_sub & i want to return 2 values & function can accept multi
    c= x+y
    d= x-y
    return c, d

result = add_sub(4,5)

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

```
(9, -1)
<class 'tuple'>
```

In [45]:
```
def update():
    x=8
    print(x)
update()
```

```
8
```

In [47]:
```
def update():
    x=8
    print(x)
update(8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[47], line 4
      2     x=8
      3     print(x)
----> 4 update(8)

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

In [49]:
```
def update(x):
    x=8
    print(x)
update(100)
```

```
8
```

In [51]:
```
def update(x):
    x= 8
    print(x)
update(10)
```

```
8
```

In [53]:
```
def update(x):
    x = 8
    print(x)

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

```
8
10
```

In [57]:
```
def update(x):
    x = 8
    #print(x)

a = 10
update(a)
print(a) # this print will update 8 to 5
```

```
10
```

In [ ]:
```

```