function in python

In [1]:
```
def hello():
    print('good evening')
hello()
```

```
good evening
```

In [2]:
```
def hello():
    print('good evening')
hello()

def hello():
    print('good evening')
hello()

def hello():
    print('good evening')
hello()
```

```
good evening
good evening
good evening
```

In [3]:
```
def hello():
    print('good evening')
hello()

hello()

hello()

hello()
```

```
good evening
good evening
good evening
good evening
```

In [4]:
```
def add (x,y):

    c=x+y
    print(c)
add(5,6)
```

```
11
```

In [5]:
```
def add (x,y,z):

    c=x+y +z
    print(c)
add(5,6,7)
```

```
18
```

In [6]:
```
def hello():
    print('good evening')
hello()
```

```
def add (x,y):

    c=x+y
    print(c)
add(5,6)
```

```
good evening
11
```

In [7]:
```
def hello():
    print('good evening')
def add (x,y):

    c=x+y
    print(c)
hello()
add(5,6)
```

```
good evening
11
```

In [8]:
```
def greet():
    print('hello')
    print('good noon')

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

def sub(x,y,z):
    d = x-y-z
    print(d)

greet()
add(5,4)
sub(10,2,4)
```

```
hello
good noon
9
4
```

In [9]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c,d

add_sub(4,5)
```

Out[9]: (9, -1)

In [10]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result = add_sub(4,5)
print(result)
print(type(result))
```

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

In [11]:
```python
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result, result1 = add_sub(4,5)

print(result)
print(result1)

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

```
9
-1
<class 'int'>
<class 'int'>
```

In [12]:
```python
def add_sub_mul(x,y):
    c= x+y
    d= x-y
    e = x*y
    return c, d, e

result, result1, result2 = add_sub_mul(4,5)

print(result)
print(result1)
print(result2)

print(type(result))
print(type(result1))
print(type(result2))
```

```
9
-1
20
<class 'int'>
<class 'int'>
<class 'int'>
```

In [13]:
```python
def update():
    x = 8
    print(x)
update()
```

```
8
```

In [14]:
```python
def add(a,b): # a & b are called as formal argument
    c = a+b
    print(c)
add(5,6)
```

```
11
```

In [15]:
```python
# positional argument

def person(name,age):
    print(name)
```

```
    print(age)

person('nit',22)
```

```
nit
22
```

In [16]:
```
def person(name,age):
    print(name)
    print(age)

person(22,'nit')
```

```
22
nit
```

In [17]:
```
def person(name,age):
    print(name)
    print(age-1)

person(22,'nit')
```

```
22
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 5
      2     print(name)
      3     print(age-1)
----> 5 person(22,'nit')

Cell In[17], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age-1)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [18]:
```
def person(name,age):
    print(name)
    print(age+2)

person('nit',22)
```

```
nit
24
```

In [19]:
```
def person(name,age):
    print(name)
    print(age)

person('nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[19], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit')

TypeError: person() missing 1 required positional argument: 'age'
```

In [20]:
```
def person(name,age):
    print(name)
    print(age+1)

person(22)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(22)

TypeError: person() missing 1 required positional argument: 'age'
```

In [21]:
```
def person(name):

    print(name)
    print(age+1)

person('nit', 22)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 6
      3     print(name)
      4     print(age+1)
----> 6 person('nit', 22)

TypeError: person() takes 1 positional argument but 2 were given
```

keyword argument

In [22]:
```
def person(name,age):

    print(name)
    print(age+1)

person('nit', 22)
```

```
nit
23
```

In [23]:
```
def person(name, age):

    print(name)
    print(age+1)

person(22, 'nit')
```

```
22
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[23], line 6
      3     print(name)
      4     print(age+1)
----> 6 person(22, 'nit')

Cell In[23], line 4, in person(name, age)
      1 def person(name, age):
      3     print(name)
----> 4     print(age+1)

TypeError: can only concatenate str (not "int") to str
```

In [24]:
```
def person(name, age):

    print(name)
    print(age+1)

person(age = 22, name = 'nit', phone = 879678)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[24], line 6
      3     print(name)
      4     print(age+1)
----> 6 person(age = 22, name = 'nit', phone = 879678)

TypeError: person() got an unexpected keyword argument 'phone'
```

In [25]:
```
def person(name, age, phone):

    print(name)
    print(age+1)
    print(phone)

person(age = 22, name = 'nit', phone = 879678)
```

```
nit
23
879678
```

In [26]:
```
def person(name, age=18):

    print(name)
    print(age)

person('nit')
```

```
nit
18
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: