Python functions

June 5, 2025

[1]:
```
print("hello world!")
```

```
hello world!
```

[3]:
```
# Rather than using print everytime you can use only
# greet to print hello world which makes the task easy.

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

[5]:
```
greet()
```

```
good evening
```

[7]:
```
greet()
```

```
good evening
```

[9]:
```
def greet():
    print('good evening')

greet()
```

```
good evening
```

[11]:
```
# to print 3 times
def greet():
    print('good evening')

greet()
greet()
greet()
```

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

[1]:
```
def add(x,y):
    c=x+y
    print(c)
```

```
add(5,6)
```

```
11
```

[3]:
```
def add(x):
    c=x+y
    print(c)

add(5,6)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[3], line 5
      2     c=x+y
      3     print(c)
----> 5 add(5,6)

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

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

add(5,6,7)
```

```
18
```

[7]:
```
def greet():
    print('good evening')
greet()
print()

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

add(5,6)
```

```
good evening

11
```

[9]:
```
# standard way to write

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

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

greet()
print()
add(5,6)
```

```
good evening

11
```

[11]:
```
def greet():
    print('good evening')

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

def sub(x,y):
    c=x-y
    print(c)

greet()
add(5,6)
sub(5,6)
```

```
good evening
11
-1
```

[15]:
```
# print can be replaced with return

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

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

add(20,10)
sub(20,10)
```

[15]:
```
10
```

[17]:
```
def add(x,y):
    c=x+y
    return c

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

print(add(20,10))
print(sub(20,10))
```

```
30
10
```

[19]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d

print(add_sub(20,10))
```

```
(30, 10)
```

[21]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d

print(add_sub(20,10))
result=add_sub(20,10)
print(type(result))
```

```
(30, 10)
<class 'tuple'>
```

[23]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d

result1,result2=add_sub(20,10)

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

```
<class 'int'>
<class 'int'>
30
10
```

[25]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d

result=add_sub(10,20)
result1=add_sub(10,20)
print(result)
print(result2)
print(type(result))
print(type(result))
print(type(result1))
```

```
(30, -10)
10
<class 'tuple'>
<class 'tuple'>
<class 'tuple'>
```

1 FUNCTION AHD TWO MAIN CONCEPT - WITHOUT ARGUMENT, WITH ARGUMENT

[ ]:
```
#WITHOUT ARGU
# WITH ARGU

- this is define in 2 part
1. formal arg
2. actual arg
 this is devided into 4 parts

Positional arg
keyword
default
variable
```

[28]:
```
def update(x):
    x=8
    return x
update(10)
```

[28]:
```
8
```

[30]:
```
def update(x):
    x=8
    return x
a=10
```

```
print(update(a))
print(a)
```

```
8
10
```

[32]:
```
def add(x,y): # x& y is called formal argument
    c=x+y
    return c

add(4,5) # 4 and 5 is called actual arguments
```

[32]:
```
9
```

2 POSITIONAL ARGUMENT

[35]:
```
def add(x,y): # x & y is called formal argument
    c=x+y
    return c

add(4,5) # 4&5 is called actual arguments
```

[35]:
```
9
```

[39]:
```
# positional arguments

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

add(4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[39], line 7
      4     c=x+y
      5     return C
----> 7 add(4)

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

[41]:
```
def add(x):  # x& y is called formal argument
    c=x+y
    return c
add(4,5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[41], line 4
      2     c=x+y
      3     return c
----> 4 add(4,5)

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

[43]:
```
def person(name,age):
    print(name)
    print(age)

person('nit',22)
```

```
nit
22
```

[45]:
```
def person(name,age):
    print(name)
    print(age)

person(22,'nit')
```

```
22
nit
```

[47]:
```
def person(name,age):
    print(name)
    print(age+1)

person(22,'nit')
```

```
22
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[47], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(22,'nit')

Cell In[47], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age+1)
```

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

3 KEYWORD ARGUMENT

[50]:
```
def person(name,age):
    print(name)
    print(age+1)

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

```
nit
23
```

[52]:
```
def person(name,age,salary):
    print(name)
    print(age+1)

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[52], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(age=22,name='nit')

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

#Default argument

[55]:
```
def person(name, age, age2):
    print(name)
    print(age)
    print(age2)

person(age=20, name ='nit', age2 = 21)

# this is called keyword argument
```

```
nit
20
21
```

[57]:
```
def person(name,age):
    print(name)
    print(age)
```

```
person('nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[57], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit')

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

[59]:
```
def person(name,age=18):
    print(name)
    print(age)

person('nit')
```

```
nit
18
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: