

15th March, 2023

7:28 am

## Functions

### Syntax

```
def add_three(x):  
    """Increase x by 3."""  
    x = x + 3  
    return x
```

Annotations:  
- **def**: keyword  
- **add\_three**: function name  
- **(x)**: input parameter  
- **"""**: doc string  
- **x = x + 3**: operation  
- **return x**: output

Whenever the function "add\_three" is called, 3 is added to the "x", input parameter.

1. Operations can be performed on functions:  
 $x = 2$   
 $y = \text{add\_three}(x) + \text{add\_three}(x) - \text{add\_three}(x)$

2. A function can have as many input and output as needed.

\* args  $\rightarrow$  positional arguments  
\* kwargs  $\rightarrow$  keyword arguments  
 $\rightarrow$  Allows one to pass multiple inputs

Example:  $\text{def my\_func}(*\text{args}, **\text{kwargs})$

Annotations:  
- **\*args**: positional arguments  
- **\*\*kwargs**: keyword arguments

```
for arg in args:  
    print(arg)  
for key, value in kwargs.items():  
    print(f"key: {key} value: {value}")
```

Annotations:  
- **key: {key}**: positional argument  
- **value: {value}**: keyword argument

$\text{my\_func}(1, 2, \text{"hello"}, a=1, b=2, c=\text{"hello"})$

Annotations:  
- **1, 2, "hello"**: positional arguments  
- **a=1, b=2, c="hello"**: keyword arguments

Output:

```
1  
2  
hello  
a = 1  
b = 2  
c = "hello"
```