

Web Application Development using Python

Introduction to Functions

Prepared by George Khoury



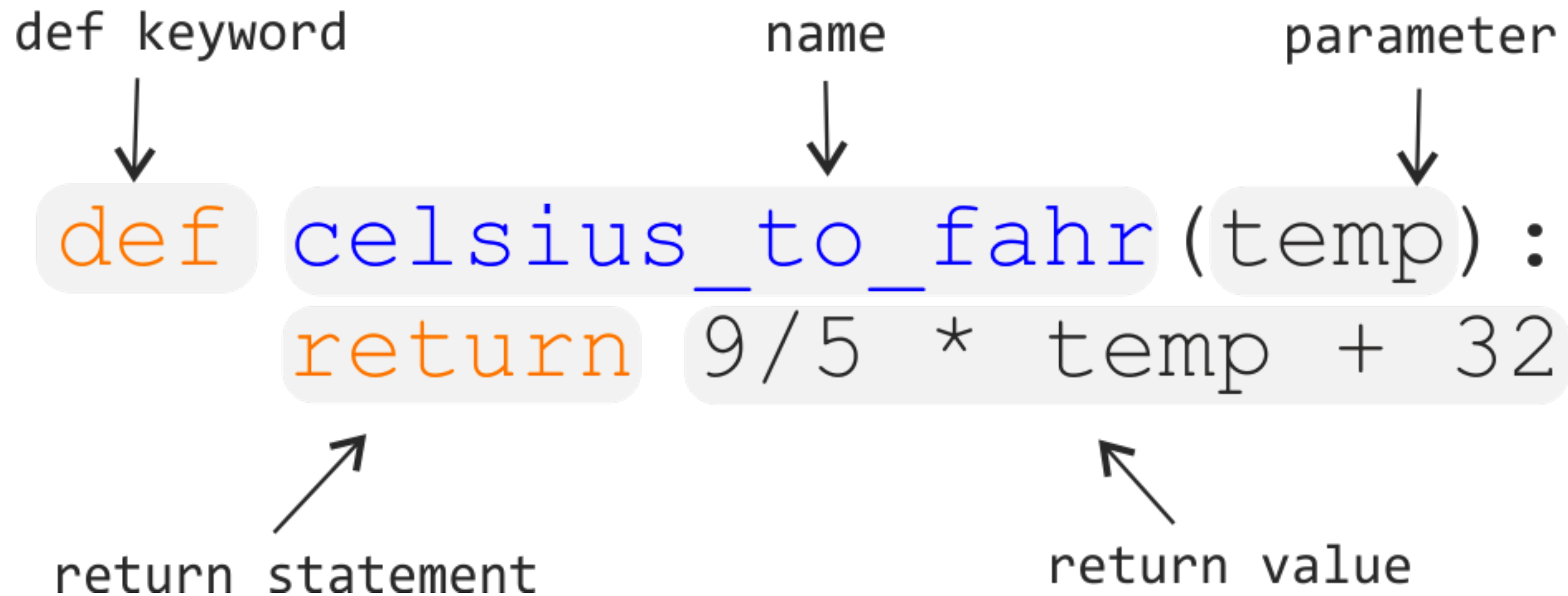
Outline

- Basic structure
- Parameters
- Return value
- Functions vs. Methods



Basic Structure

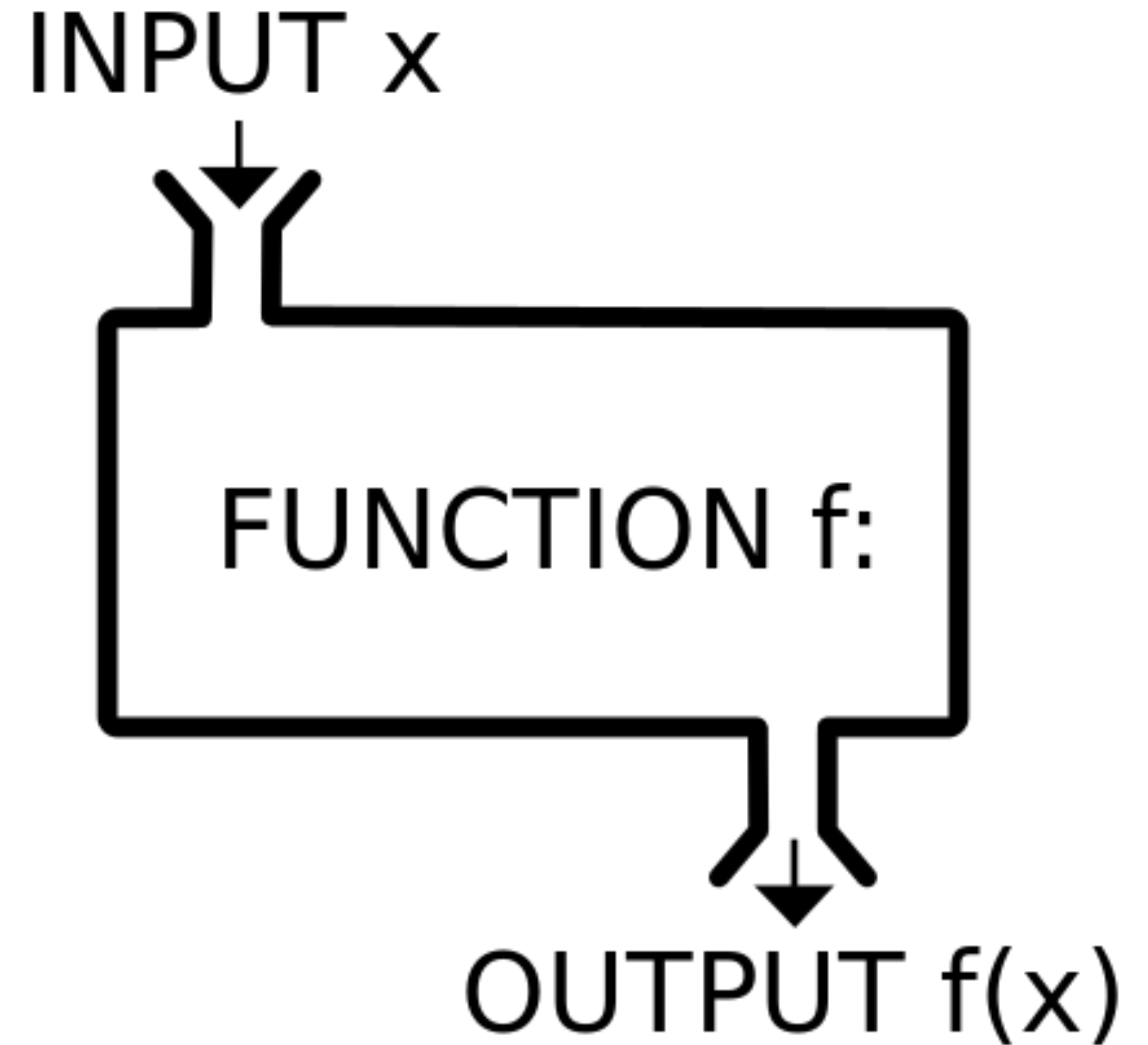
W2/S1/ex0.py



Basic Structure

W2/S1/ex{0,1}.py

- What is a **function signature**?
- What is a **function parameter**?
- What is a **function body**?
- What is a **function call**?



Function vs. Methods

W2/S1/ex{0,1}.py

- **Functions**

- A function is a block of code to carry out a specific task, will contain its own scope and is called by name.
- All functions may contain zero(no) arguments or more than one arguments.
- On exit, a function can or can not return one or more values.

```
def my_function():  
    pass
```

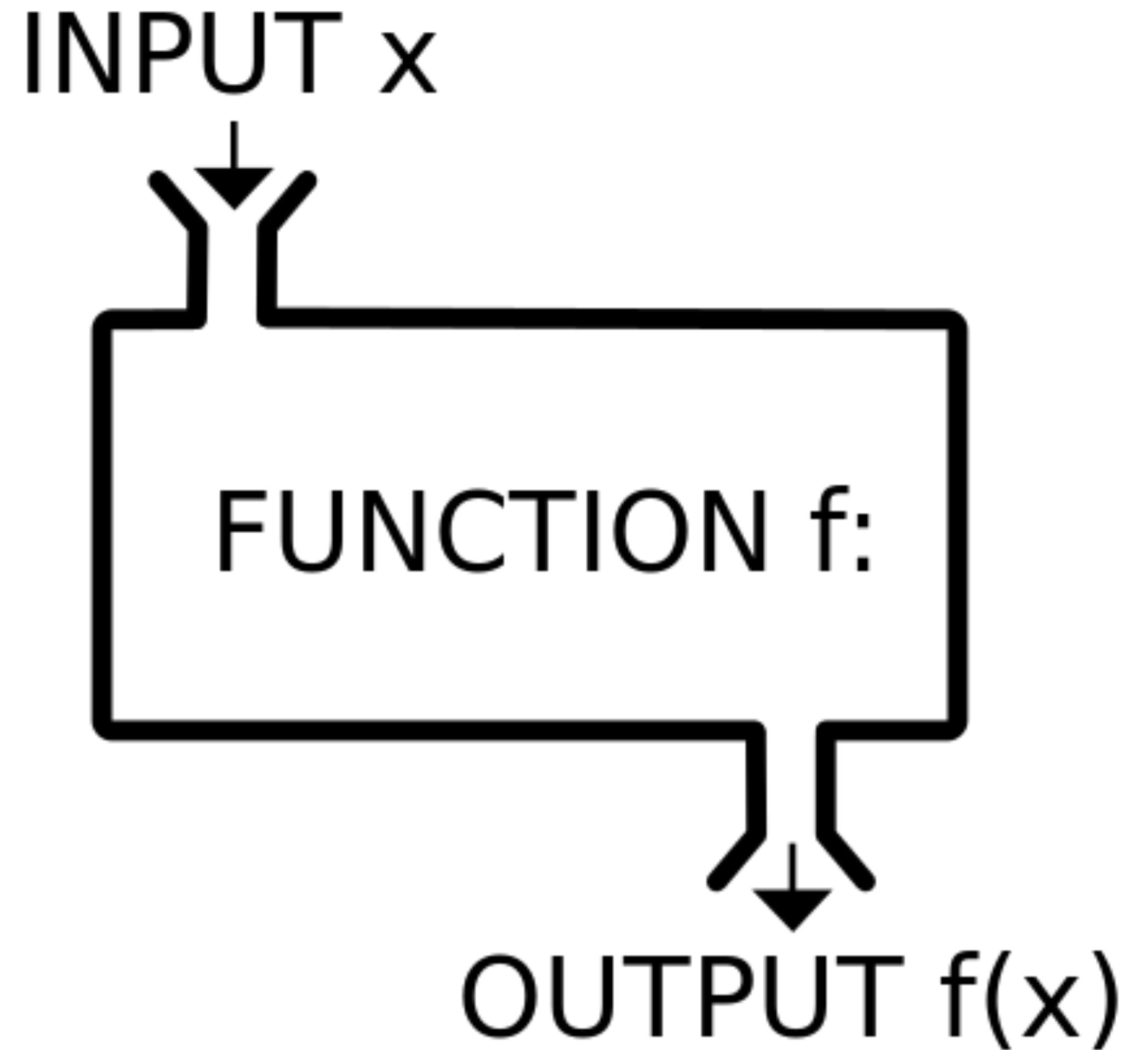
- **Methods**

- A method in python is somewhat similar to a function, except it is associated with object/classes.
- Methods in python are very similar to functions except for two major differences.
 - The method is implicitly used for an object for which it is called.
 - The method is accessible to data that is contained within the class.

```
class Test:  
    def my_method(self):  
        pass
```

Functions

- **Built-in Functions**
 - `len()`, `range()`, `min()`, etc.
 - `int()`, `str()`, `list()`
- **User-defined Functions**
 - `say_hello()`, `add()`, `minimum()`
- Anonymous Functions (`lambda`)



Resources

- <https://docs.python.org/3.7/tutorial/controlflow.html#defining-functions>
- <https://docs.python.org/3.7/tutorial/controlflow.html#more-on-defining-functions>