Self-defined functions

Week 5

How to define a function?

```
def function_name(parameters):
    # Function body
    # Code to perform the tasks
    # ...
    return result
```

- **def:** This keyword is used to define a function.
- **function_name:** This is the identifier for the function.
- When you *call* the function, you use the name of the function.
- Name should be in lowercase, with words separated by underscores (whitespaces in the name are not possible – as in the case of variables).
- Name should describe what the function does

How to define a function?

```python
def function_name(parameters):
    # Function body
    # Code to perform the task
    # ...
    return result
```

- **Parameters:** These are inputs to the function.
- Parameters are enclosed in parentheses and **separated by commas**.
- Parameters are **input values** to the function.
- The **function header** ends with a **colon**.

How to define a function?

```
def function_name(parameters):
    # Function body
    # Code to perform the task
    # ...
    return result
```

- **Function body:** Function is a "program inside a program".
- The function body contains the code that defines the behavior of the function.
- Note that the whole function body is ***indented***.

How to define a function?

```
def function_name(parameters):
    # Function body
    # Code to perform the task
    # ...
    return result
```

- **return:** The return statement specifies the value to be returned to the caller.
- The ***function call*** gets the value that is returned from the function.

return() ≠ print()

- A **return** statement is used to **end the execution of the function.**

- The execution of the function stops to the *first* encountered return while running.

- There can be **several** `returns` in one function – for example, in different branches of `if-else` statements.

- `return() ≠ print()`

- If you do not anything with the returned value, it gets lost!

- If you write a function, it should (usually) have a return statement!

How to call a function?

- Suppose you have a function `fun(p)` with one parameter p.

- We you call this function, you give a value to p.

- Function returns a value, so you need to do something with that returned value.

- For example,
  ```
  x = 66
  y = fun(x)
  ```