

Functions

Functions allow you to define new statements and what they do, with arguments passed through to the function.

The main purpose is to write a section of code once and then use it multiple times, rather than writing it again every time you need it.

Example:

```
def Greet(name):  
    print('Hello', name)
```

Notes:

- You can call functions anything you want, though it makes sense to name them descriptively, for clarity.
- The same goes for function arguments (e.g. 'name' in the above example).
- `print()` is a pre-defined function in Python.

Function Definitions

The code in the above example is the function definition for `Greet()`.

In function definitions in Python, the code that makes up the function is indented. It's important that it's all indented the same amount (standard Python convention is 4 spaces, but it can be any number of spaces).

When the interpreter comes to non-indented text again, that's how it recognises the end of the function. If there is no more text (because it's the end of the file), the interpreter will use the end of the file to recognise the end of the program.

Parameters

Parameters or arguments are how you pass data into functions.

Each time you call the function, an assignment statement is performed (e.g. for the above example, `name = 'Ann'`).

You can have as many parameters as you like in Python (your hardware will eventually stop you, but not for a very long time). Parameters are separated by commas.

- It is possible to specify optional parameters (with default values), as well as variable numbers of parameters—these will be dealt with later.

The *return* Statement

Example:

```
return EXPRESSION
```

The *return* statement does 3 things:

1. Calculates the value of EXPRESSION.
2. Marks the result as the answer from the function.
3. Exits from the function.

Not all functions return values (e.g. the `Greet()` function above, which just prints values).

Modules in IDLE

Assignments for this course are submitted as Python modules, files that contain Python code and end in '.py'.

These can be run within IDLE (the editor we use) in order to use the functions within. Many modules can be loaded at once, though this is normally done through the use of *import* statements, which we will see later.

Reserved Words / Keywords

Note that *def* and *return* are reserved words. This means that, since they have special functions in Python, they can't be used for anything else. You will get an error if you try to run code where you call something one of these words.

Built-in Functions

`print()` is a built-in function, rather than a reserved word. This was a change from Python 2 to Python 3. This allows you to, for example, pass `print()` as an argument to another function.

Built-in functions are conceptually no different from functions you might write yourself, except that they have already been written by someone else.