

Modules in Python

Before getting too far afield, it's import to understand Python modules – because we will be importing and using them so often.

They are quite simple: **they are merely Python files — i.e., files that end in `.py` and available to your program.**

In fact, any time you create a new Python (`.py`) file alongside your `main.py` file, you've created a **module** from which you can **import** procedures or variables or whatever. Take a quick look [at the Python docs](#) for more. Note this excerpt:

Modules in Python

Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module; definitions from a module can be imported into other modules or into the main module (the collection of variables that you have access to in a script executed at the top level and in calculator mode).

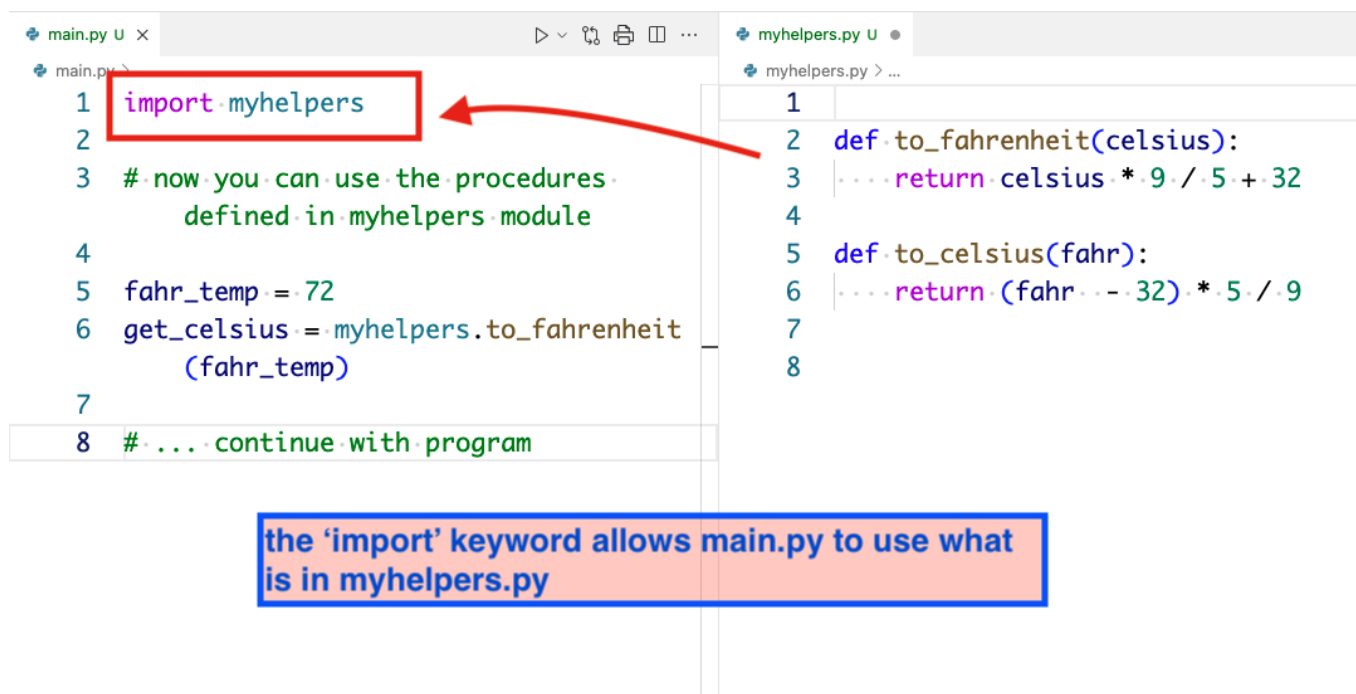
A module is a file containing Python definitions and statements. The file name is the module name with the suffix `.py` appended. Within a module, the module's name (as a string) is available as the value of the global variable **`name`**.

If that sounds a little too close to gobbleygook, you're not alone. It is ***always*** best to see a simple example. So here goes:

Below are two Python files:

- `main.py`
- `myhelpers.py`

The file or *module* `myhelpers.py` contains two procedures that are now available to be used (or ***called***) in the `main.py` file because `main.py` uses the **import** keyword to get at those procedures, e.g.,



```
main.py
1 import myhelpers
2
3 # now you can use the procedures
  defined in myhelpers module
4
5 fahr_temp = 72
6 get_celsius = myhelpers.to_fahrenheit
  (fahr_temp)
7
8 # ... continue with program

myhelpers.py
1
2 def to_fahrenheit(celsius):
3     return celsius * 9 / 5 + 32
4
5 def to_celsius(fahr):
6     return (fahr - 32) * 5 / 9
7
8
```

the 'import' keyword allows main.py to use what is in myhelpers.py

Note

1. The file into which you are importing a module ***need not be named*** *main.py*
2. The import example above assumes the files are in the same directory or folder — i.e., alongside each other — like so:

```
some-directory/  
├── main.py  
└── myhelpers.py
```

By Barley-Benincasa Lab & Studios  in cooperation with The Intern

© Copyright 2022-2023.