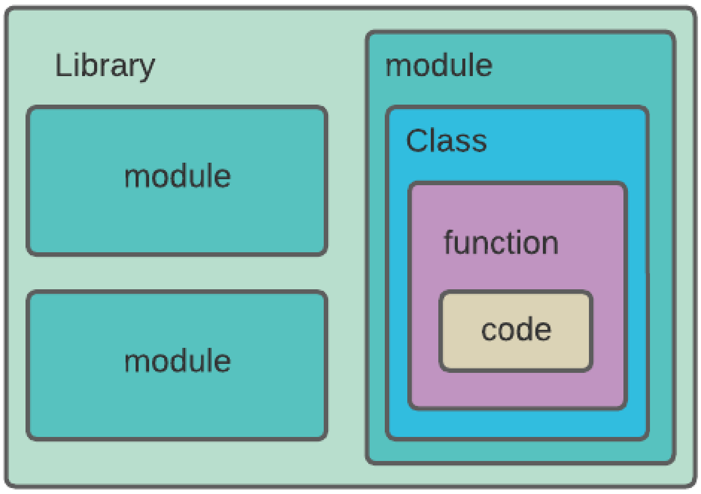
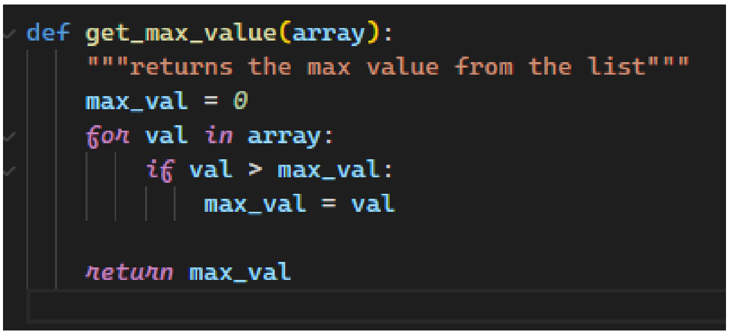
How programmes are organised

Background

To structure your code in a way that makes it readable and easier to scale you need to organise them in libraries, modules and classes. A library is a folder which contains all the folders related to a topic. A module is a folder, which contains all the files related to the same domain. A module is characterised by a **__init__.py** file which should be kept empty and is used to make the python interpreter aware that the folder is considered to be a module. Modules are a collection of classes. Classes are a collection of variables called properties and functions called methods.



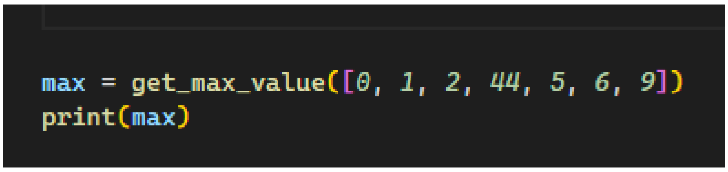
Methods are blocks of code like the following **(a)** which are run when the function is called as shown below **(b)**.



```
def get_max_value(array):
    """returns the max value from the list"""
    max_val = 0
    for val in array:
        if val > max_val:
            max_val = val

    return max_val
```

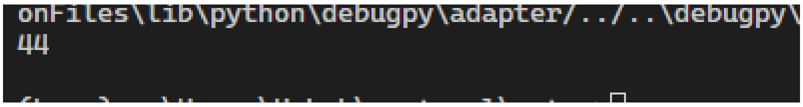
(a)



```
max = get_max_value([0, 1, 2, 44, 5, 6, 9])
print(max)
```

(b)

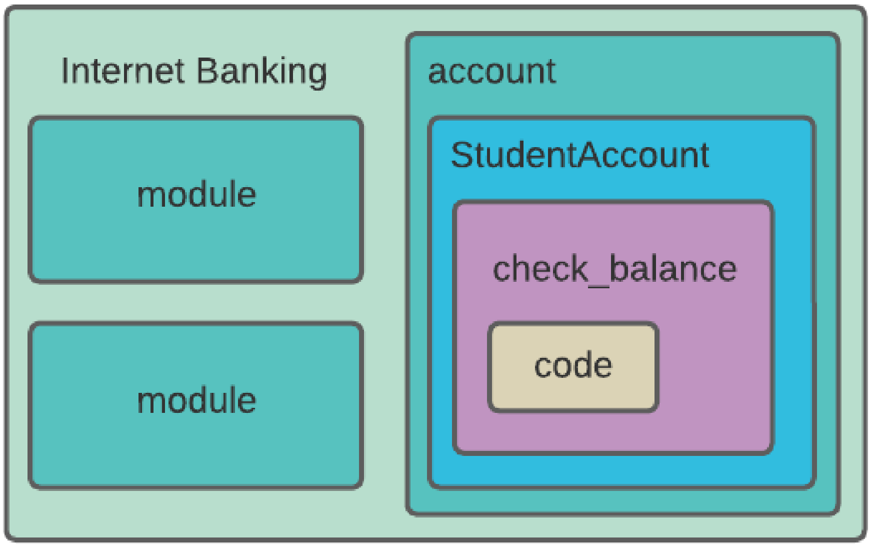
Output



```
onFiles\lib\python\debugpy\adapter/../..\debugpy\
44
```

Example of a library

The following example shows how a library "**Internet banking"** can be used organised. The library will have lots of folders, one of the folders is **account** the account folder has a fie called **student_account.py** where there is a **StudentAccount** class, the **StudentAccount class a check_balance** method which contains a block of code which allows you to check the account balance.



An example of the following library can be shown here kesler20/notes (github.com)