Packages and Modules

This lesson will introduce the concept of packages and modules in python.

WE'LL COVER THE FOLLOWING

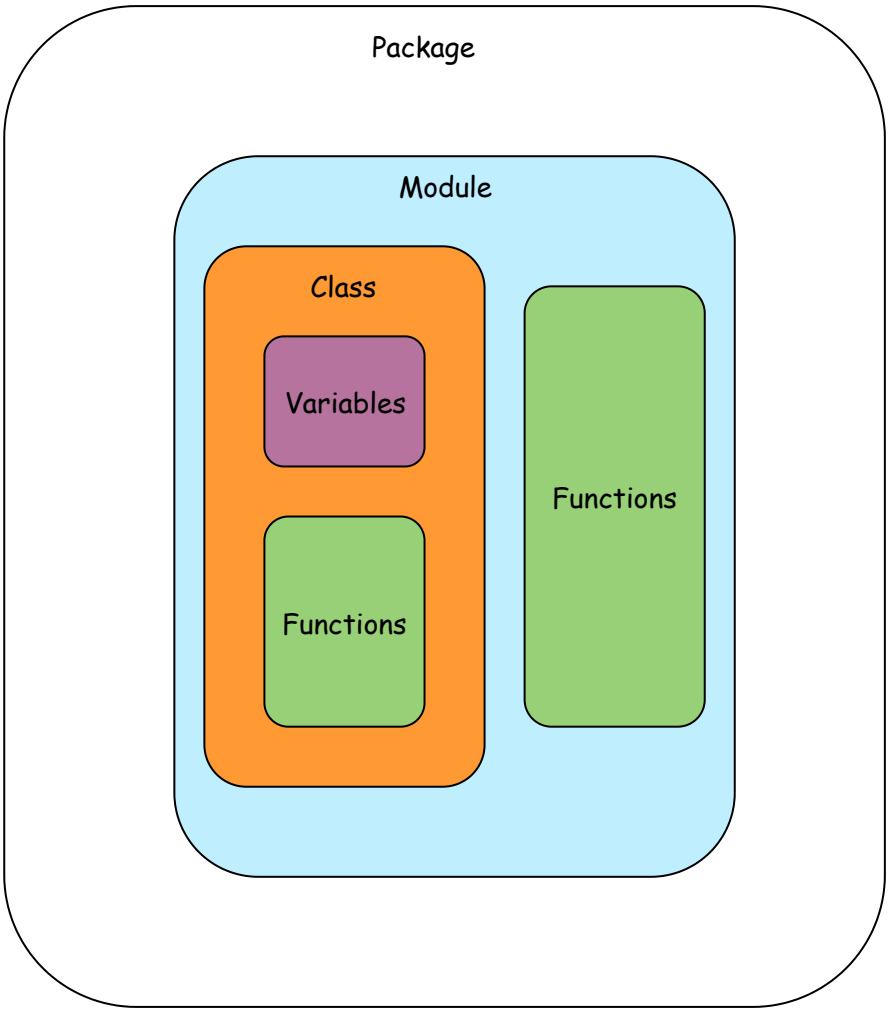
Python is known for its brilliant support for third party resources. The language makers can't add every functionality to the language, therefore, the open-source community has created multiple *packages* that can be used when needed. We do not need to write code for everything. People have already done that for us. We only need to use that code in our program.

Classes and Modules

A **class** in python is a collection of one or more functions and variables. The functions and variables that are defined in a class are called its **members**. Member functions are formally called `methods`.

A **module** in python is a collection of one or more classes and functions that we can reuse in our programs.

A collection of different modules, classes, and functions is known as a **package**. The illustration below describes this hierarchy



Some popular packages in python are *math*, *numpy*, *pandas*, *scipy*, etc. We will look at all of these in this course.

To use an outside function in our program, we have to **import** it first. We can import a complete module, a complete class, or a single function. We do that with the `import` statement. The following are different ways the `import` statement can be used:

```
import modulename
# or
from modulename import classname
# or
from modulename import functionname
```

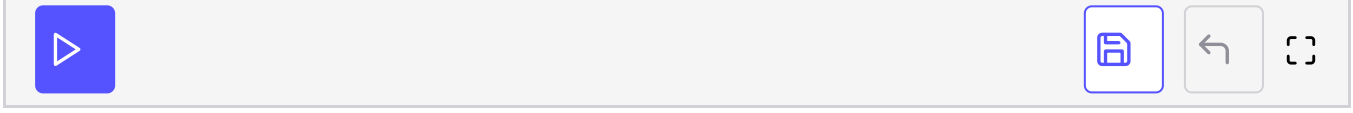
Importing a Module

Let's look at an example with the `datetime` module. This module has functions and classes related to the current date and time, apart from other functionalities.

functionalities.

```
# Import module
import datetime

# Use module
today_date = datetime.date.today()
print(today_date)
```



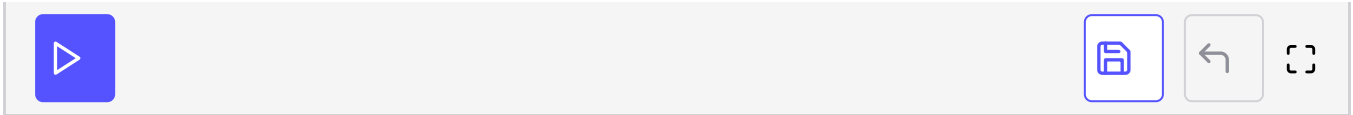
We import the complete module in **line 2**. We use the module in **line 5**. We access the `date` class in the module by writing `.date`. Then we access the `today` function from the class by writing `.today()`. This function does not require any arguments. It returns the current date which we print in **line 6**.

Naming imports

We can name our imports with the `as` keyword. We can then refer to them using this keyword.

```
# Import a module
import datetime as dt

# Use the module
today_date = dt.date.today()
print(today_date)
```

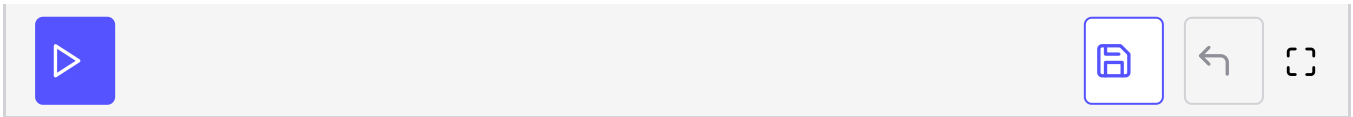


In this instance, we import the `datetime` module as the alias `dt`. Everywhere in the program below, we can use `dt` as we have used in **line 5**.

Importing a Class

```
# Import class from module
from datetime import date as dt

# Use class function
today_date = dt.today()
print(today_date)
```



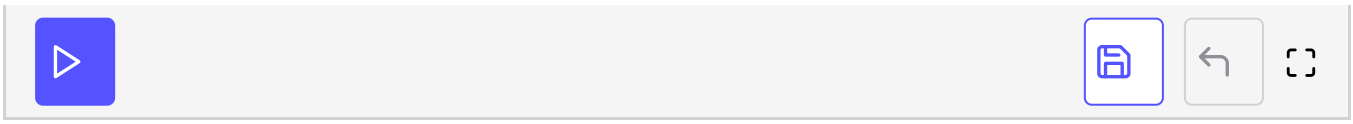
There is a class called `date` available in the package. We import it in the first line as `dt` and use its function `today` in **line 5**.

Import using `*`

We can import everything from a module using the `*` and use it in our code.

```
# Import every thing from module
from datetime import *

# Use class function
today_date = date.today()
print(today_date)
```



We can import everything from a module by writing `*`. This means that we can use the `date` class without prepending the name of the package before it, as we have done in **line 6**.

So, we now have the necessary basic knowledge of python to move forward in this course. In the next lesson, you will be tested on the concepts learned in this chapter.