What is the Python Standard Library?

In this section, we will explore the Python Standard Library.

We'll cover the following

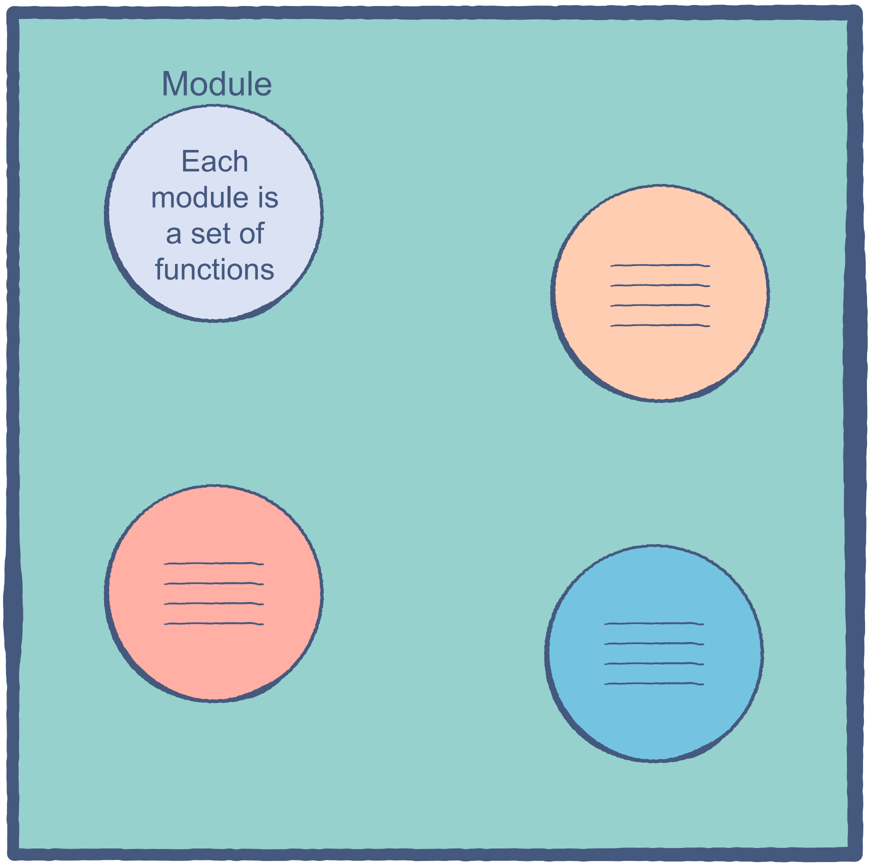
- Definition
- Importing a Module

Definition

> The Python Standard Library, PSL, is a collection of pre-defined **modules**, or sets of **methods** which help us perform certain tasks in Python.

The library contains many useful utilities which spare us a lot of time. There are different sorts of complex mathematical functions, high-level data types, network programming tools, and this is just the tip of the iceberg!

The Python Standard Library - A collection of modules



Generally, a module contains functions related to a particular aspect of programming. This makes things easy because we know which part of our program requires which module.

In Python, we can make our own modules, but that's a story for another time. We'll be focusing on what Python offers us out-of-the-box.

Importing a Module

To use the methods of a module, we must import the module into our code. This can be done using the `import` keyword.

Let's import the `datetime` module which contains several methods related to the current date and time. As always, these methods can be accessed using the `.` operator:

```
# Importing modules at the beginning of the program is a good practice

import datetime

date_today = datetime.date.today()  # Current date
print(date_today)

time_today = datetime.datetime.now()
print(time_today.strftime("%H:%M:%S"))  # Current time
```

In the code above, `datetime.date` and `datetime.datetime` are classes in the `datetime` module. Each class contains its own methods.

If we only want a particular class from a module, we can use the `from` keyword in the following format:

```
from module import class
```

Let's try this out:

```
from datetime import date

# Now we only have the methods of the date class
date_today = date.today()  # Current date
print(date_today)
print(date_today)

# These won't work

# time_today = datetime.datetime.now()
# print (time_today.strftime("%H:%M:%S"))# Current time
```

We can also give our own names to the modules that we import by using the `as` keyword. Let’s rename `datetime` to `dt` and use it in the program:

```
import datetime as dt

date_today = dt.date.today()  # Current date
print(date_today)

time_today = dt.datetime.now()
print(time_today.strftime("%H:%M:%S"))  # Current time
```

We’ve learned how to import and use PSL modules in our code. In the next lesson, we’ll look at some popular modules.