Module 4. Modules, Libraries, Third Party Packages

Modules

Modules provide a way to break down our code into smaller, manageable parts, making it easier to understand, maintain, and collaborate with others.

Modularization = Software code that can quickly become very *complex*, and in this case, it is essential to break it down into smaller pieces that can be *developed*, *tested*, and *maintained separately.*

Modules can be:

- **internal**, when they are developed by you or your co-workers, in the same code base as your application).
- they can come from Python's so-called *Standard Library* (**built-in** modules),
- or can be **external** (*3rd party*), when they come from outside of your domain, and you potentially have no idea how they work internally.

Standard Library

Standard Library = the *basic set of tools* directly coming with Python language.

You've used this before when you used *data types* (string, numeric types, collections, etc.), *built-in functions* (e.g. len, print).

There are a good number of modules that are *included in the Standard Library*.

Third-Party Packages

Third-party packages = these modules are developed and shared by the community, and most of them are open-source, which means that you can *feel free to use them in your projects* without any obligations.

You just need to download such a module, import it into your own code and you're off to the races.

Important to note:

When you run third-party code, that code has the same power over your computer as your own code.

pip: Package Installer for Python

We use **pip**, which stands for **p**ackage **i**nstaller for **P**ython to download packages containing Python modules.

It comes with the official Python installation since version 3.4, so once you installed Python, **pip** will automatically be installed too.

pip collects the packages to install from a package repository called PyPi - The Python Package Index.

The **Replit** sandbox **has pip installed by default**, so you can use it from the shell without any extra configuration.

To install a third-party package, all you need to do is to *run the following command with the name of the selected package:*

```
pip install <package_name>
```

Once the installation is done, read the package's own documentation (which you can usually find on its website) to find out how to use it in code.