Modules and Packages

PIERIAN DATA

Using PyPi
With pip install

PIERIAN DATA

- PyPI is a repository for open-source third-party Python packages.
- It's similar to RubyGems in the Ruby world, PHP's Packagist, CPAN for Perl, and NPM for Node.js.

Complete Python Bootcamp

- So far we’ve really only used libraries that come internally with Python.
- There are many other libraries available that people have open-sourced and shared on PyPi.
- We can use **pip install** at the command line to install these packages.

Complete Python Bootcamp

- By installing Python from python.org or through the Anaconda distribution you also installed **pip**
- **pip** is a simple way to download packages at your command line directly from the PyPi repository

Complete Python Bootcamp

- There are packages already created for almost any use case you can think of!
- A quick google search will usually help you discover a link to the PyPi page for the package, or for the package documentation.

Complete Python Bootcamp

- Let’s quickly show you how to download and install external packages.
    - Windows Users: Command Prompt
    - MacOS/Linux Users: Terminal

Writing Your Own Modules and Packages

PIERIAN DATA

Complete Python Bootcamp

- Now that we understand how to install external packages, let's explore how to create our own modules and packages.
- Modules are just .py scripts that you call in another .py script.
- Packages are a collection of modules.
- Let's create some examples!

__name__
and
"__main__"

PIERIAN DATA

Complete Python Bootcamp

- An often confusing part of Python is a mysterious line of code:
    - if __name__ == "__main__":

Complete Python Bootcamp

- Sometimes when you are importing from a module, you would like to know whether a modules function is being used as an import, or if you are using the original .py file of that module.

Complete Python Bootcamp

- Let’s explore this some more, but make sure to check out the full explanatory text file that is in this part’s folder!