

## Day 03

I'm assuming that you are here after watching the Day 03 video, now that you know about pip, modules and its types. Great!

Now let me confuse you a little bit. Because that's the point where I got confused while I was on day 03.

When you search "pandas" on the internet it says "**pandas – Python Data Analysis Library**" and when I don't install pandas on my local computer and try to import it inside my python code, it throws an error like this:

```
>>> import pandas
Traceback (most recent call last):
  File "<python-input-0>", line 1, in <module>
    import pandas
ModuleNotFoundError: No module named 'pandas'
```

Well, why it says **module** not found, why it is not saying **Library** not found error? because we have just seen that internet says that pandas is a library which is used in data analysis. Then why I'm seeing an error of **ModuleNotFoundError**, why not library? To understand this concept, it's important to first know the key difference between packages and modules in Python. A module is a single Python file that contains code and statements, similar to a Python script. On the other hand, a package is a folder that contains multiple modules. This folder can also have subfolders, which in turn can contain more modules.

Now coming back to our confused problem, why it is not showing **libraryNotFoundError**? Well, Python's import system works with modules and packages, not libraries. The term "library" is just a human-friendly way to describe a collection of code, but Python itself doesn't officially recognize it. Instead, a module is a single Python file, while a package is a folder containing multiple modules. When you try to import something like pandas, Python searches for a module or package with that name. If it doesn't find it, it raises a ModuleNotFoundError, not a "LibraryNotFoundError," because Python does not recognize "library" as a technical term. In Python's internal system, everything being imported is treated as a module, even if it is actually a package. This is why Python does not have a separate error like "PackageNotFoundError"—it considers both modules and packages as part of the module system and raises the same ModuleNotFoundError when it cannot find them.

Phew! That was pretty confusing. You can tweet about it and share the knowledge with others.