Built-In Libraries vs. External Libraries

As we covered in an earlier course, the Python Standard Library comes as part of the Python insta for the most common tasks you can do with Python. But there's tons of other things you might w not all of them are in the standard library. This is where external modules come into play. When c module that they think others might find useful, they publish it in ***PyPI*** -- also known as the ***Pytho*** (https://pypi.org). We can browse this repository of Python modules to find the module we need. projects, which are classified by different categories, like topic, development status, and intended

In this module, we're going to be ***transforming*** and ***converting*** images. To do that, we'll be using manipulation: the ***Python Imaging Library (PIL)***. The original PIL library hasn't been updated sin Python 3. Fortunately, there's a current *fork* of PIL called ***Pillow***, that properly supports Python 3 Pillow library is packaged with the name **pillow**, but the module name in Python is still **PIL**.

If you try to import the PIL module on a computer that doesn't have pillow (or PIL) installed, you n

```
>>> import PIL
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'PIL'

```

Okay, looks like I don't have that module yet! As we covered in an earlier course, there are severa to your Python environment. PIL is a pretty common library, and on Linux it's usually available as

```
user@ubuntu:~$ sudo apt install python3-pil
Reading package lists... Done
Building dependency tree
(...)
Unpacking python3-pil:amd64 (4.3.0-2) ...
Setting up python3-pil:amd64 (4.3.0-2) ...

```

For other environments, you should use Python's package installer, ***pip3***. Like this:

```
$ pip3 install pillow
Collecting pillow
  Downloading https://files.pythonhosted.org/packages/85/28/2c72ba965b52884a0bc
     |████████████████████████████████| 3.9MB 1.7MB/s
Installing collected packages: pillow
Successfully installed pillow-6.2.1

```