In Preparation for Monday's Class : Setting up Python

In class on Monday, we are going to spend some time talking about how to set up and configure Python for this course.

If you want to try to do some of the steps in class on Monday, following along with me, I have included some instructions below that you will want to do before class.

In addition to stepping through the process, I'll try to give a big-picture overview of the relationships between program languages, language distributions (like Anaconda), toolboxes, repositories for toolboxes, virtual environments, managing virtual environments, environment tools (like Anaconda Navigator and conda), IDEs vs. Notebooks, and configuring those to use the right virtual environment. These are all things you need to have some basic knowledge of to do real work in scientific computing and data science.

In an intro programming course using Python, you can probably use any relatively recent 3.x.x version of Python and be fine, maybe doing some basic configuration to make sure you have the right packages in the right places.

In other contexts, where you are using discipline-specific specialized packages, in our case psychopy, in the case of computational neuroscience tensorflow and keras, you often need to configure your Python setup in particular ways. For example, psychopy will only work (appropriately) with Python 3.8.x. You might have an older version or a newer version installed and working on your computer, and while whatever version you downloaded was fine for an intro course, you need to configure things for this course.

We will be using the Anaconda distribution of Python. If you want to use a different distribution (say from python.org, and install packages using pip) or you need to have a small installation (using something like miniconda), and you know what you are doing, you will need to configure Python on your own (using the version of Python and the set of packages we will go over in class); Jason can help out when you cannot use Anaconda or you cannot get things working.

If you have never installed Anaconda before, install it. Go to https://www.anaconda.com and it should auto-detect the version for your operating system and let you download and install.

If you have Anaconda from before, and have used Python before, check to see what version of Python you have. Either in a Jupyter Notebook or a Python IDE, type in and run the following code snippet:

```
import sys
print(sys.version_info)
```

It should print out something like this:

```
sys.version_info(major=3, minor=8, micro=13, releaselevel='final', serial=0)
```

That means that I have version 3.8.13 installed (in my case, this is the version in my current environment, I actually have 3.9.x installed too). If you have a later version (3.9.x or 3.10.x) you almost certainly have older versions too (the Anaconda distribution should include 3.7.x through 3.10.x). If you have 3.7.x or older, your best bet might be to just download Anaconda again (it should keep all old versions and all old environments intact, just add newer versions).

If you have Anaconda installed, you should be able to open an app called Anaconda-Navigator. We'll be using that to configure environments on Monday.

In class, we will use Jupyter Notebooks and PyCharm (an IDE). You are free to use whatever IDE you want, but you will need to be able to configure it to use the right environment. Even if you want to use another IDE, you can also download and set up PyCharm to follow along when we use it in class.

PyCharm is free for students, if you go to https://www.jetbrains.com/shop/eform/students

Do not download PyCharm from Anaconda-Navigator or from elsewhere on the jetbrains web site. Those may let you use PyCharm for free, but it will be a scaled down version, or it will be a trial version before being asked to pay. As a student, you can download and use the Pro version for free. Make sure you use your vanderbilt.edu email address.