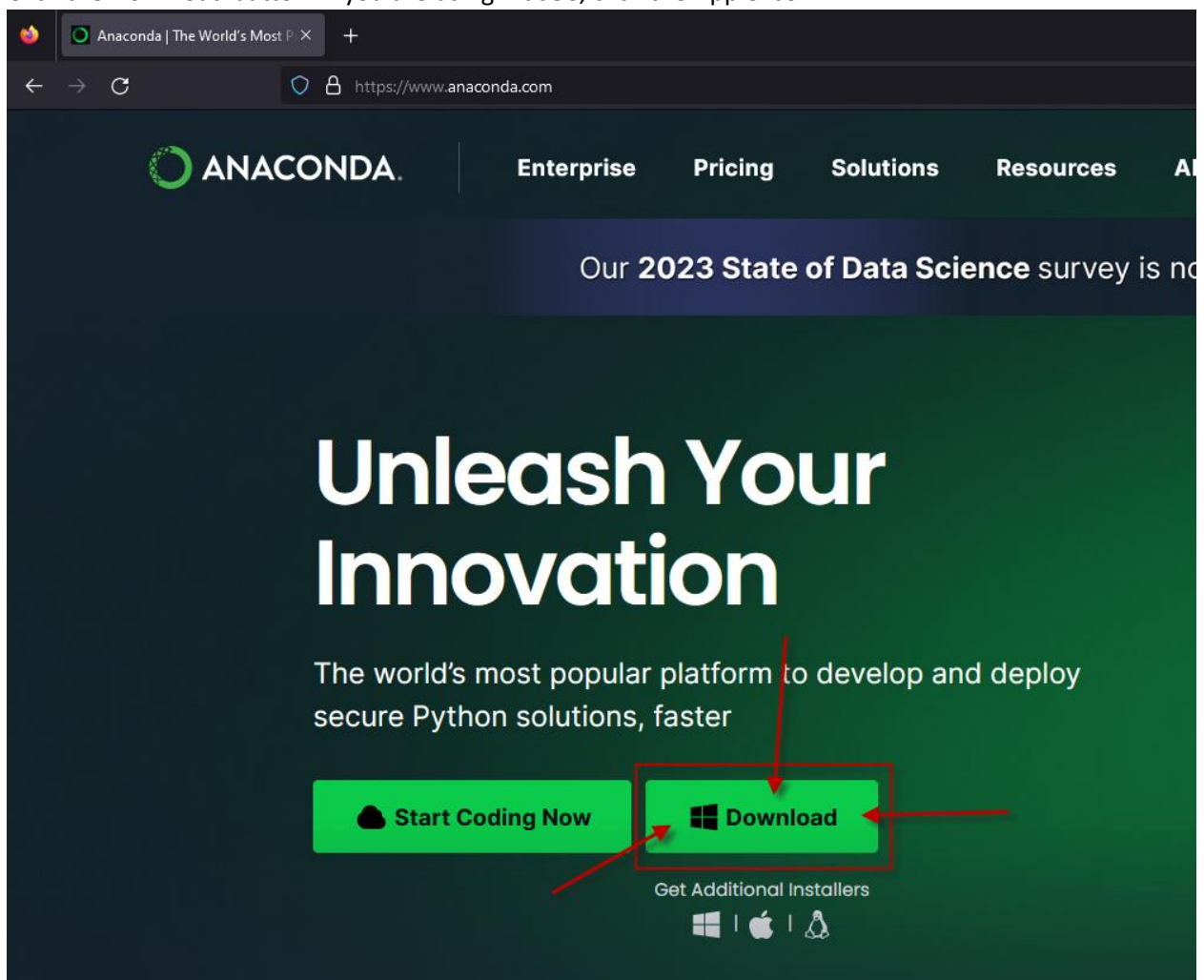
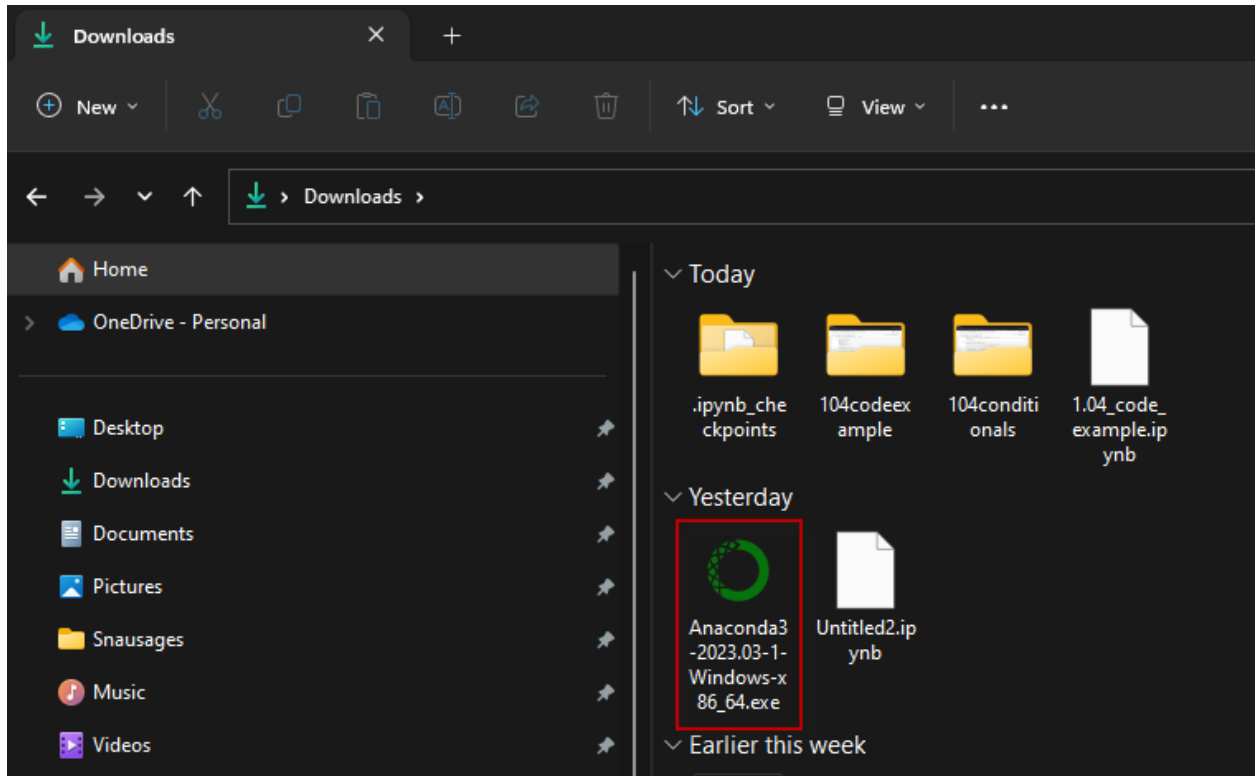


# How to Install Python and Create a Virtual Environment

1. Open your web browser and navigate to [www.anaconda.com](https://www.anaconda.com).
2. Click the Download button. If you are using macOS, click the Apple icon.

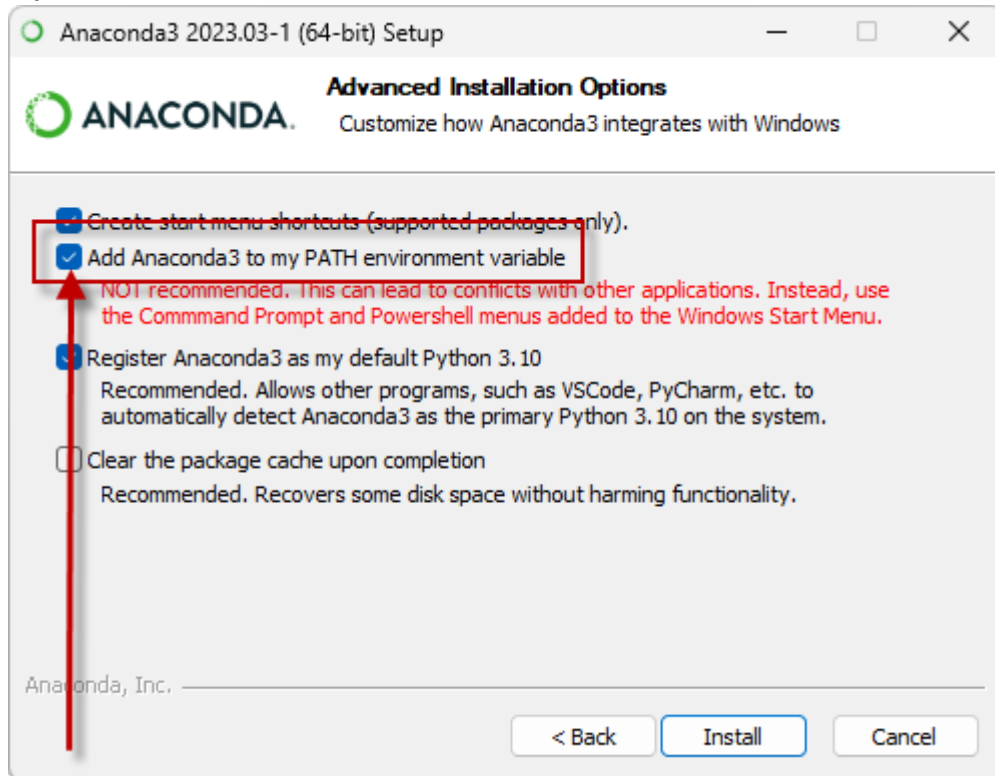


3. Once the file is finished downloading, navigate to your Downloads folder and double-click the installer:



4. Click **Next** to continue.
5. Click **I Agree** for the license agreement.
6. Leave the **Just Me** radio button clicked, and then click **Next**.
7. Choose where you want to install Anaconda, and then click **Next**.

8. **Important note for PC users:** On the next screen, make sure the checkbox for **Add Anaconda3** to my PATH environment variable is checked ON:



9. Click **Install**.
10. Once Anaconda is finished installing, click **Next**.
11. Click **Next** again, and if you want to read some documentation about Anaconda, leave the boxes checked. Otherwise, uncheck them, and click **Finish**. For macOS users, skip the option to install DataSpell.
12. Navigate in your Apps menu to your Anaconda folder, and in the folder, click **Anaconda Navigator**. **Tip:** It is useful to right-click Anaconda Navigator and click **Pin to Start** to add it to your list of quickly-accessible apps.
13. When Anaconda starts, it will check for updates. You can install the updates or sometimes not install them. We recommend you install them to keep the program up to date.
14. To launch Jupyter Notebook, simply click the **Launch** button under the Jupyter Notebook logo!
15. You can now code like a pro in Jupyter Notebook! If you want to open a Python file, just navigate on the **Files** tab to the file you wish to open, and then double-click the file.

## How to Create a Virtual Environment

As Dartanian described in Episode 1.02, the benefits of creating a virtual environment are many. Depending on how far you go in your coding journey, you'll find virtual environments critical to your success. With virtual environments in Python, you can:

1. Install different versions of Python and other packages to see how your code runs in those versions
2. Make mistakes without impacting your primary Python code
3. Troubleshoot problems with code in a safe environment

**Knowing how to create a virtual environment is not a requirement of this course, it's just a very powerful concept and tool that all professionals use.**

The most popular and common method to create a virtual environment within Python is through the Anaconda Python distribution.

To install a virtual environment on your computer using Anaconda:

1. From your Start menu, under Apps, launch Anaconda Powershell Prompt. (On macOS, the program is reachable through your Apps folder.)
2. To verify conda is installed in your path, type **conda -V** and hit Enter:



```
Anaconda Powershell Prompt
(base) PS C:\Users\h > conda -V
conda 23.3.1
(base) PS C:\Users\h >
```

3. The version of conda will display if it's installed in your path.
4. If you need to update the conda environment, enter **conda update conda** and press Enter.
5. To see the available versions of Python available for you to use in your virtual environment, type **conda search ^python\$** and press Enter.
6. To create a virtual environment, type **conda create -n envname python=x.x anaconda** and press Enter.
  - a. Let's say you want to create a virtual environment named Mars that uses Python version 3.10.8. You would type **conda create -n Mars python=3.10.8 anaconda** and press Enter.
7. This part can take a long time, because your computer will automatically download all the different files it needs to emulate that version of Python.
8. Once you're finished downloading and installing your virtual environment, you can type **conda activate Mars** and press Enter (in macOS or Linux, the command is **source activate** or **source deactivate**). Once you travel to Mars, you'll notice the prompt says (Mars) before the rest of the directory listing. This is how you know you're in your virtual environment!
9. To leave Mars and go back to your regular instance of Python, simply type **conda deactivate** and press Enter. (Mars) will change to (base) in your directory listing.



10. If you want to delete Mars completely, you can type **conda remove -n Mars -all** and press Enter.

Great job! There are a few other commands you might find useful as you advance your knowledge of coding. They're listed in the Glossary that's included with your class materials. Happy coding everybody!