

Data Science Installation Guide

Programs We'll Be Installing

In order to be an effective Data Scientist, you'll need to be familiar with a number of programming tools. This is known as your *development environment*.

For this course, we'll be using **Python** as our programming language, performing version control with **Git**, and backing up our projects on **GitHub Enterprise**.

Where You'll Use These Programs

You'll use these programs both before, during, and after our course.

For instance, as part of our course pre-work, we'll ask you to complete some Python code challenges. In order to do that, you'll need to start practicing in an environment similar to a professional setting.

This is where our first program comes in: Anaconda

What is Anaconda?

Anaconda is a python software bundle, owned by a company in Austin called Continuum Analytics. They specialize in enterprise python deployments. Anaconda comes with helpful tools, such as jupyter notebook (interactive editing of python in a web browser), ipython (interactive python within a shell), spyder (an integrated development environment for python), and conda (a package manager - the thing that orchestrates library dependencies for python and allows you to create things called virtual environments).

To keep things simple, we'll walk you through the graphical interface installation for the Anaconda Navigator.

Before We Begin...

Before we download Anaconda, you'll need to confirm two things:

1. Which type of processor your computer is using, and
2. Which version of Python we want.

Anaconda will ask whether you are using a 32 or 64 bit processor, so it's a good idea to verify this first. If you don't know - never fear, we'll help you figure that out.

In this class, we'll be using Python 3. Note that when we refer to python 3, we are talking about the major version of the package. Modern software uses a thing called semver, or [semantic versioning](#). Python 3 is actually python 3.x.y, where x and y represent the minor and patch versions, respectively, of the software. Any version of python where 3 is the major version will be compatible with this course, for example, python 3.7.0 is the stable release at the time of writing.

How to Identify Your Processor Type

Mac:

To tell if you are running the 32-bit or 64-bit kernel (which matters for some device drivers), launch System Profiler and click on the Software heading in the Contents section.

The line 64-bit Kernel and Extensions will say Yes if you are running the 64-bit kernel and No if you are running the 32-bit kernel.

Note: Mac OS X 10.6 Snow Leopard is the last release that a 32-bit Intel Mac can run, Mac OS X 10.7 Lion and up are, by default, 64-bit. Additionally, System Profiler was renamed to System Information in OS X 10.7.

Windows:

1. Press the window key + R. In the run dialog, type in `msinfo32.exe` and hit enter.
2. Look at the **System Type** field under **System Summary**. If the field is **x64-based PC**, you have a 64 bit machine. Else you have a 32 bit machine.

Download Anaconda

Ok! So now that you know:

1. Your processor type: **32 or 64 bit**
2. Our required Python version: **Python 3**

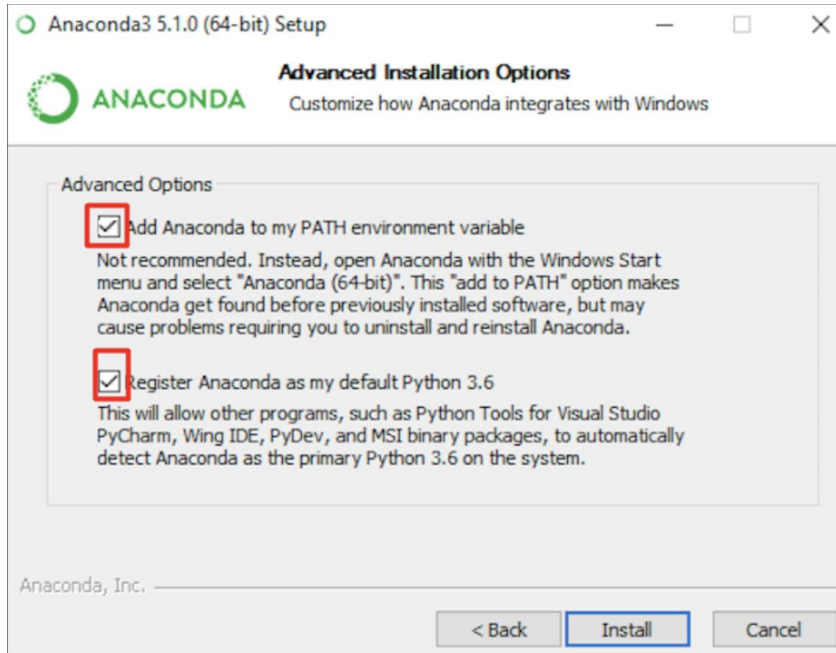
Go ahead and download the appropriate version of Anaconda for your machine. When prompted, select the Graphical Installer for convenience. We'll guide you through the prompts as well.

[Click here to download Anaconda.](#)

Note: You do not need to provide your email address when prompted.

Installation Walkthrough

- Open and run the Anaconda installer
- Read the licensing terms and press “I Agree”
- Select to install for “Just Me”
- Use the default Installation Location
- For Advanced Installation Options, select both



- Press the Install button (this could take a few minutes)
- Press skip and do not install VSCode (we won't be using it)
- Then continue and finish the installation

Great Work! What's Next?

Great work! We'll use Anaconda to open Jupyter notebooks and run python code. But before we get started, we'll also need to download a **version control platform** (git) to save our code.

What is Git?

Git is a *version control system* that tracks changes made to project files on your computer or online.

Git creates backups of your work and lets you rewind to previous copies if you run into issues. This makes it easy to coordinate versions and work across multiple machines and people.

Git is run as a command line program. User-friendly platforms, like Github, are built on top of Git, which means they require us to install Git before we can use them.

We'll install Git now so we can use Github later.

Git: Installation & Configuration

PC Users:

1. First, remember how you identified your computer processor (32 vs 64 bit) earlier? Keep that in mind.
2. Go to [this link and click download for your operating system.](#)
3. Note that if you select "Windows" it will automatically download the 32-bit version. If you want the 64-bit version, click through and select that one (and feel free to delete the other one). Note that 32-bit versions are backward compatible and will run on 64-bit machines.
4. Once download is complete, click to run the installer.
5. Please select the following options:
 - Use Git from the Windows Command Prompt
 - Use the native Windows Secure Channel Library
 - Select VIM as Git's default text editor
 - Use Git from the Windows command prompt
 - Use the native Windows Secure Channel Library
 - Checkout Windows-style, commit Unix-style line endings
 - Use MinTTY

Mac Users:

1. Go to [this link and click download for your operating system.](#)
2. Run the installer.
3. Select the default settings and continue until finished.

Now that we've installed git, let's go ahead and quickly set up Github.

Create A Github Account

Github is an online platform for hosting projects and sharing code. Github is built on top of Git and features an online interface to access your code. In this class, we'll use Github to access course materials and submit our work.

However, Github.com is a public community site. Instead, we'll be using a private version called Github Enterprise, hosted on General Assembly's servers.

In order to use Github with your class, you will first need to create an account.

1. Click on [this link.](#)

2. Press the "Sign Up" link in the upper-right hand corner.
3. Create a new account using your work email. Make sure to choose a username similar to your real name!
4. Post your Github username in the #classroom Slack channel, so we can invite you to the course repo.

Great work so far! By now you've:

- Installed Anaconda
- Installed Git
- Created an online account for GA's version of Github.

Well done! These tools will make up your development environment in and outside of the classroom.

For the last portion of our lesson, we'll provide a quick tutorial on how to open a data science notebook using Anaconda.

Let's Launch Anaconda!

Go ahead and launch Anaconda Navigator.

Windows:

- Start → Anaconda Navigator

Mac:

- Use Spotlight Search (⌘ + spacebar, then search Anaconda) to launch.

Note: Once you select it, get up, stretch, and give yourself a high five. It may take a few minutes to fully launch.

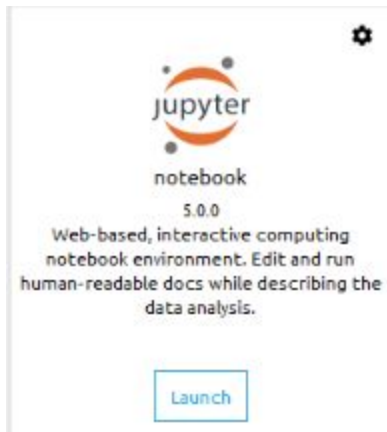
What Are Jupyter Notebooks?

Next, we'll use Anaconda to open a **Jupyter Notebook**.

Jupyter notebooks are an open-source framework used to write, run, and share data analysis. The notebook is made up of frames that can store code or commentary. Frames can be run to test or visualize their outputs, and can even build on top of each other to perform specific functions.

How To Get Started

Once you've opened Anaconda Navigator, try launching a new Jupyter Notebook in your browser by clicking the **Launch** button on the Jupyter Notebook tile.



The Jupyter Menu Interface

Opening up Jupyter Notebook should create a new tab in your browser. Here, you'll see the Jupyter logo and a list of your computer's folders. This menu interface works just like Windows Explorer (PC) or Finder (Mac), so feel free to browse around!

Navigate to a folder where you'd like to store materials for our class. This will be the destination folder for our new notebook.



Launching a New Notebook

To create and launch a new Jupyter Notebook, click on the "New" button in the top right corner and select the environment you want. Remember, we will be using "Python 3" in this class.

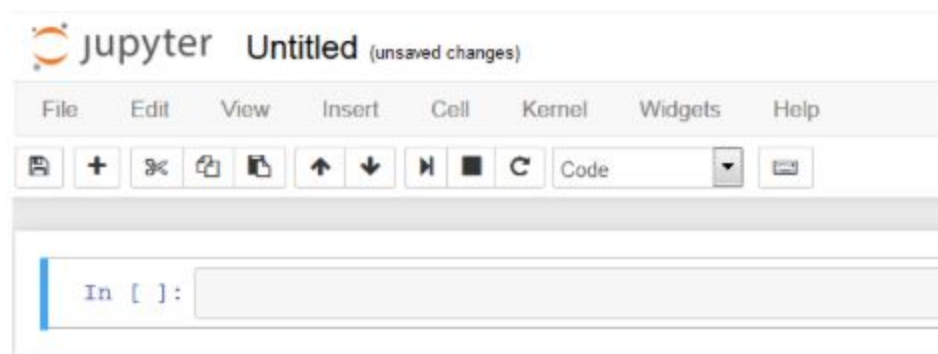


The Notebook Interface

Clicking on the "New" button should open a blank notebook in a new browser tab. Doesn't look like much, does it?

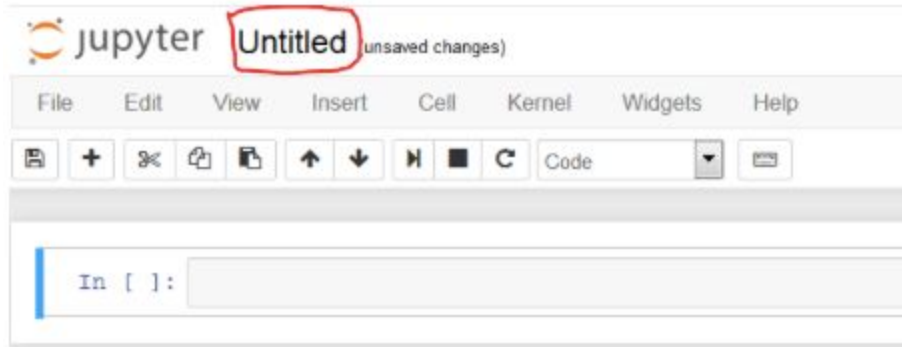
Notebooks are a surprisingly powerful tool. Each *dataframe* can store and run code, and build on the code output by previous frames. You can include text to explain and walk through your analysis, and output visuals to help indicate what's going on.

The menu functions let you save, copy, reorganize, or run any code you enter into your frames. We'll spend more time going over these functions in class.



Naming Your New Notebook

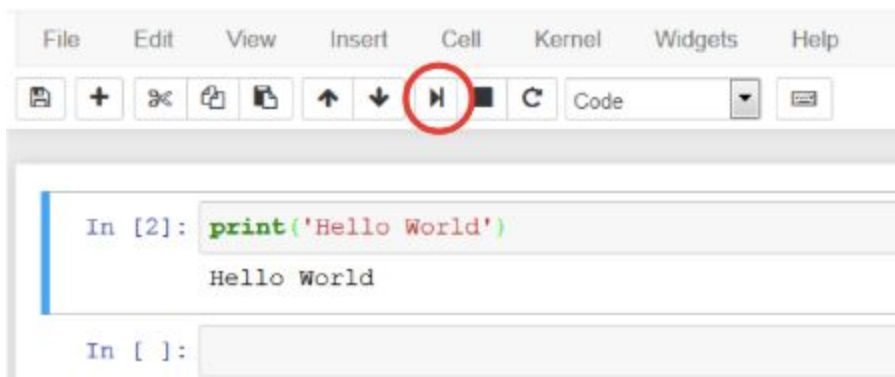
To change the filename of your notebook, click where it says "Untitled" and enter a new name in the pop up screen. We'll call this one "GA Demo."



Running Code In Your Notebook

Try entering `print('Hello World')` into the first cell of your new notebook. Then click on the **Run** button.

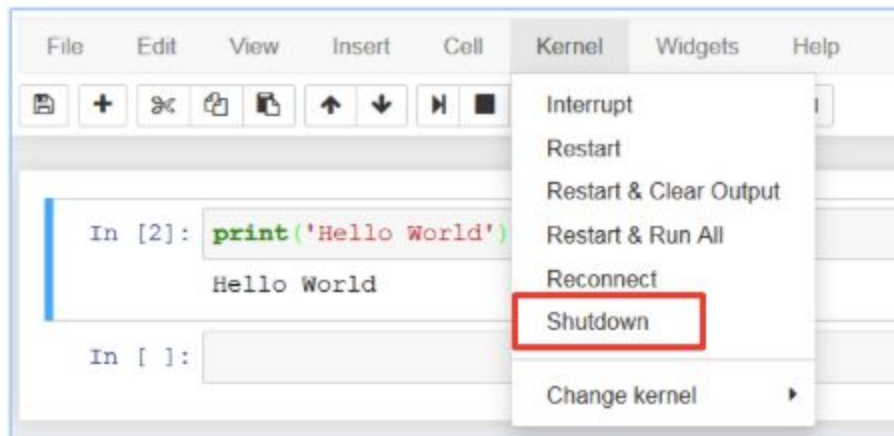
Pro Tip: On Windows, the shortcut key for "run" is **control + return**.



Shutting Down Your Notebook

Ok, great work. Now let's walk through how to save and find your work later.

- Click on the **Save** button on the far-left side of the interface.
- Then navigate into the "Kernel" menu and click "Shutdown".



View Your New Notebook

- Next, go ahead and close your new notebook's tab in your browser.
- Then go back to our original Jupyter tab, where you can see your folder menu.
- You should now see a new file - your new notebook!

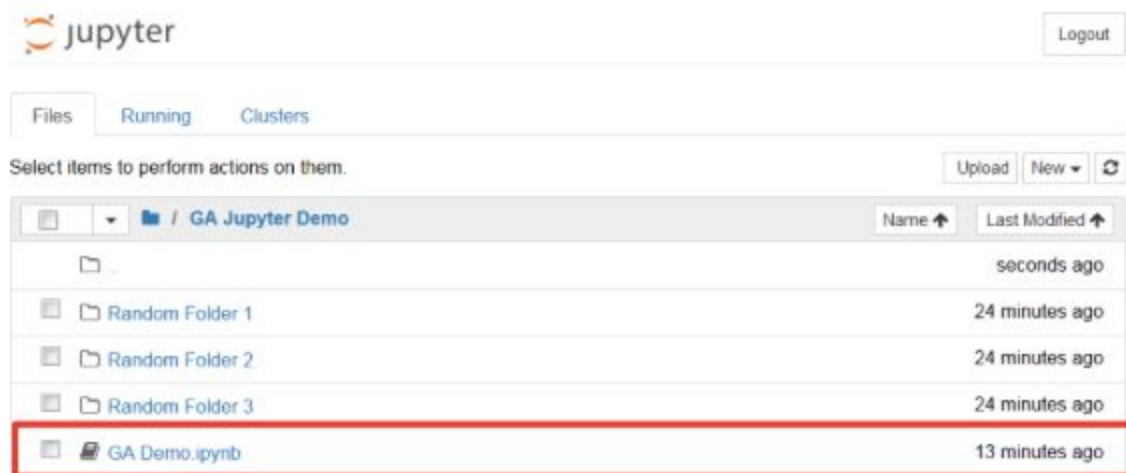


Restart or Launch Any Notebook

You can restart your new notebook at any time just by clicking on it.

Remember, the Jupyter notebook menu shows all the files on your computer, just like Explorer or Finder. We'll be sharing many notebooks with you in this course - you can save them to your computer and open them here whenever you want!

Congratulations - that's it! You're all set. So are you ready to do some data science?



Let's Review

Anaconda

Anaconda is an open-source data science package that pulls together most of the commonly used tools and libraries we'll be using to analyze data in this course, including **Python** and **Jupyter Notebooks**.

Git

Git is a version control system that logs your code and can be used to backup projects. It's useful for collaboration and can be used to manage and merge different versions of your work.

Github

Github is an online platform that attempts to simplify and build on the version control functionality provided by **Git**.

Github allows you to store your code in an easily accessible online interface and provides a visual interface to manage your versions or collaborate with others. Github is also a social community.

In this class, we'll use a private version called **Github Enterprise** that is run on GA's own servers.

Jupyter Notebooks

Jupyter Notebooks are a tool used to write, run, test, visualize, and share data analysis and data science work.

Notebooks are an open-source tool, built and maintained by a community of passionate programmers who want to make it easier for everyone to work with data.

Lastly...

We'll be using a few additional tools in the classroom, **Slack**, **Chrome** and a text editor. Please download and install them using the links below.

- [Slack for Mac](#) or [Windows](#) is a messaging platform that we'll use to chat in class. You will receive an email invite to join the Slack team.
- [Chrome](#) is the web browser we'll use to take advantage of its built-in developer tools.
- Install the following text editor:
 - [Atom](#)

And That's It!

Congratulations on making it this far. You're now ready to start learning data science.