**Using Python:** How to install Python on your personal computer and getting started



*Written by Kyle Monahan on March 20th, 2019*

*Last updated August 27th, 2020*

# **Introduction**

[Python](https://www.python.org/) is an interpreted, high-level, and general-purpose programming language. It was created by Guido van Rossum and first released in 1991 and focuses on creating highly “human readable” code with significant whitespace. If you are looking to write a script (a set of instructions for the computer) to automate repetitive tasks, Python can help.

In this tutorial, we will provide instructions for installing Python and getting started with the programming language. At Tufts, we have deployed Anaconda Python, which is an open-source version (also called a *distribution*) of Python that contains additional functionality through included bundles of software called packages.

Anaconda allows you to:

* Download 1,500 data science packages in Python
* Use Jupyter Notebooks to create collections of Python code and results, called notebooks
* Manage libraries, environments, and dependencies
* Use deep learning libraries such as *scikitlearn* and *TensorFlow*
* Work on a Mac, Windows, or Linux PC
* Use virtual environments to create a reproducible workflow

Python is available in computer labs such as the [Data Lab](http://datalab.tufts.edu/) for use by the Tufts community. If you would like it on your personal or Tufts-owned computer, you will need to download and install the software.

Table of Contents

[**Introduction** 1](#_Toc25701082)

[**Installing Python on your personal computer** 1](#_Toc25701083)

[**Learning Python in the Data Lab** 2](#_Toc25701084)

[**Learning Python on your own** 2](#_Toc25701085)

[**Use Python on the research cluster** 3](#_Toc25701086)

[Learning Geospatial Python 4](#_Toc25701087)

[Additional program documentation 4](#_Toc25701088)

# **Installing Python on your personal computer**

To install Python, you can follow the steps below:

1. Navigate to the link to the most recent version of Anaconda Python:
	1. Link: <https://docs.anaconda.com/anaconda/install/>
2. Click on your operating system:
	1. [Installing on Windows](https://docs.anaconda.com/anaconda/install/windows/)
	2. [Installing on macOS](https://docs.anaconda.com/anaconda/install/mac-os/)
	3. [Installing on Linux](https://docs.anaconda.com/anaconda/install/linux/)
3. A new page will appear. Click on “Download the Anaconda installer.”
4. Follow the instructions on that page. To run the installer:
	1. For Mac:
		1. To start, **click** on the downloaded \*.dmg file.
		2. A graphical window will appear, click “**Continue**” to start the installation process.
		3. Review the Readme and Click “**Continue**” if you accept the Terms and Conditions of the Read Me.
		4. Review the License and Click “Continue” if you accept the Terms and Conditions of the License.
		5. Note the installation size (about 2.26 GB). Click **Install** to install the programs, packages, and other materials.
		6. You also have the option to install Visual Studio Code, another IDE. If you’d like to, you can click **Continue** to install that.
	2. For PC:
		1. To start, click on the downloaded \*.exe file.
		2. A graphical window will appear, click “**Continue**” to start the installation process.
		3. Review the Readme and Click “**Continue**” if you accept the Terms and Conditions of the Read Me.
		4. Review the License and Click “Continue” if you accept the Terms and Conditions of the License.
		5. Note the installation size (about 2.26 GB). Click **Install** to install the programs, packages, and other materials.
		6. You also have the option to install Visual Studio Code, another IDE. If you’d like to, you can click **Continue** to install that.

# **Learning Python in the Data Lab**

In addition, we have [PyCharm Educational](https://www.jetbrains.com/pycharm-edu/) installed on computers in the [Data Lab](http://datalab.tufts.edu/). PyCharm is an integrated development environment (also called an *IDE*) for Python and provides additional assistance in writing Python code. By default, PyCharm uses Anaconda Python in the Data Lab.

If you select “Learners” after starting PyCharm, and then go to Browse Courses > Introduction to Python, you can go through an initial training in Python right in the Data Lab! Details here: <https://www.jetbrains.com/pycharm-edu/learners/>

All students with a valid \*.edu email can use PyCharm for free on their own laptop as well! Sign up at the website link above to start learning.

# **Learning Python on your own**

To get started with Python, take a look at the following links:

* **Getting Started**: For getting started check out the Python.org beginners guide, available in English, Spanish, and Chinese:
	+ Beginners Guide: <https://wiki.python.org/moin/BeginnersGuide>
	+ Other materials:
		- Automate the Boring Stuff with Python: <https://automatetheboringstuff.com/>
		- Introduction Tutorial for Python 3: <https://docs.python.org/3/tutorial/index.html>
	+ These materials are good to get an initial overview of Python, and get used to specific terms (number, strings, lists, control flow, data structures, tuples, etc.)
	+ **Software Carpentry** (along with Data Carpentry) are organizations dedicated to a kind and thoughtful introduction to working with digital tools. Their Python tutorials assume little background in programming and are useful for getting started:
		- <https://swcarpentry.github.io/python-novice-inflammation-2.7/>
		- <https://swcarpentry.github.io/python-novice-gapminder/>

* **Online Training**: In addition, we have access to tutorials from [Tufts Lynda campus,](https://access.tufts.edu/software/lynda-campus) now called LinkedIn Learning:
	+ *Python Statistics Essential Training*: Focusing on how to do statistics on Python <https://www.lynda.com/Python-tutorials/Python-Statistics-Essential-Training/711826-2.html?org=tufts.edu>
	+ *Learning Python:* This is more of an introductory course, best to do after reviewing the getting started guide listed above.<https://www.lynda.com/Python-tutorials/Learning-Python/661773-2.html?org=tufts.edu>
	+ *Advanced Python:* This is a great course. <https://www.lynda.com/Python-tutorials/Advanced-Python/699337-2.html?org=tufts.edu>
	+ *Python Graphical User Interface (GUI) Development*: <https://www.lynda.com/Python-tutorials/Python-GUI-Development-Tkinter/802858-2.html?org=tufts.edu>
* **Online platforms:** Another option is to use an online platform for learning Python, which will step you through common workflows in a more project-focused approach. One of the most popular are **DataCamp** and **CodeAcademy**. Other examples are provided below, with varying costs and depth.
	+ - <https://www.datacamp.com/courses>
		- <https://www.codecademy.com/>
		- <https://www.codeschool.com/>
		- <https://www.codementor.io/learn-python-online>
		- <http://www.learnpython.org/>
		- <http://www.pythonchallenge.com/>
		- <http://codingbat.com/python>
		- <https://py.checkio.org/>
* Other online guides:
	+ - <http://www.tutorialspoint.com/python/>
		- <http://docs.python-guide.org/en/latest/intro/learning/>
		- <https://developers.google.com/edu/python/>
* **Tufts Data Lab**: Recordings and instructional materials on similar statistical software are available on the Tufts Data Lab website: <https://sites.tufts.edu/datalab/learning-statistics/stats-online-tutorials/>
	+ We have recordings of the Python workshops available online. We post these with the material used during the workshops (code, learning materials and Powerpoint files where applicable). You can access these at the link here: <http://go.tufts.edu/StatsWorkshopRecordings>
* **Further Questions**: Reach out to Kyle Monahan, Senior Data Science Specialist anytime with questions on how to access and use Python for your research project: DataLab-Support@elist.tufts.edu

# **Use Python on the research cluster**

Python Version 2.7 and Python Version 3.6 are available on the research cluster. You can request a cluster account by going to research.uit.tufts.edu and submitting the application form.

To use Python on the research cluster, you have to take the following steps after requesting an account:

1. Login to the OnDemand portal at <https://ondemand.cluster.tufts.edu/> using your Tufts UTLN (username) and password
2. Open a connection to the cluster by going to Clusters > HPC Shell Access at the top menu.
3. You will see a black screen with your UTLN (username, e.g. jsmit01) with login next to it, like this: [jsmith@login001 ~]$
	1. This means you are on a **login node** of the cluster. From here, you can either: 1) request an interactive node or 2) send a job to the scheduler (SLURM) using the commands srun or sbatch.
	2. To request an interactive node, type: srun -p interactive --pty --x11=first --mem=16000 bash
		1. srun is the command
		2. -p is an option to select a certain parition or grouping of nodes
		3. --pty means pseudo-terminal – you want to obtain a command line interface back, with x11 graphics functionality if you are using Fast X.
		4. --mem = 16000 16GB of RAM
		5. Bash – the bash shell is the command line to obtain
	3. To send a job, you can send the script directly using srun/sbatch.
4. After you request an interactive node (step 3b) you can run python scripts like normal.

## Learning Geospatial Python

Geospatial data is more complex, as it commonly consists of many files of formats. Examples of this are things like shapefiles, latitude and longitude within and Excel sheet, and remote sensing data. Luckily, we can use Python to work with these sorts of data.

* **Getting Started with Geospatial data:**
	+ **Textbook on Geospatial Analysis in Python:** <https://github.com/PacktPublishing/Mastering-Geospatial-Analysis-with-Python>
	+ **Introduction to Geospatial Data in R** (good for general approach to coding)**:** <https://datacarpentry.org/organization-geospatial/04-geo-landscape/index.html>
	+ **Using Geopandas in Python:** <https://www.hatarilabs.com/ih-en/introduction-to-spatial-analysis-in-python-with-geopandas-tutorial>
* **High-Performance Python for Geospatial applications:**
	+ **Examples of Scaling Geospatial Data Analysis with dask:** <http://matthewrocklin.com/blog/work/2017/09/21/accelerating-geopandas-1>
	+ **General parallel Python with async:** <https://training.talkpython.fm/courses/explore_async_python/async-in-python-with-threading-and-multiprocessing>
* **Creating raster layers with the Geodata Abstraction Library (GDAL):**
	+ **Documentation**: [https://pcjericks.github.io/py-gdalogr-cookbook/raster\_layers.html#](https://pcjericks.github.io/py-gdalogr-cookbook/raster_layers.html)

## Additional program documentation

For more information on the following programs, see the links below:

* **Which statistical software should I use?**
	+ <https://tufts.box.com/v/WhichStatisticalSoftwareToUse>
* **QGIS and Geospatial databases:**
	+ <https://tufts.box.com/v/UsingQGIS2019>
* **Tableau Desktop**
	+ <https://tufts.box.com/v/UsingTableau2018>
* **Tableau Server for interactive dashboards in the cloud**
	+ <https://tufts.box.com/v/UsingInstallationTableauServer>
* **NVivo for text analysi**s
	+ <https://tufts.box.com/v/NVivoVDIQualtricsSentiment>
	+ <https://tufts.box.com/v/UsingInstallingNVivo>
* **Stata for statistics**
	+ <https://tufts.box.com/v/UsingInstallingStata>
* **Choosing a computer for ArcMap**
	+ <https://tufts.box.com/v/ChoosingAComputerforGIS>
* **Using SPSS for statistics**
	+ <https://tufts.box.com/v/UsingInstallingSPSS>
* **RStudio for statistics**
	+ <https://tufts.box.com/v/UsingInstallingRStudio>