

Introduction to Python 101

Wikipedia:

“Python is an interpreted, high-level, general-purpose programming language... Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects”

These notes are intended to help you:

- Set up the Python environment
- Learn how to install additional packages
- Execute simple Python commands
- Run a first sample program

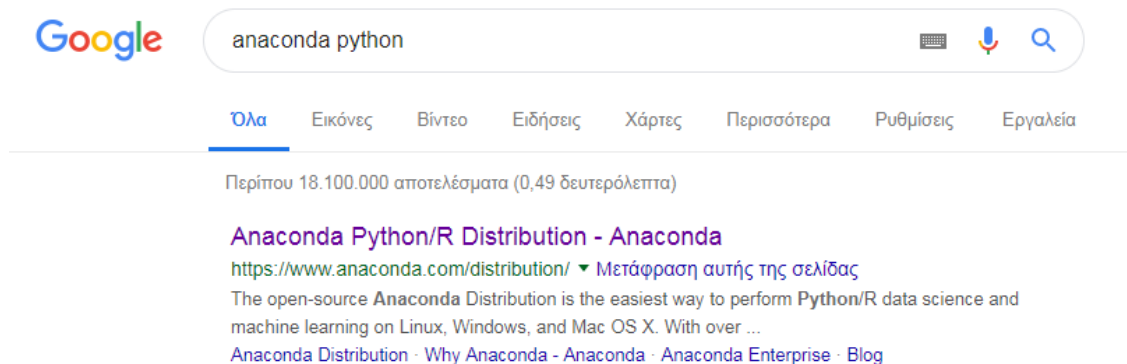
Set up the Python environment

The **core Python package** is easy to install, but not what you should choose for these lectures. The reason is that these lectures require the entire scientific programming ecosystem, which the core installation doesn't provide.

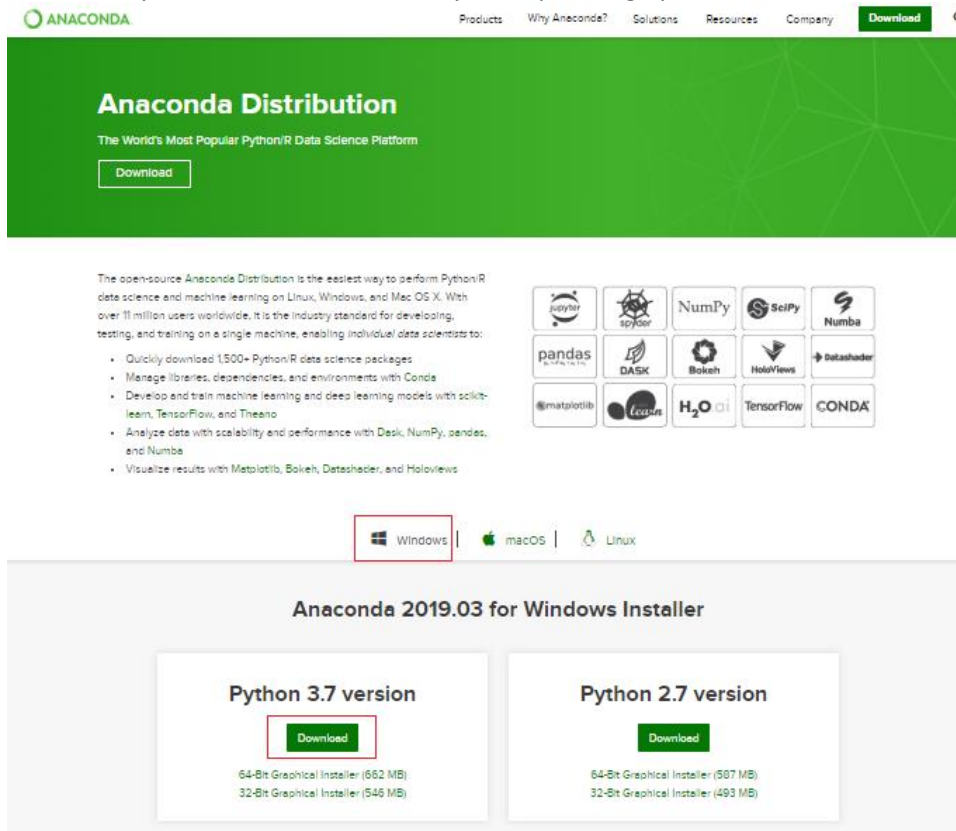
For this course purposes we choose to work with Anaconda which is one of the free Python distributions that contains:

1. The core Python language **and**
2. the most popular scientific libraries like:
 - NumPy that includes fundamental matrix and array processing capabilities
 - SciPy that provides additional functionality including
 - Linear algebra
 - Optimization
 - Distributions and random number generation
 - etc., etc.
 - Matplotlib for creating figures and graphs

At this point you will have to download the version of Anaconda that suits your operation system (Windows, MAC or Linux). Search for “anaconda python” and go to the first result:

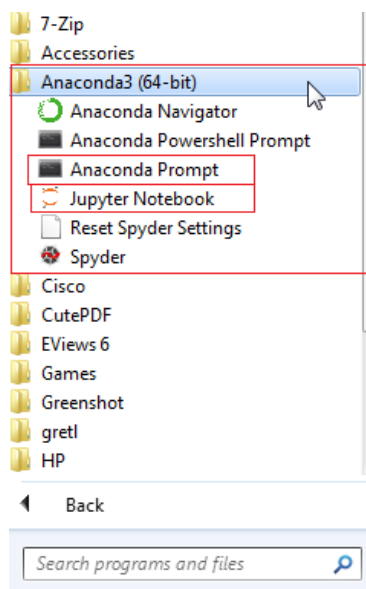


Make sure you have selected correctly the Operating System click Download.



The screenshot shows the Anaconda Distribution website. At the top, there's a navigation bar with links: Products, Why Anaconda?, Solutions, Resources, Company, and a green Download button. The main header features the Anaconda logo and the text "Anaconda Distribution" and "The World's Most Popular Python/R Data Science Platform". Below this is a green "Download" button. A paragraph describes the open-source distribution as the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. To the right, a grid of logos for various data science libraries is shown: Jupyter, Spyder, NumPy, SciPy, Numba, pandas, DASK, Bokeh, Holoviews, Daskml, matplotlib, Leaflet, H2O, TensorFlow, and CONDA. Below the text, there are three tabs for Windows, macOS, and Linux. The Windows tab is selected, showing the "Anaconda 2019.03 for Windows Installer". Under this, there are two options: "Python 3.7 version" and "Python 2.7 version". Each has a green "Download" button and lists the file sizes for 64-bit and 32-bit graphical installers.

Once the download is completed, proceed with installing the application. When completed you should be able to see the corresponding folder in your Programs list:



For the purposes of this course we will mainly be using 'Jupyter Notebook' and 'Anaconda Prompt'

Jupyter notebooks are one of the many possible ways to interact with Python that provide a browser-based interface to Python with

- The ability to write and execute Python commands directly in your browser
- Formatted output also in the browser, including tables, figures, animation, etc.
- The ability to mix in formatted text and mathematical expressions between cells

‘Anaconda Prompt’ is the environment that will allow us to install additional packages.

Learn how to install additional packages

For our purposes, you will need only to install QuantEcon which is a high performance, open source Python code library for economics. To do so:

Open ‘Anacond Prompt’ window, type:

Conda install –c conda-forge quantecon

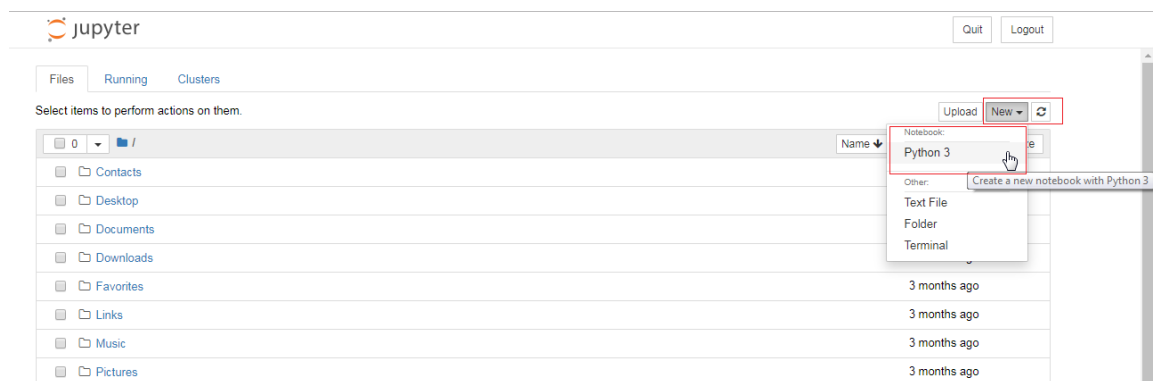
press enter and follow the instructions

Please refer to the following link for more information:

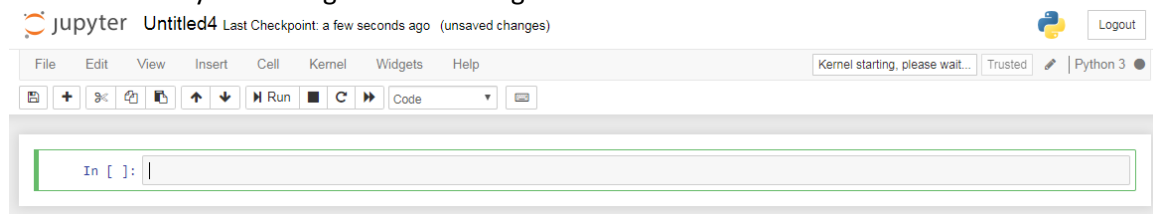
<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html>

Execute simple Python commands

Open ‘Jupyter Notebook’. This will open a new tab in your internet browser that will look like this:



Select New -> Python 3 to get the following:



This is the place where we will be writing our code.

Run a first sample program

Write the following code in the first cell block that appeared when you opened the notebook.

This sample code is intended to create an array that contains 100 random draws from the Standard Normal Distribution.

```
from random import normalvariate
import matplotlib.pyplot as plt
ts_length = 100
epsilon_values = [] # An empty list
for i in range(ts_length):
    e = normalvariate(0, 1)
    epsilon_values.append(e)
plt.plot(epsilon_values, 'b-')
plt.show()
```

To execute the code you can either press Shift + Enter or press Run.

This is what is supposed your tab to look like if everything has been executed correctly:

