

Wen.Gong@Oracle.com - IT Consultant

- taught python at Chapel Hill Chinese School (2012.9 - 2014.6)
 - web2py
 - pyGame

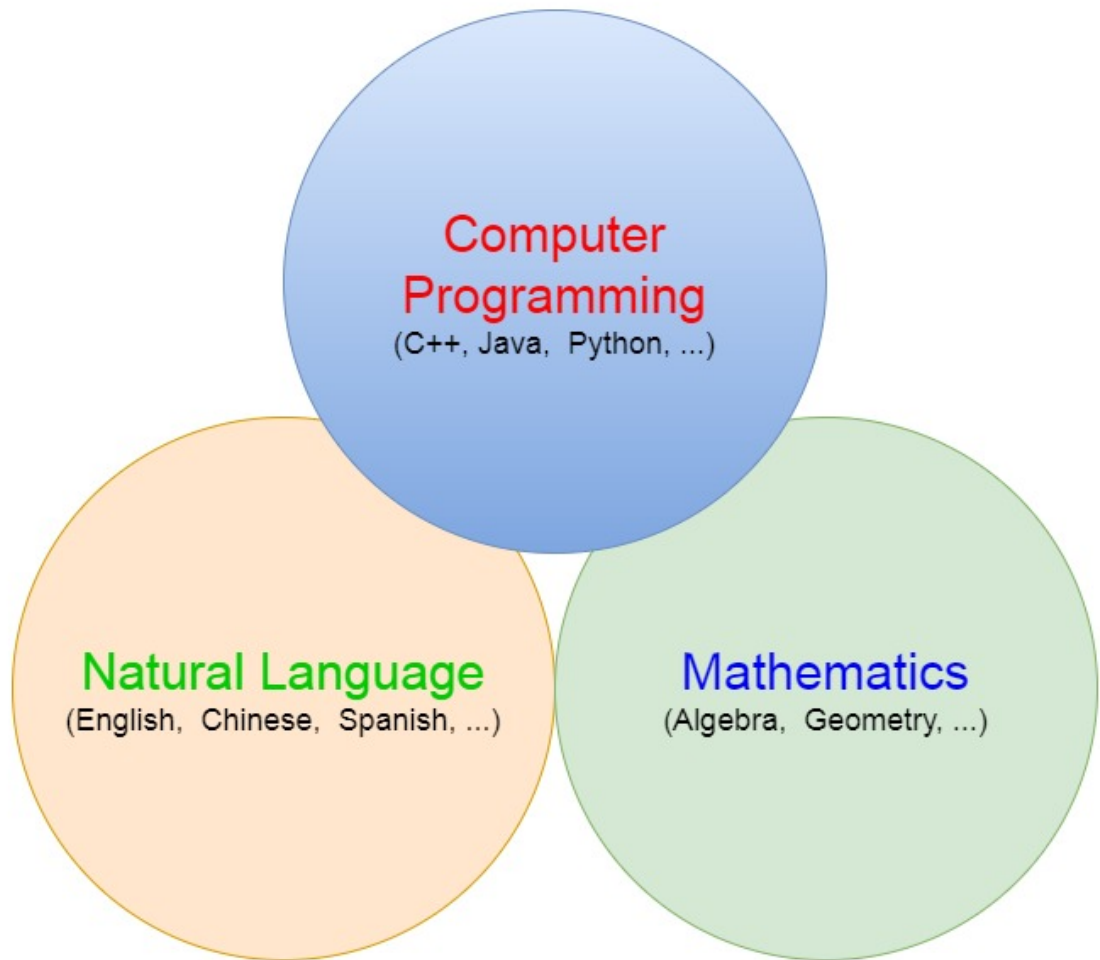
```
In [1]: from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

- Out[1]:
- Motivation
 - 3 Basic Language skills
 - Natural Language - Talk to People
 - Mathematics - Talk to Nature
 - Programming Lang - Talk to Computer
 - Why Python?
 - Popular programming langs
 - What is Python?
 - Doing Science in Python
 - Doing Math in Python
 - Doing Physics in Python
 - Doing Biology/Chemistry in Python
 - AI - Computer Vision
 - 4 ways to interact with Python
 - Online Demo
 - Command Console
 - Integ. Development Env (IDE) - Spyder
 - Jupyter Notebook
 - Learning Plan
 - Install Anaconda

Motivation

3 Basic Language skills

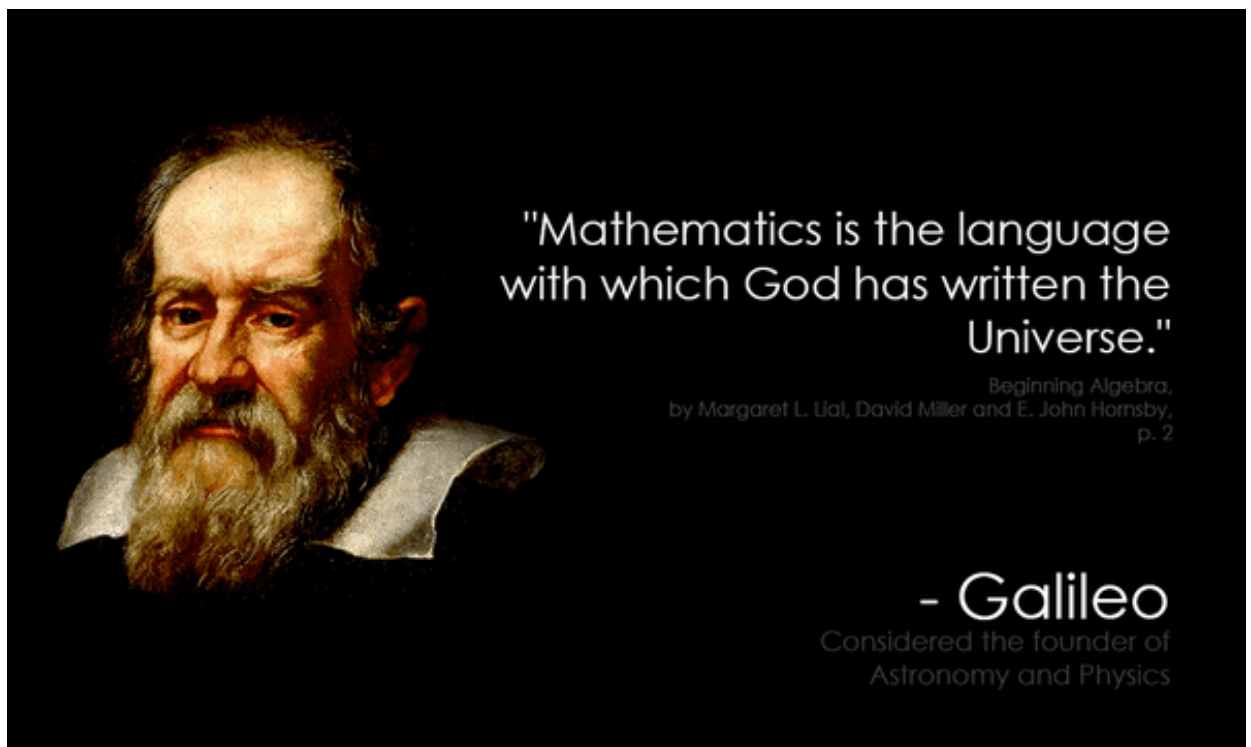
3 basic language skills



Natural Language - Talk to People



Mathematics - Talk to Nature



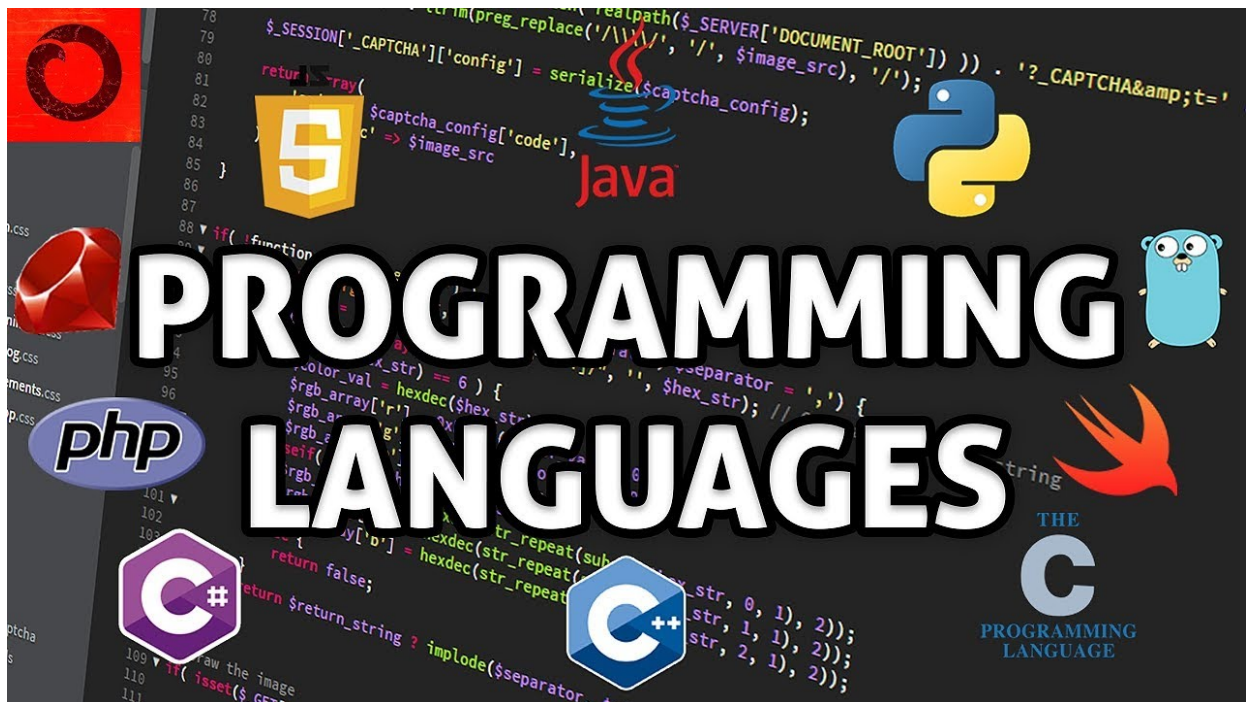


Programming Lang - Talk to Computer

Google DeepMind AI beats world chess master



Why Python?



Popular programming langs

The TIOBE Programming Community index is an indicator of the popularity of programming languages.

<https://www.tiobe.com/tiobe-index/> (<https://www.tiobe.com/tiobe-index/>)

Programming Language	2017	2012	2007	2002	1997	1992	1987
Java	1	1	1	1	15	-	-
C	2	2	2	2	1	1	1
C++	3	3	3	3	2	2	5
C#	4	4	7	14	-	-	-
Python	5	7	6	11	27	-	-
Visual Basic .NET	6	19	-	-	-	-	-
PHP	7	6	4	5	-	-	-
JavaScript	8	9	8	8	22	-	-
Perl	9	8	5	4	4	10	-
Assembly language	10	-	-	-	-	-	-
COBOL	25	27	17	9	3	9	9
Lisp	31	12	15	12	9	5	2
Prolog	33	31	26	16	20	11	3
Pascal	112	14	19	97	8	3	4

[A Comparison of Programming Languages \(https://fusion809.github.io/comparison-of-programming-languages/\)](https://fusion809.github.io/comparison-of-programming-languages/)

What is Python?

Python

Python (`dev-lang/python` [package]: [ES](#), [GPO](#), [PT](#); `dev-python` [category]: [ES](#), [GPO](#), [PT](#), [WP](#)) is a widely and extensively-used high-level general-purpose multi-paradigm programming language that is particularly invaluable as a cross-platform scripting language. It is named after the BBC TV series *Monty Python's Flying Circus*. Python is licensed under its own free, permissive (BSD-like) license called the [Python Software Foundation License](#). It is probably the most flexible programming language I have seen and it is used for numerical computations, scientific computing, writing and working with web applications, application software and package management systems. Its design philosophy emphasizes code readability and concision. It also automatically performs some tasks that users would have to perform manually if they were working with lower-level languages like C. For these reasons it, and JavaScript, are usually the programming languages that people interested in programming, are recommended to learn first, before they learn more complicated programming languages like C, C++ and Java. Its major caveat is its speed, however, it is worthwhile noting that its speed is significantly dependent on how it is implemented. The standard, official implementation of Python is [CPython](#) which while it is more efficient than most implementations (like Jython), is less efficient than the [PyPy](#) implementation. Regardless of the implementation used, however, it is usually significantly slower than compiled languages like C, C++ and Java.


There are also two main versions of Python presently in widespread use: Python 2 and Python 3. Most programs I have come across have greater support for Python 2 than for Python 3. Python also has its own command-line package manager called [pip](#) (`pip`). Some programs I am particularly familiar with that are written predominantly (if not exclusively) in Python that are compatible with Linux systems include:



- [Anaconda](#) — an operating system installer used by Fedora and most of its derivatives, see the [Fedora](#) section for details.
- [DNF](#) — the default package manager of Fedora ≥22.
- [Entropy](#) — the default binary package manager of Sabayon Linux.

Doing Science in Python


SciPy.org



[Install](#)

[Getting Started](#)

[Documentation](#)

[Report Bugs](#)

[SciPy Central](#)

[Blogs](#)

SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:

 NumPy Base N-dimensional array package	 SciPy library Fundamental library for scientific computing	 Matplotlib Comprehensive 2D Plotting
 IPython Enhanced Interactive Console	 Sympy Symbolic mathematics	 pandas Data structures & analysis

Doing Math in Python

- [SageMath](#) — a mathematics program that uses a Python-like syntax and integrates several free pieces of mathematics software in a single command-line and notebook interface, with interfaces available for several proprietary mathematics programs too.

The screenshot shows the Sage Notebook interface. At the top, there's a header with the Sage logo and "The Sage Notebook" text. Below that, the notebook title is "Lorenz equations". The interface includes a menu bar with options like "File", "Action", "Data", and "sage". There are also buttons for "Typeset", "Load 3-D Live", and "Use java for 3-D". On the right side, there are buttons for "Print", "Worksheet", "Edit", "Text", "Revisions", "Share", and "Publish". The main area contains a Python script that defines parameters, solves the Lorenz equations, and plots the result as a 3D wireframe plot.

```

x,y,z=var('x,y,z')

# Next we define the parameters
sigma=10
rho=40
beta=8/3

# The Lorenz equations
lorenz=[sigma*(y-x),x*(rho-z)-y,x*y-beta*z]

# Time and initial conditions
N=250000
tmax=100
h=tmax/N
times=srange(0,tmax+h,h)
ics=[0,1,1]
sol=desolve_odeint(lorenz,ics,times,[x,y,z],rtol=1e-13,atol=1e-14)
X=sol[:,0]
Y=sol[:,1]
Z=sol[:,2]

# Plot the result
from mpl_toolkits.mplot3d import axes3d
from matplotlib import pyplot as plt
def plot1():
    fig = plt.figure(1)
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_wireframe(X, Y, Z, rstride=10, cstride=10)
    ax.set_xlabel('X(t)')
    ax.set_ylabel('Y(t)')
  
```

Doing Physics in Python

SymPy 1.1.1 documentation » SymPy Modules Reference »




Table Of Contents

[Physics Module](#)

- [Contents](#)

Previous topic

[Calculus](#)

Next topic

[Hydrogen Wavefunctions](#)

This Page

[Show Source](#)

Quick search

Physics Module

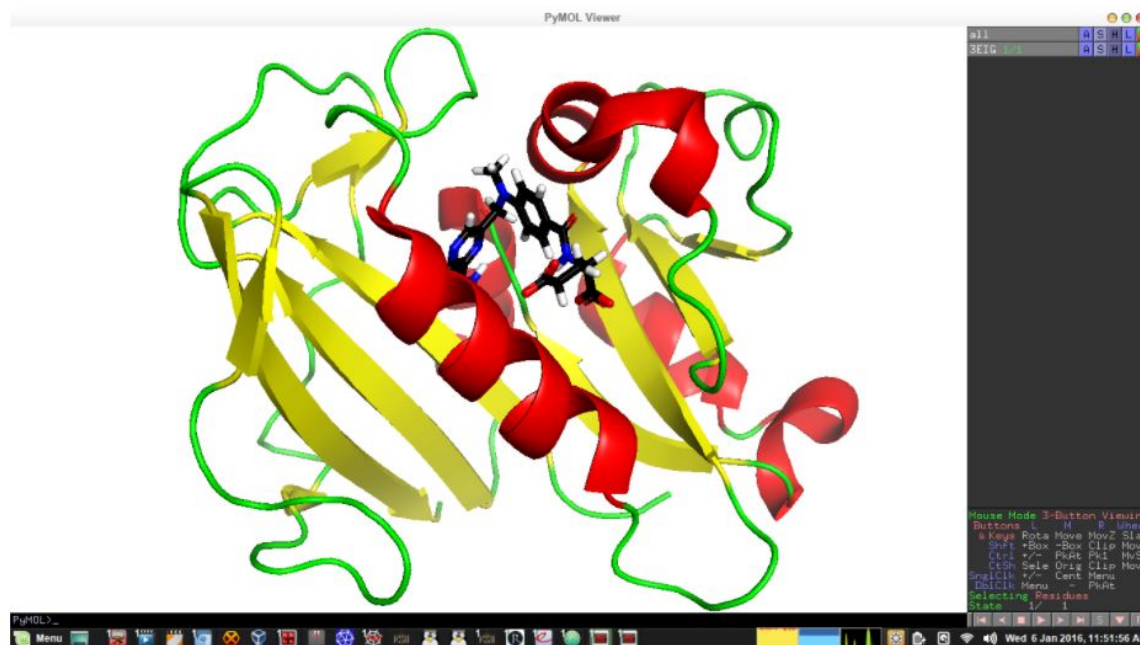
A module that helps solving problems in physics

Contents

- [Hydrogen Wavefunctions](#)
- [Matrices](#)
- [Pauli Algebra](#)
 - [References](#)
- [Quantum Harmonic Oscillator in 1-D](#)
- [Quantum Harmonic Oscillator in 3-D](#)
- [Second Quantization](#)
- [Wigner Symbols](#)
 - [References](#)
 - [Credits and Copyright](#)
- [Unit systems](#)
 - [Philosophy behind unit systems](#)
 - [Dimensions](#)
 - [Quantities](#)
 - [The need for a reference](#)
 - [Literature](#)
 - [Examples](#)
 - [Dimensional analysis](#)
 - [Equation with quantities](#)
 - [Dimensions and dimension systems](#)
 - [Unit prefixes](#)
 - [Units and unit systems](#)
 - [Physical quantities](#)
 - [Conversion between quantities](#)
- [High energy physics](#)
 - [Gamma matrices](#)

Doing Biology/Chemistry in Python

- **PyMOL** — an application for viewing and performing calculations relating to molecules, especially macromolecules like proteins, DNA double helices and RNA helices. Which is managed by pip, so running `user...$ sudo pip install pymol` should install it.



AI - Computer Vision

[Caffe Demo \(http://demo.caffe.berkeleyvision.org/classify_upload\)](http://demo.caffe.berkeleyvision.org/classify_upload)

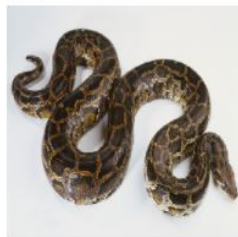
[EVA System \(http://www.image-net.org/eva/\)](http://www.image-net.org/eva/) (not working)

Caffe Demos

The **Caffe** neural network library makes implementing state-of-the-art computer vision systems easy.

Classification

[Click for a Quick Example](#)



Maximally accurate

Maximally specific

diapsid

1.55305

reptile

1.48065

snake

1.44930

colubrid snake

0.94117

boa

0.61518

4 ways to interact with Python

Online Demo

Need to install, access it anywhere, may not work all the time.

The screenshot shows the learnpython.org website. The browser address bar displays 'learnpython.org/en/Variables_and_Types'. The page has a navigation bar with links like 'Welcome / Variables and Types'. Below the navigation bar, there's a section titled 'Variables and Types' with a subtitle 'Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors. Start Now!'. There are buttons for 'Previous Tutorial' and 'Next Tutorial'. The main content area starts with the heading 'Variables and Types' and a paragraph explaining that Python is completely object oriented and not statically typed. It then introduces the 'Numbers' section, stating that Python supports integers and floating point numbers. A code editor shows a script named 'script.py' with two lines: '1 myint = 7' and '2 print(myint)'. To the right of the script is an 'IPython Shell' window showing the output of the script: 'In [2]: type(myint)' followed by 'Out[2]: int'. A 'Run' button is located below the script editor. The page is powered by DataCamp.

Command Console

Need basic DOS / Unix knowledge and skill

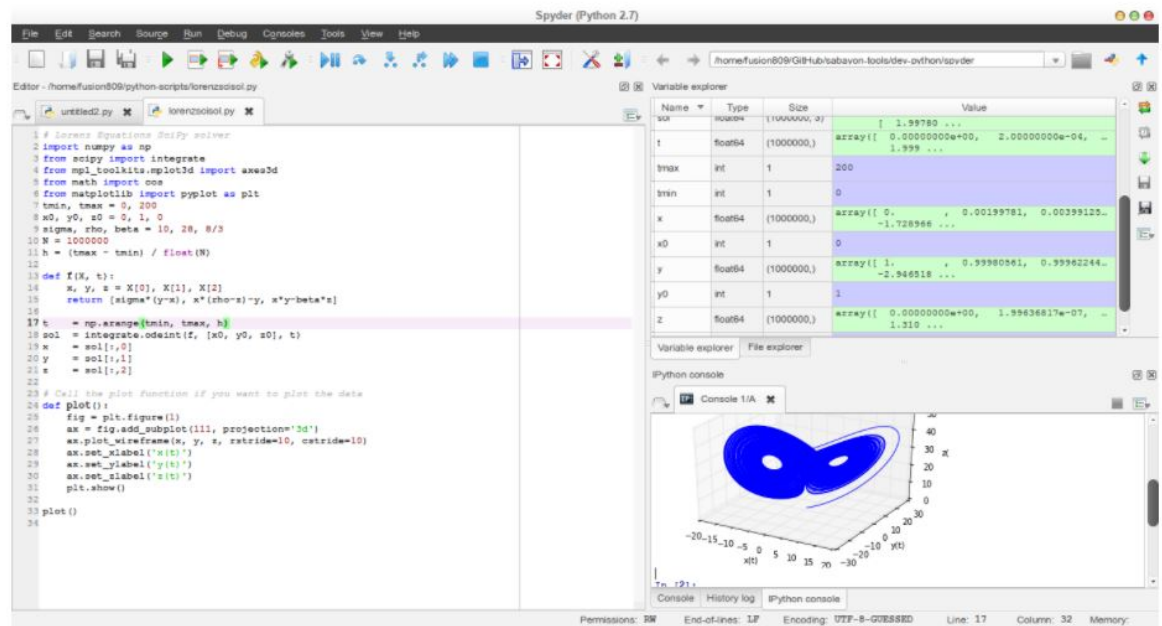
The screenshot shows a Windows Command Prompt window with the following text:

```
C:\Users\WGONG>python
Python 3.5.2 |Anaconda custom (64-bit)| (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 1+2
3
>>>
>>> print("Hello Python")
Hello Python
>>>
>>> import math
>>> print(math.pi)
3.141592653589793
>>>
>>> print(math.sqrt(2))
1.4142135623730951
>>>
>>> answer = True
>>>
>>> type(answer)
<class 'bool'>
>>>
>>> type(1.23)
<class 'float'>
>>>
>>> quit()
C:\Users\WGONG>
```

Integ. Development Env (IDE) - Spyder

Productivity tool for professional developers (syntax highlighting, debugging, dependency, ...)


- **Spyder** — a cross-platform free and open-source IDE for scientific computing in Python. It is managed by pip, so running `user@...$ sudo pip install spyder` should install it, provided its dependency PyQt4/PyQt5 is already installed.



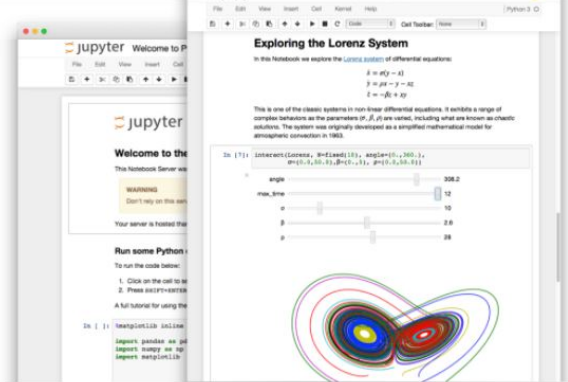
Jupyter Notebook

Python in a browser, web-server running locally.

- We will use notebook for instructions and assignments




[Install](#)
[About](#)
[Community](#)
[Documentation](#)
[NBViewer](#)
[Widgets](#)
[Blog](#)
[Donate](#)




The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.




Language of choice

The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.




Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the [Jupyter Notebook Viewer](#).



Interactive widgets

Code can produce rich output such as images, videos, LaTeX, and JavaScript. Interactive widgets can be used to manipulate and visualize data in realtime.



Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, dplyr, etc.

Learning Plan

1. Lectures in ZOOM cloud meeting
2. Hand-on lab with TA (Teaching Assistant)
3. Homework
4. Study & Practise on your own

[learn python : teaching plan](#)

https://docs.google.com/spreadsheets/d/1AWb_c5c8adzWIC_BLRKcKqr3-IR-JBQDP5HMJE3PPaU/edit?usp=sharing

Install Anaconda (<https://www.anaconda.com/>)



ANACONDA.
The Most Popular Python Data Science Platform

4.5M+	1,000+	150+
Users	Data Science Packages	Enterprise Customers

With over 4.5 million users, Anaconda is the world's most popular Python data science platform. Anaconda, Inc. continues to lead open source projects like Anaconda, NumPy and SciPy that form the foundation of modern data science. Anaconda's flagship product, Anaconda Enterprise, allows organizations to secure, govern, scale and extend Anaconda to deliver actionable insights that drive businesses and industries forward.

ANACONDA DOCUMENTATION
Powered by Continuum Analytics

Search

Anaconda Platform
Welcome
▼ Anaconda Distribution
Anaconda Navigator or conda?
Packages available in Anaconda
High performance
Previous versions
Product archive
Installation
System requirements
Silent mode install
Installing Anaconda on a non-networked machine
Detailed installation information
Installing on Windows

[Home](#) > [Anaconda Distribution](#) > Installation

INSTALLATION

On Windows, macOS, and Linux, it is best to install Anaconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, if you need to, you can install Anaconda system wide, which does require administrator permissions.

TIP: If you don't want the hundreds of packages included with Anaconda, you can [install Miniconda](#), a mini version of Anaconda that includes just conda, its dependencies and Python.

TIP: Looking for Python 3.5? See our [FAQ](#).

System requirements

- License: Free use and redistribution under the terms of the [Anaconda End User License Agreement](#).
- Operating system: Windows Vista or newer, macOS 10.7+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others.
- Windows XP supported on Anaconda versions 2.2 and earlier. See [Old package lists](#). Download it from our [archive](#).
- System architecture: 64-bit x86, 32-bit x86, or Power8.
- Minimum 3 GB disk space to download and install.

In []: