

Urban5123: Programming Tools for Urban Analytics

Introduction to the Python Coding Environment

Qunshan Zhao
Senior Lecturer in Urban Analytics
9 January 2023

Course Overview

Programming Tools for Urban Analytics

Lecture 1 (09/01/23)	Introduction to the Python Coding Environment	Qunshan Zhao
Lecture 2 (16/01/23)	Unix Shell, Git and GitHub	Qunshan Zhao
Lecture 3 (23/01/23)	Python Basics	Qunshan Zhao
Lecture 4 (30/01/23)	Python Advanced	Qunshan Zhao
Lecture 5 (06/02/23)	Python for Scientific Computing and Data Science	Qunshan Zhao

Course Overview

Programming Tools for Urban Analytics

Lecture 6 (13/02/23)	Python for Visualization	Qunshan Zhao
Lecture 7 (20/02/23)	Advanced Database Manipulation with Python	Qunshan Zhao
Lecture 8 (27/02/23)	API, Online Data Extraction/Scraping by Python	Qunshan Zhao and Nick Ves
Lecture 9 (06/03/23)	Python for Machine Learning and Deep Learning	Qunshan Zhao and Faraz Awan

Course learning outcomes

- Write code according to best practice and produce tidy data (Tools: Python)
- Collaborate effectively with other analysts using appropriate tools (Tools: Git and GitHub)
- Produce documentation for their work which makes the processes behind analyses transparent and reproducible. (Tools: Unix Shell, GitHub, Jupyter Notebook)
- Set up, connect to and query a simple relational and non-relational database (Tools: Python, SQL, MongoDB)
- Retrieve and analyse data from an Application Protocol Interface (API) (Tools: Python)
- Perform basic machine learning tasks (Tools: Python)

Classroom and after-class expectation

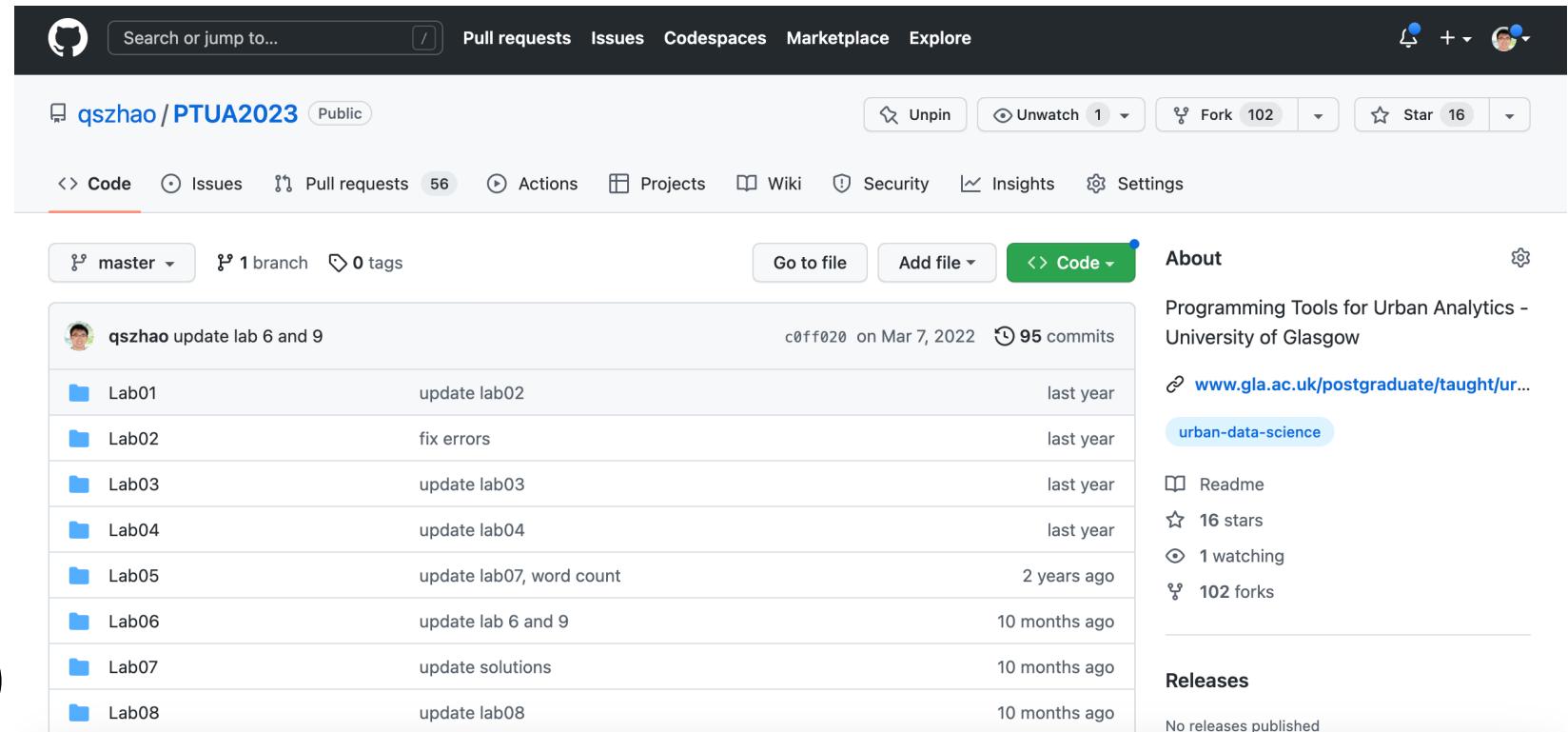
- Read the essential readings before each of the lecture;
- Engage in the classroom discussion and activities
- Actively engaged in the in-class lab session;
- Practise coding through the Data Camp after classes (**very important!**);
- Actively thinking and questioning the course content;
- Course tutor: Yang Wang (Yang.Wang@glasgow.ac.uk)

Assessment

- Formative assessment
 - Due on Monday the **6th February**, one-page data science report proposal
 - It does not count for the final grade, but helps you get feedbacks from us
- Summative assessment
 - Due on Monday the **27th March**, final individual report in HTML file (generated from the Jupyter Notebook) (**3,000 words** not including code and reference) along with original Jupyter Notebook.
 - Detailed guidance is provided on Moodle.
 - It counts for 100% of the final grade

Course materials

- Moodle: lecture slides and all other course related documents
- GitHub repository: lecture notebooks, lab exercises (<https://github.com/qszhao/PTUA2023>)



What is open source?

Open source products include permission to use the source code, design documents, or content of the product. It most commonly refers to the open-source model, in which open-source software or other products are released under an open-source license as part of the open-source-software movement. Use of the term originated with software, but has expanded beyond the software sector to cover other open content and forms of open collaboration.

Source:

https://en.wikipedia.org/wiki/Open_source

PySAL – the Python Spatial Analysis Library



The Python Spatial Analysis Library
for open source, cross platform
Geospatial Data Science

 Lib Core spatial data structures, file IO. Construction and interactive editing of spatial	 Explore Modules to conduct exploratory analysis of spatial and spatio-temporal data	 Model Estimation of spatial relationships in data with a variety of linear, generalized-	 Viz Visualize patterns in spatial data to detect clusters, outliers, and hot-spots
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Source: <https://pysal.org/>

PySAL:spopt – Spatial Optimization

☰ README.md



PySAL
Python Spatial Analysis Library

spopt : Spatial Optimization

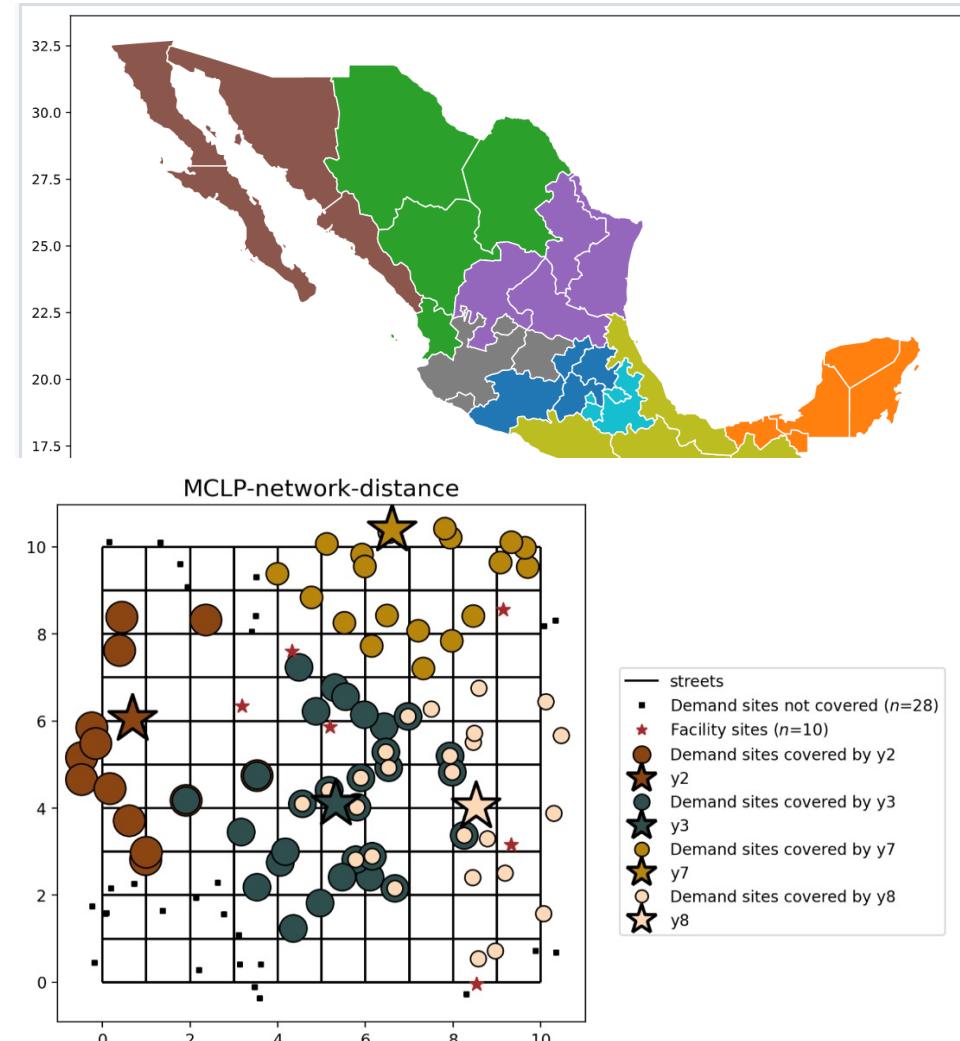
Regionalization, facility location, and transportation-oriented modeling

release v0.4.1 CI Continuous Integration passing codecov 74% docs current License BSD 3-Clause code style black
JOSS 10.21105/joss.03330 DOI 10.5281/zenodo.4444156

Spopt is an open-source Python library for solving optimization problems with spatial data. Originating from the `region` module in [PySAL \(Python Spatial Analysis Library\)](#), it is under active development for the inclusion of newly proposed models and methods for regionalization, facility location, and transportation-oriented solutions.

Source: <https://github.com/pysal/spopt>

<https://www.ubdc.ac.uk/news-media/2021/september/open-source-facility-location-modelling-spopt-development/>



Group Discussion

Why open source?

Post your answer to Padlet:

<https://padlet.com/1990zqs/6foi5d4ctu0zguh3>



Why open source?

To me:

- It's **free!**
- It's required by your future employer!
- You can learn how to code and write an algorithm directly by reading the original programming code (**no more Blackbox anymore**).
- Easy collaboration (open-source coding, open-source book, etc..).
- Increase visibility, replicability, and reproducibility of the projects.
- Reduce duplication of money/time efforts.
- No commercial license requirement so no limitations after you graduate from the university or change your jobs.

And many more....

Linux – an open source operating system

- A family of open source **Unix-like** operating systems based on the Linux kernel.
- Popular Desktop Linux distributions include **Debian** and **Ubuntu**. **Android** is made from Linux kernel as well.
- You can control a Linux system all by it's Linux Shell (Terminal).

Source:

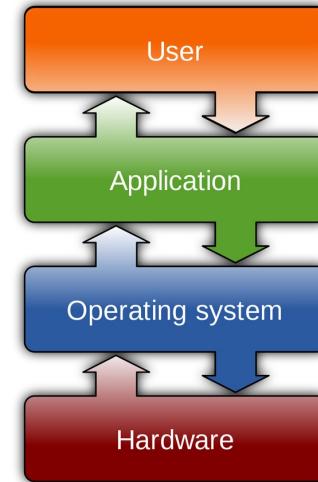
<https://medium.com/@abeysinghechamath/writing-your-own-operating-system-2e0909938249>

https://en.wikipedia.org/wiki/Operating_system



How about MacOS and Windows?

- MacOS is based on the **Unix** operating system as well (same as Linux), so they share the similar shell/terminal components with Linux.
- Windows dominates the desktop operating system. Its terminal system is very different from Mac and Linux!



Source:

<https://medium.com/@abeysinghechamath/writing-your-own-operating-system-2e0909938249>

https://en.wikipedia.org/wiki/Operating_system

Something about Python

- Python is a high-level interpreted language.
- Python is interactive.

```
Python 3.4 Python (command line)
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 3
>>> a
3
>>> b = 4
>>> b
4
>>> a+b
7
>>> a*b
12
>>>
```

- Python is easy to learn and read
- Python is one of the most popular programming language now

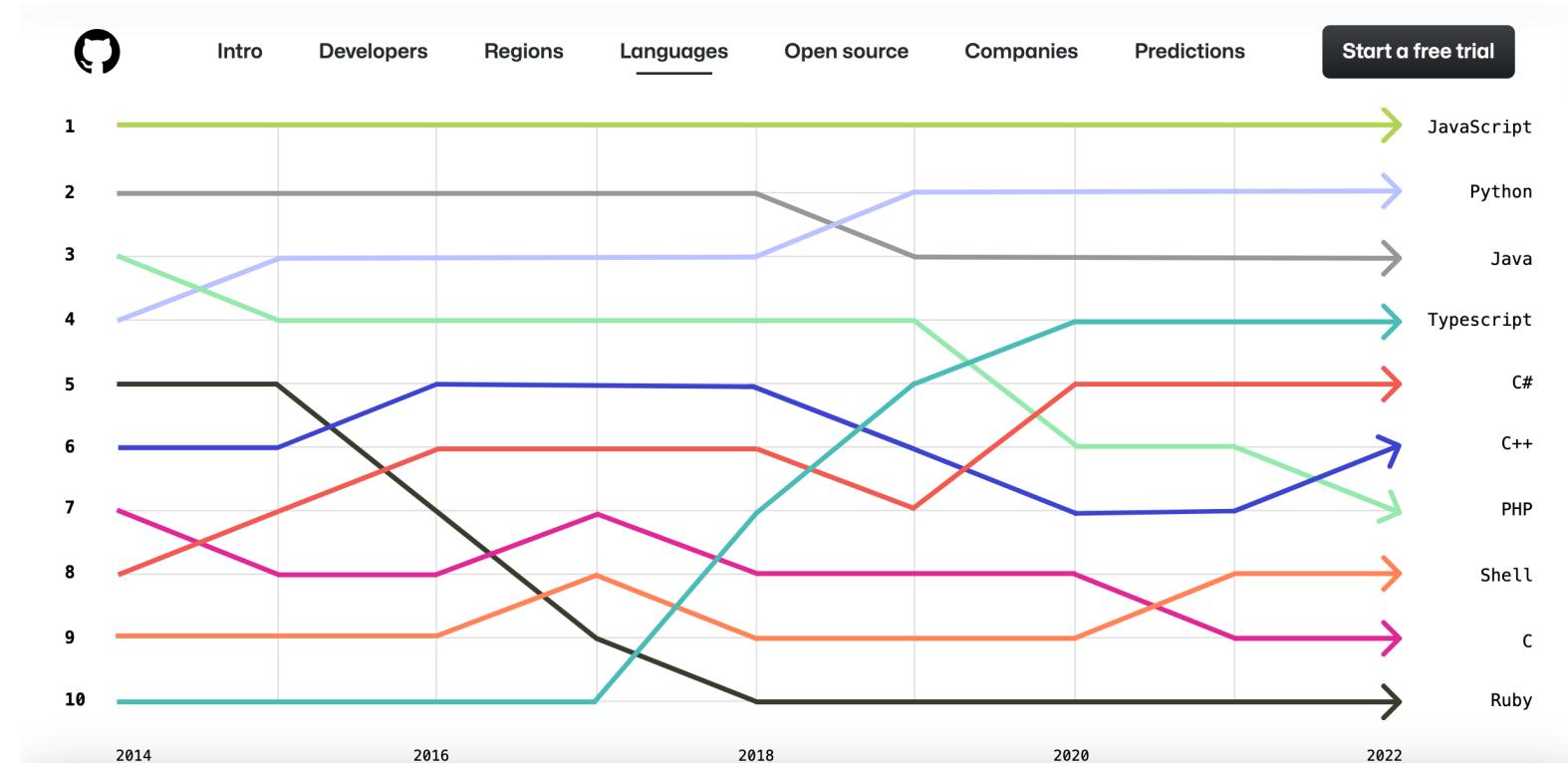


Why Python?

Top languages over time

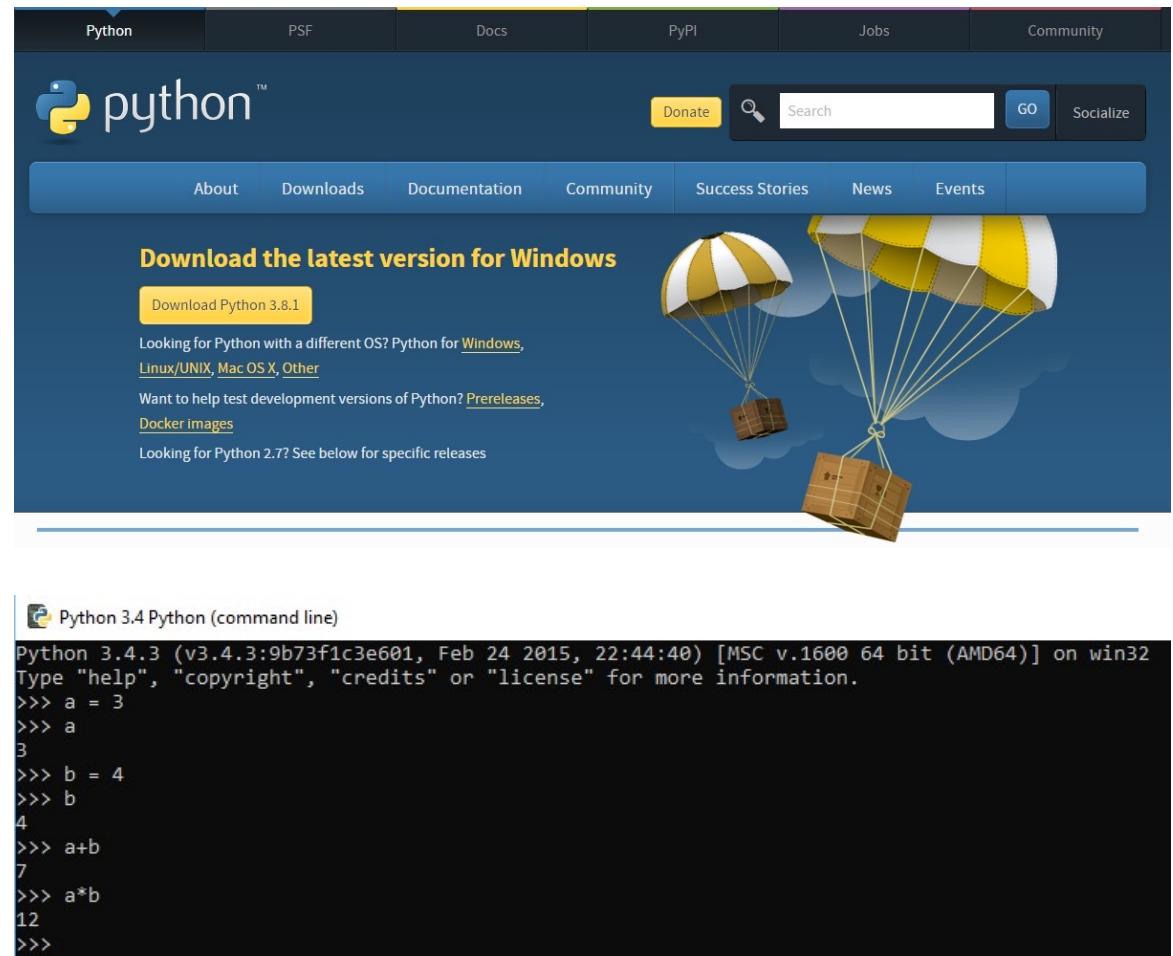
in GitHub:

- In 2022, Python still ranks as the **second most popular language on GitHub** by repository contributors.



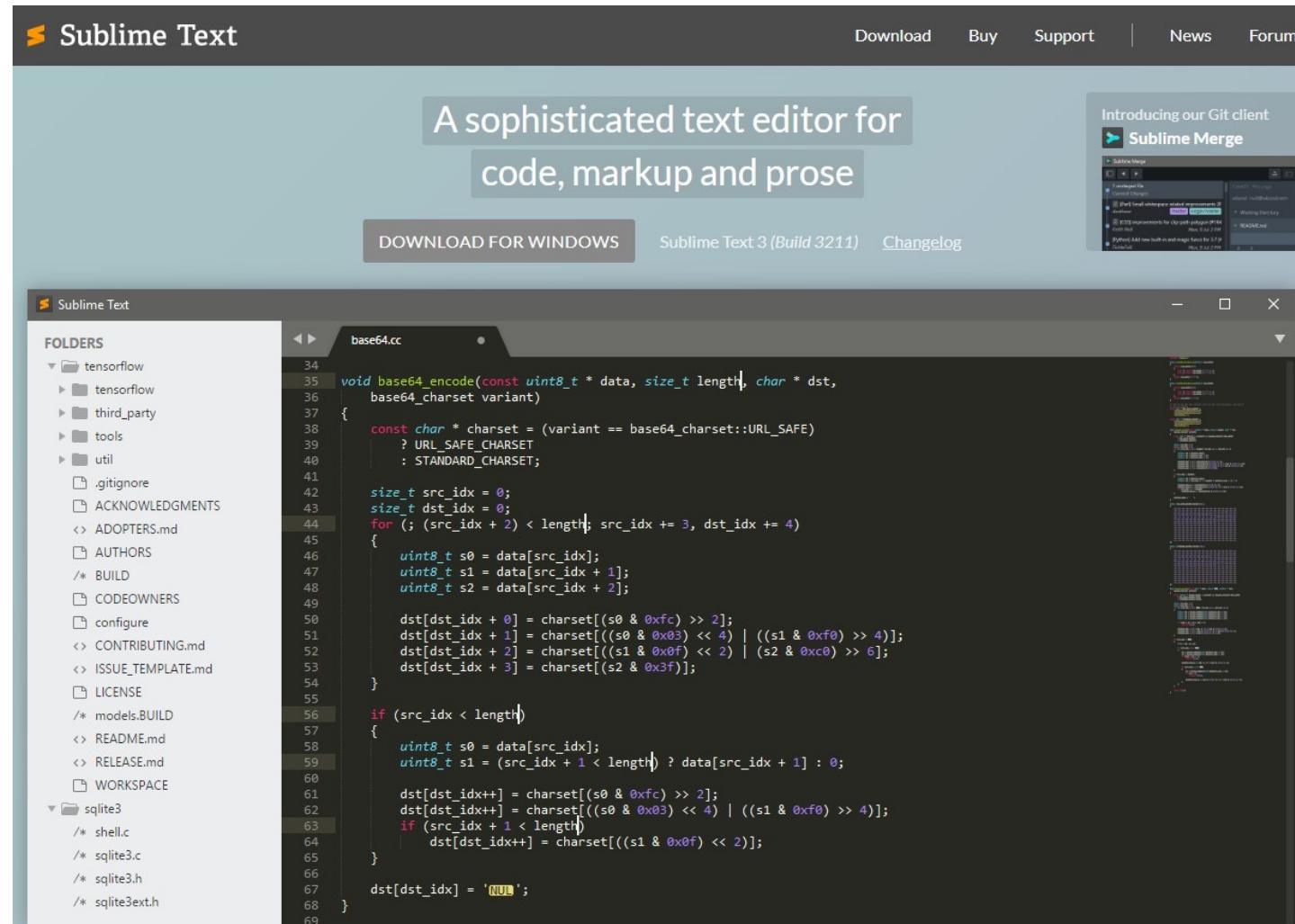
How to start with Python quick and dirty?

- You can download Python from its official website
- You would better to start with Python 3, since Python 2 is not maintained anymore after Dec 31st, 2019.
- You may still see old Python 2 packages and use it.



You may need a Code/Text Editor

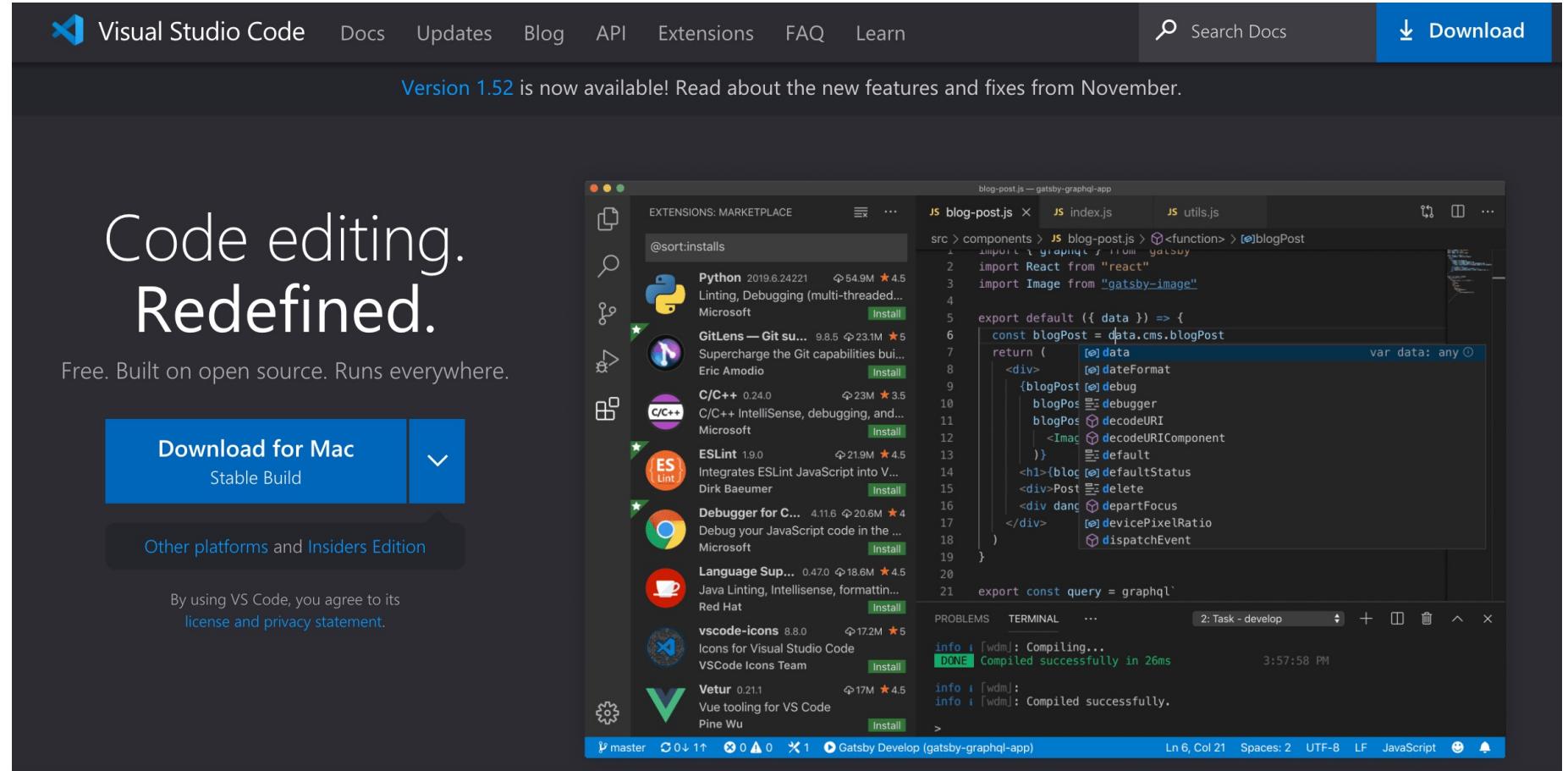
Sublime text:



Source: <https://www.sublimetext.com/>

You may need a Code/Text Editor

VScode:



Source: <https://code.visualstudio.com/>

More complicated? You may need an IDE

Integrated development environment (IDE)

- An IDE is a **software application** that provides comprehensive facilities to computer programmers for software development;
- An IDE normally consists of at least a **source code editor**, **build automation tools** and a **debugger**.
- Sometimes a **version control system** or a **graphical user interface (GUI)** are integrated

IDE for Python

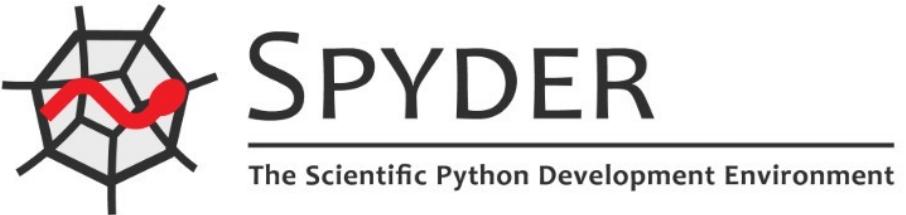
Pycharm:

A screenshot of the PyCharm IDE interface. The top navigation bar shows the project name "djtp_first_steps" and file tabs for "tests.py", "models.py", and "admin.py". The main code editor displays Python test code for a Django application. A search bar at the top right contains the text "result". To the right of the editor is a "Database" panel titled "Django default" which lists tables like "auth_group", "auth_permission", and "django_admin_log". At the bottom of the interface are several toolbars and status indicators, including "Debug", "Console", "Variables", "Watches", and "Event Log".

Source: <https://www.jetbrains.com/pycharm/>

IDE for Python

Spyder:



File Edit Search Source Run Debug Console Projects Tools View Help

C:\Users\Tess\Documents\spyder - Spyder (Python 3.8)

Project explorer

Spyder

Data

Editor

File

gitHub

codeScope

continuousIntegration

doc

img_src

requirements

root_profiling

scripts

src

tests

app

lens

__init__.py

cl_options.py

mac_stylesheet.py

pygmentsdev.py

rottext.py

skewify

tour.py

config

defaults

fonts

images

plugins

utils

widgets

windows

workers

zippy

dependencies

interpretipy

orthopy

plotipy

pycompipy

pygplot

pyplotipy

pytplotipy

styler_ipy

styler_ipython

styler_ipydm

styler_ipyJDF

styler_pynotebook

checkpoints

coocortex

coocortex_ipynb

coocortex_jml

temp.py interpolation.py __init__.py und_heber.py und_mean.py README.md

Editor C:\Users\Tess\Documents\16740156420fb7bb2abf7c\eebase4-51407aca17b0f5768950774bd1d99367b3174012420b67bb86SciCeebase4-551407aca17b0f5768950774bd1d99367b

Outline

Variable explorer

Name Type Size Value

bars container.BarContainer 26 BarContainer object of matplotlib.cont...

df DataFrame (3, 2) Column names: bools, ints

filename str 1 C:\ProgramData\Anaconda3\lib\site-pac...

list_text list 2 [DataFrame, Numpy array]

rows int 1 344

r float64 1 7.4110005000000005

radii float64 (20,) Min: 0.49820000000000007 Max: 9.85684074042551

region tuple 2 (slice, slice)

rgb float64 (45, 45, 4) Min: 0.0 Max: 1.0

series Series (3,) Series object of pandas.core.series.Se...

test_name NoneType 1 NoneType object of builtin module

Variable explorer Help File explorer Find in file Breakpoints Statistic analysis Profiler Online help

Python console

Console (A) Console (B) Custom namespace

```
In [1]: In the rgh colors of the shaded surface calculated from shade.
....: rgh = ls.shade(z, cnmaps.gist_earth, vert_exag=0.3, blend_mode='soft')
....: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgh,
....:                         linewidth=0, antialiased=False, shade=True)
....:
....: plt.show()
```

90°

135°

45°

180°

225°

315°

270°

-84.4 84.42 -94.39 64.20

72 77 73 71 73 71

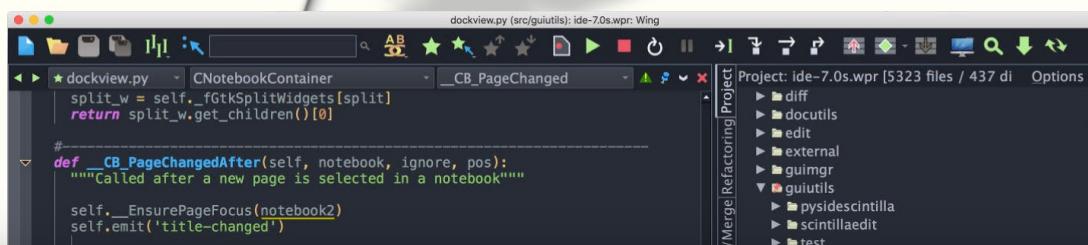
400 450 500 550 600 650 700

Revisions: RW End-of-line LF Encoding: UTF-8 Line: 26 Column: 4 Memory: 48% CPU: 36%

Source: <https://www.spyder-ide.org/>

IDE for Python

Wing:



The screenshot shows the Wing Python IDE interface. On the left, there's a code editor window titled "dockview.py (src/guiutils); ide-7.0s.wpr: Wing" containing Python code. The code includes methods like `__CB_PageChanged` and `__CB_PageChangedAfter`. To the right of the code editor is a "Project" browser pane listing various Python packages and files. A large feather graphic is positioned in the center-right area of the interface.

WING PYTHON IDE
THE INTELLIGENT DEVELOPMENT ENVIRONMENT FOR PYTHON

Wing Python IDE was designed from the ground up for Python, for a more productive development experience.

TRY WING PRO

Current version: 9.0.2

Get More Done

Type less and let Wing worry about the details. Get immediate feedback by writing your Python code interactively in the live runtime. Easily navigate code

"The best Python IDE. And I have tried them all!"
Ahmed Ali

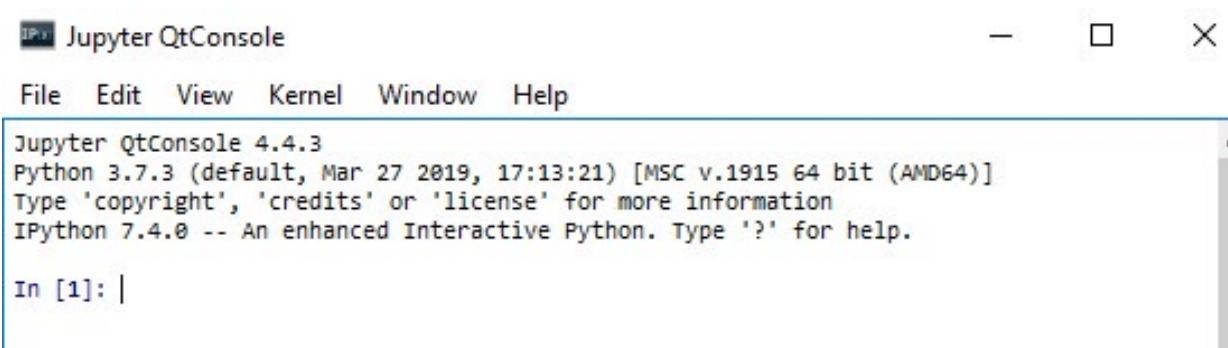
"I just keep lovin' it! Best Python IDE ever!"
Diogo Baeder

[Testimonials & Reviews](#)
[Why Wing Pro?](#)

Better Python - IPython

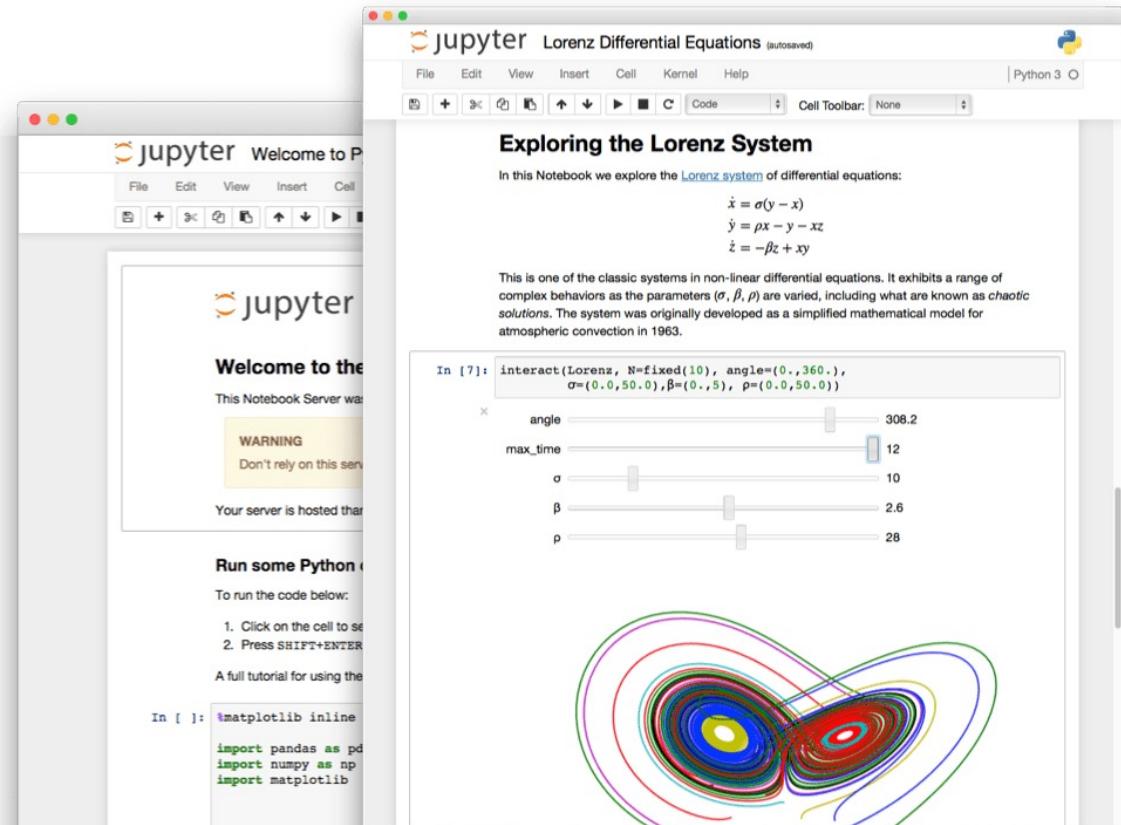
IPython (Interactive Python) is a [command shell](#) for interactive computing in multiple programming languages, originally developed for the [Python programming language](#), that offers [introspection](#), [rich media](#), shell syntax, [tab completion](#), and history. IPython provides the following features:

- Interactive shells (terminal and [Qt](#)-based).
- A browser-based [notebook interface](#) with support for code, text, mathematical expressions, inline plots and other media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into one's own projects.
- Tools for [parallel computing](#).



IPython Notebook – Now Jupyter Lab/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 narrative text.
- Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



Source: <https://jupyter.org/>

Jupyter Notebook

- You can write code, markdown, and Latex in the Jupyter Notebook together.

The image shows two side-by-side Jupyter Notebook interfaces. Both have a top navigation bar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a Python 3 kernel selection. A 'Logout' button is also present. Below the top bar is a toolbar with various icons for file operations like new, open, save, and run.

The left notebook contains the following content:

- A cell titled "# Equation" containing the LaTeX code $\$ \$ c = \sqrt{a^2+b^2} \$ \$$.
- A cell titled "## Python code" containing the following code:

```
In [2]: import math  
a = 3  
b = 4  
  
In [3]: c = math.sqrt ((a*a) + (b*b))  
  
In [4]: print(c)  
5.0  
  
In [ ]:
```

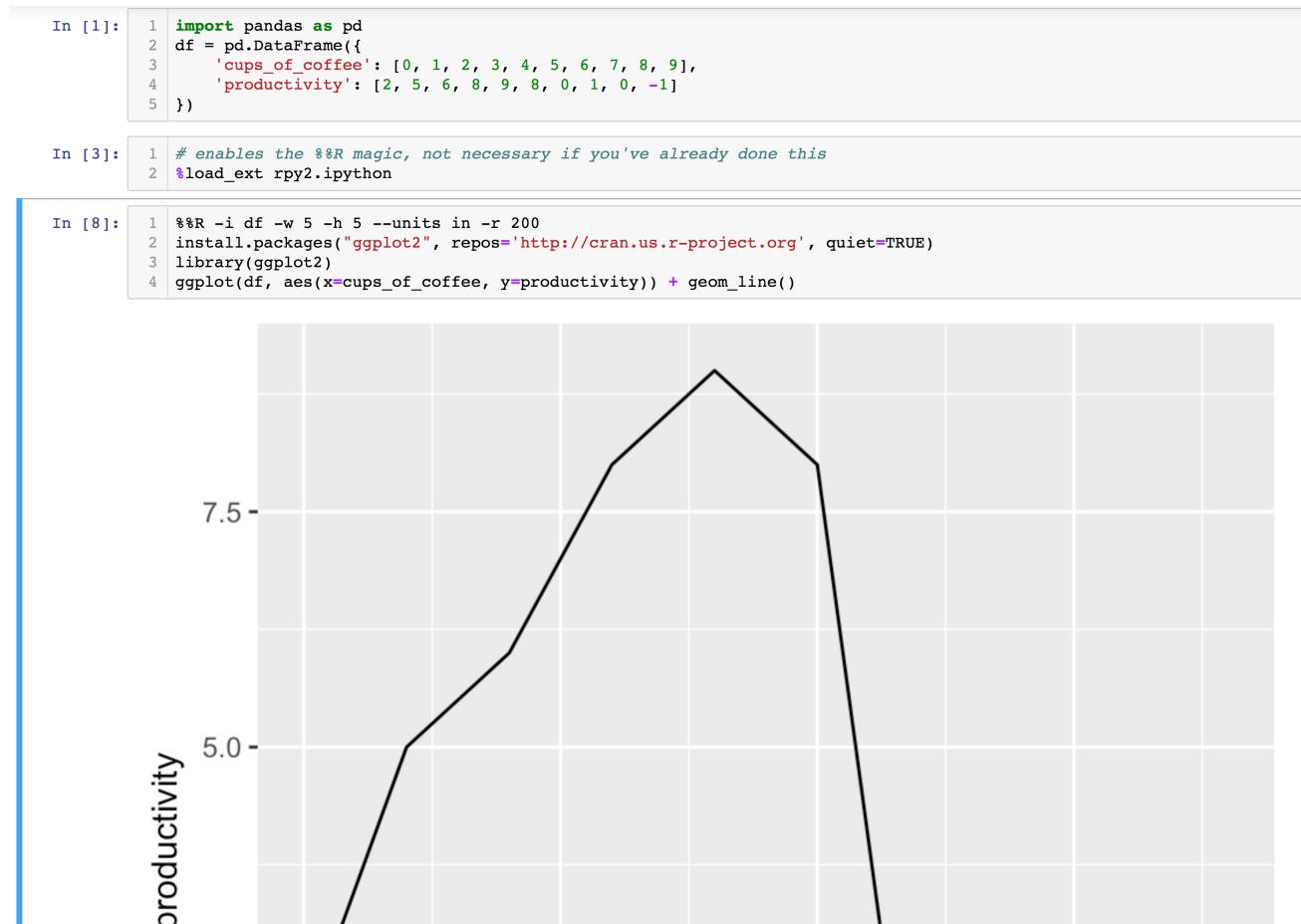
The right notebook contains the following content:

- A cell titled "Equation" displaying the rendered equation $c = \sqrt{a^2 + b^2}$.
- A cell titled "Python code" containing the following code:

```
In [5]: import math  
a = 3  
b = 4  
  
In [6]: c = math.sqrt ((a*a) + (b*b))  
  
In [7]: print(c)  
5.0  
  
In [ ]:
```

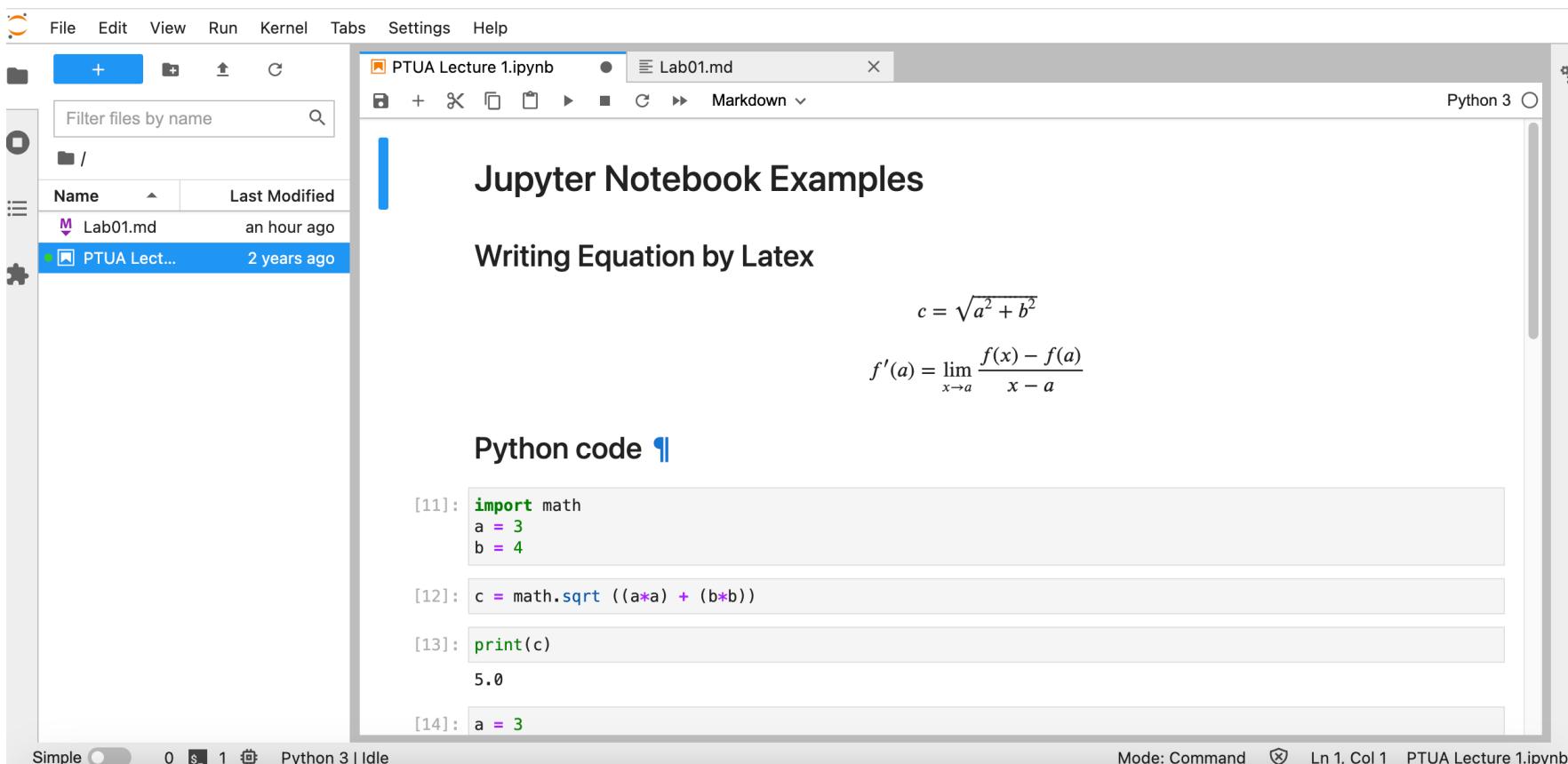
Jupyter Notebook

- You can even have Python and R in the same Notebook and use it together!



Jupyter Lab

- Web-based interactive development environment for notebooks, code, and data



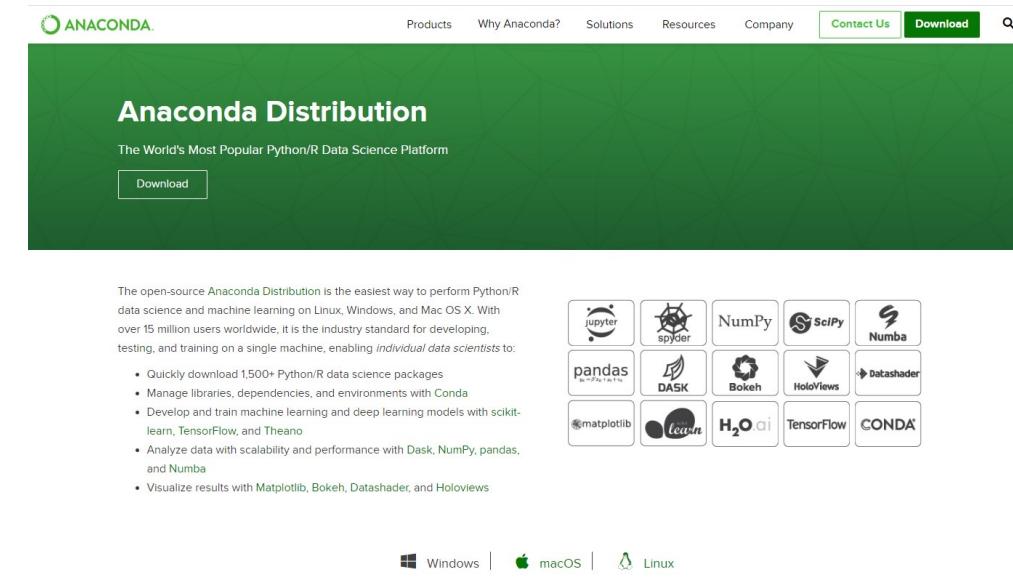
Desktop Jupyter Lab/Notebook - Anaconda

Anaconda Distribution

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download [1,500+ Python/R data science packages](#)
- Manage libraries, dependencies, and environments with [Conda](#)
- Develop and train machine learning and deep learning models with [scikit-learn](#), [TensorFlow](#), and [Theano](#)
- Analyze data with scalability and performance with [Dask](#), [NumPy](#), [pandas](#), and [Numba](#)
- Visualize results with [Matplotlib](#), [Bokeh](#), [Datashader](#), and [Holoviews](#)

It includes [JupyterLab](#), [Jupyter Notebook](#), [Ipython Console](#), [Spyder](#), etc..



Cloud Jupyter Notebook – Google Colab

Google Colab

- Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud (**no worry about installing many python packages!**).
- With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser (**no worry about your old laptops or buy a new GPU for your desktop!**).
- **Azure Notebooks** provide similar components of the cloud platform.

The screenshot shows a Google Colab notebook titled "intro_to_pandas.ipynb". The notebook interface includes a menu bar with File, Edit, View, Insert, Runtime, Tools, and Help. Below the menu is a toolbar with "+ Code" and "+ Text" buttons, and a "Copy to Drive" button. The main content area displays a section titled "Basic Concepts" with the following text:
The following line imports the *pandas* API and prints the API version:

```
[ ] from __future__ import print_function  
  
import pandas as pd  
pd.__version__
```

The primary data structures in *pandas* are implemented as two classes:

- `DataFrame`, which you can imagine as a relational data table, with rows and named columns.
- `Series`, which is a single column. A `DataFrame` contains one or more `Series` and a name for each `Series`.

The data frame is a commonly used abstraction for data manipulation. Similar implementations exist in [Spark](#) and [R](#). One way to create a `Series` is to construct a `Series` object. For example:

```
[ ] pd.Series(['San Francisco', 'San Jose', 'Sacramento'])
```

`DataFrame` objects can be created by passing a `dict` mapping string column names to their respective `Series`. If the `Series` don't match in length, missing values are filled with special `NA/NaN` values. Example:

```
[ ] city_names = pd.Series(['San Francisco', 'San Jose', 'Sacramento'])  
population = pd.Series([852469, 1015785, 485199])  
  
pd.DataFrame({ 'City name': city_names, 'Population': population })
```

Summary

Recap:

- Open-source concept
- Using Python:
 - Python terminal (write code directly in the console)
 - Code/text editor (you will have a file named XXX.py, then run it in the console)
 - Interactive Python (advanced Python with more supportive components)
 - IDE (terminal, code/text editor, debugger, and output windows together)
 - Jupyter Lab/Notebook (a web-browser based application + notebook documents)
 - Anaconda/Google Colab (Desktop or Cloud based Jupyter Notebook)

Next week:

Unix Shell, Git, and GitHub