Kikodo

Getting started with Python

What is Python?

Simply a language

Python is a language.
Existed for 15+ years.
Python 3.10 now standard.

There are many ways to use Python.

Jupyter is a tool that allows you to create **interactive documents** that **contain code, text, equations, diagrams, and other kinds of media**, in your web browser. It is a key part of the data science workflow at major companies, as it makes work more shareable, accessible, and **beginner friendly**.

Binder is a tool that allows you to run Python in the cloud, share your Jupyter notebooks with others and ensure that they can run your code and reproduce your results flawlessly.

Google Colab is a similar tool.

To use Python there are three main possibilities.

- “Directly” writing Python scripts in a .py file, then running them on the Terminal.

  This requires more expert knowledge than other methods, and is also harder to debug.

  Suited to larger projects, with many collaborators (using a system like Git). Not well suited to beginners, or agile.

- Using an IDE - we’ll see more on these shortly.

- Using Jupyter - we’ll see more shortly.

Python is the language itself. It can be installed on Windows, Linux and macOS. However, it can be trickier to install on Windows. It comes installed on macOS (can be an old version 2.7), and on most Linux distributions.

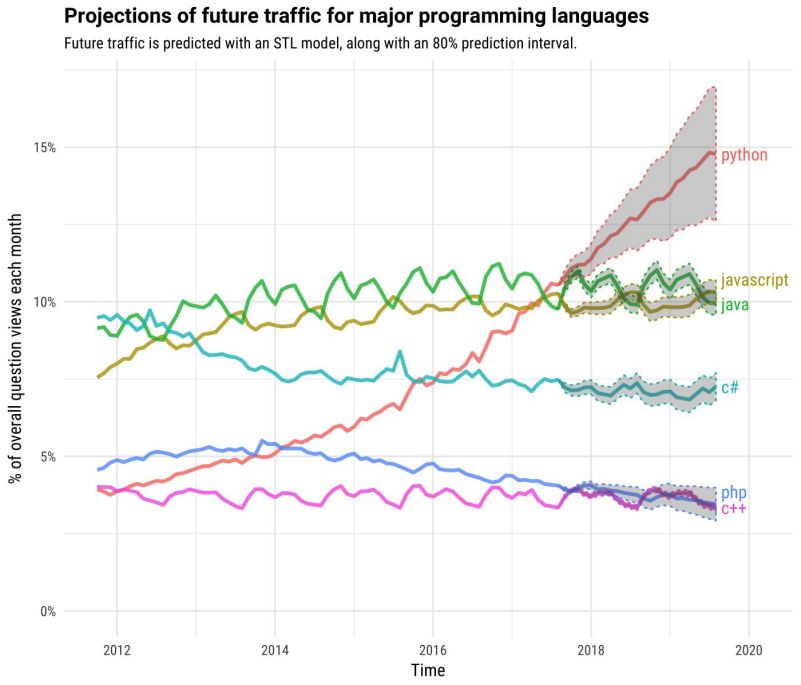
Anaconda is a cross platform package which can be installed on **Windows, Linux and macOS** in a **similar way** (cross compatible).

Provides a simple GUI (Graphical User Interface) to work with Python tools.

Jupyter comes installed with Anaconda.

What is Python used for

And who is using it?



40% of all code written today is in Python. 90% of "back-end" code.

Python has **major advantages**:

- Relatively easy for beginners to start/learn
- Intuitive syntax
- Huge community
  - Fastest growing
  - Most Stack Exchange questions
  - Open-source culture

- Huge battery of "modules" and other tools
  - Easy to install and import
  - Well-maintained
  - Security validated on pypi.org

- Python is not the fastest to run (cf. C++/Rust/Julia), but it is the fastest to **learn** and **write.**

Main industries

Main industries using Python:

- Banking
- Supply chain (FMCG + direct logistics, shipping etc)
- Marketing
- Web development

Supply chain and **Marketing** experiencing rapid adoption.

Banking and web development well established already.

Python
fastest growing

2 M
new users
pa

Python
#1 language

20 M
users

Main topics

- **Big-data + Data Science**
  - This is essentially Business Intelligence “on steroids”
  - When time-series forecasts are needed
  - When data too large for Excel
  - When data modeling is needed
  - When large batch processing is needed
  - When advanced ETL (extract-transport-load) “pipelines” are needed
  - Visualization and dashboards

- **Interoperability**
  - Interface with **Excel**
    - MS choosing to replace VBA with Python
    - Massive new user base pending
  - Interface with **SQL** and other **databases**
    - Easiest of all languages for these use cases
    - Minimize “cost of change”

Limitations of Excel

- **Scale**
  - Max data size not large enough for today
  - Slow
  - Modeling limited

- **Cooperation**
  - Limited interoperability
    - “Vendor lock-in strategy”

  - Hard to Interface with **SQL** and other **databases**
  - Macro functionality limited
    - Currently evolving to include Python

Staying safe

Local, cloud, Jupyter...

Code running “on local” cannot be accessed by anyone but you.

This is the only truly secure set up for private data.

Jupyter can run on local, on the company’s internal cloud, or on a public cloud (like Binder or Colab). G-Cloud/MS-Azure do now offer “secure cloud” options with strict data localization also.

Only the “local” or internal cloud options are truly secure. An internal cloud is the best way for teams to securely share Python projects. Always check where Jupyter is really running.

Any code launched on Binder is sent via a public connection to binder.org, and should never include secure data or results of any kind.

Binder is great for making projects public and for Python training days.

Python Careers

Entry roles and requirements

Data Engineering

This role is about grabbing data from many sources, storing it, processing it, and making it available to the people who need it.

$110,000 to $155,000 depending on skills

Data Science

This is about the analysis of data, creating reports for decision making, creating and testing machine learning models.

$100,000 to $190,000 depending on skills and role

DevOps

This is about deploying and maintaining servers.

Using Google Cloud, AWS, database systems and more to keep data-driven services live.

$90,000 to $130,000 depending on skills and role

Employers are looking for skills in:

Data Engineering

- Python
- Pandas, Numpy
- Data cleaning and preparation experience
- Database systems (SQL, PostgreSQL,...)

Data Science

- Python
- Jupyter + dashboards + report creation
- Plotly, Matplotlib, Seaborn, Bokeh,...
- Pandas, Numpy, SciKit-learn
- Machine learning
- Advanced statistics

DevOps

- Python
- Docker
- Google Cloud + AWS + Cloud
- Database deployment (SQL, PostgreSQL, MongoDB...)

Things you can do to get started

Data Engineering

- Learn Python, pandas, Numpy
- **START PROJECTS** and get them up on Github
- Contribute to Open Source projects

Data Science

- Learn Python, pandas, Numpy
- Master Jupyter
- **START PROJECTS** and get them up on Github
- Contribute to Open Source Projects
- Get certificates in statistics, machine learning
- Consider a Ph.D. program

Dev ops

- Learn Python, Bash, advanced Unix terminal
- Get certified in Docker
- Get certified in G-Cloud + AWS + Cloud
- Database deployment (SQL, Postgres, Mongo...)

Anaconda

An easy wrapper for Python and related tools

ANACONDA®

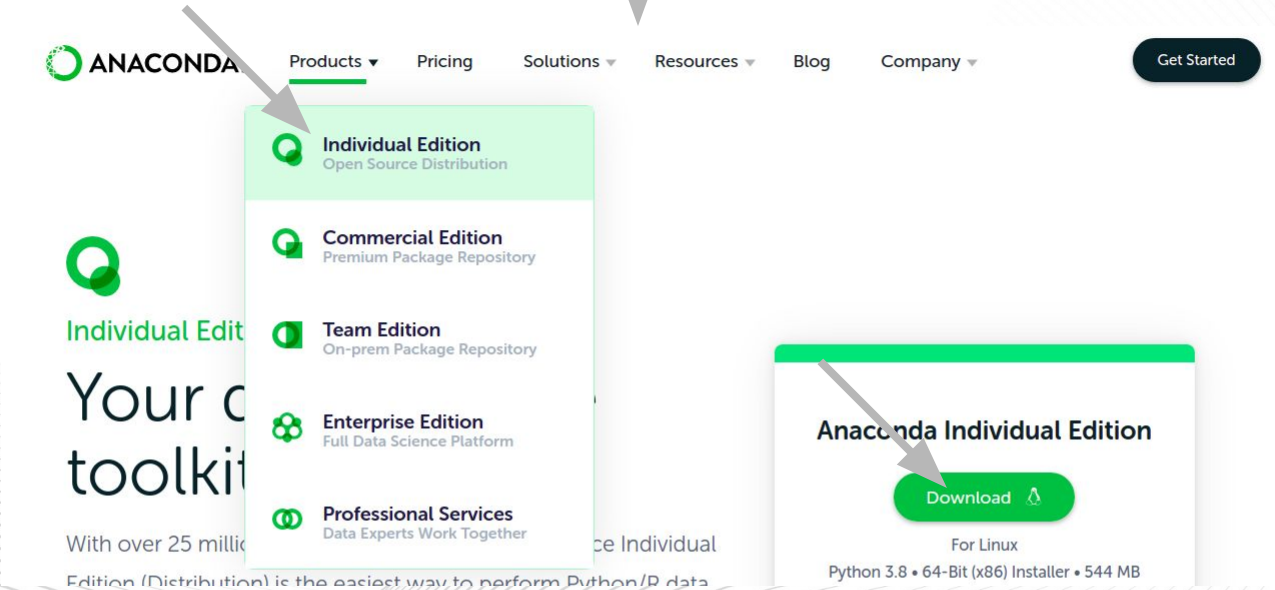
Anaconda is a cross platform package which can be installed on Windows, Linux and macOS in a similar way.

Provides a simple GUI (Graphical User Interface) to work with Python tools.

jupyter

Jupyter and many other Python tools come installed with Anaconda.

https://www.anaconda.com/



macOS/Linux vs Windows

macOS
After running the install, click and drag the Anaconda application into the “Applications” folder.

Other *NIX (Linux - Ubuntu, Mint, Debian...)

Set the bash script as executable:

```
chmod +x ./Anaconda3-2021.05-Linux-x86_64
```

You may find some missing requirements:

https://docs.anaconda.com/anaconda/install/linux/

Windows

After the download, run the installer .exe file.

Follow the GUI instructions, and Anaconda should appear in your Start Bar/Desktop, depending on the options you selected as “Anaconda Navigator”.

Spyder comes free with Anaconda.

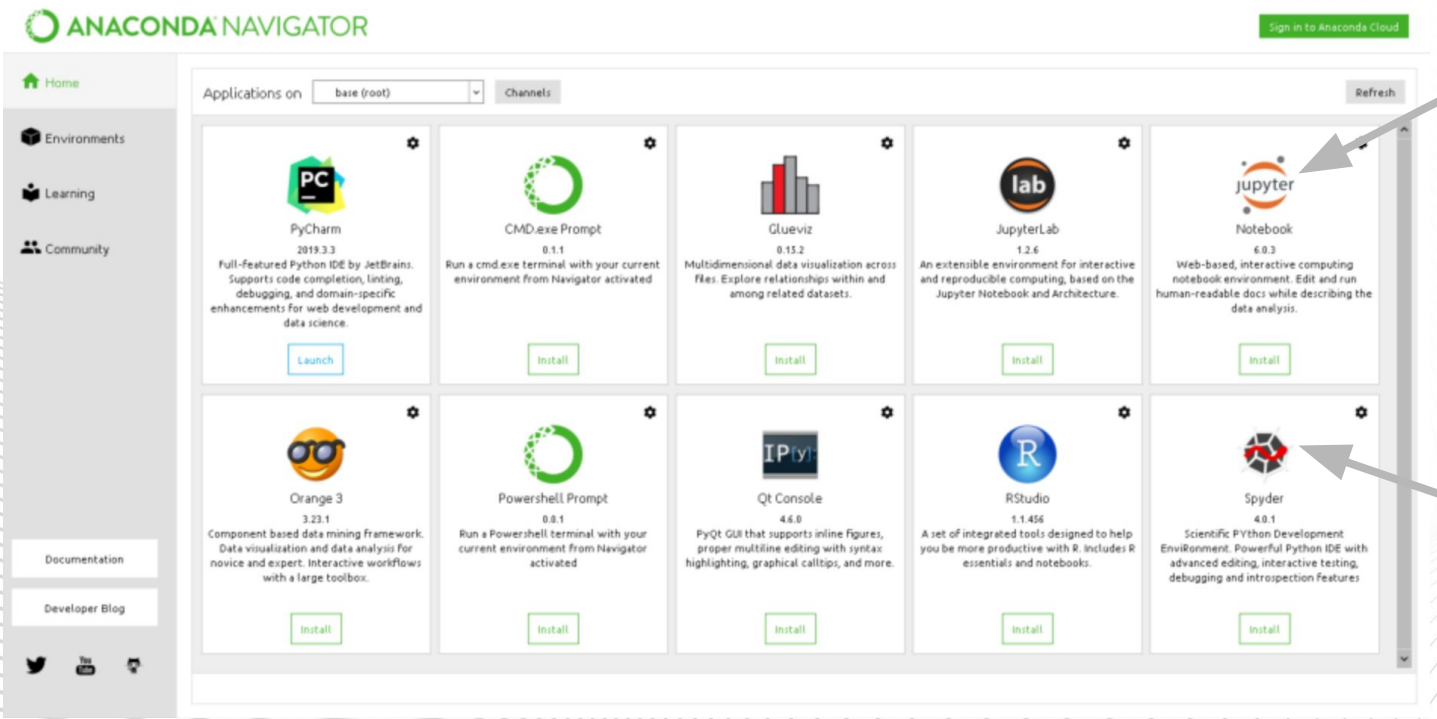
It's well-suited to beginners in Python, and comes with a neat, Matlab-style way to inspect variables as your code runs, which can help learning and understanding. Spyder is for creating .py files, and other code, but not notebooks.

Jupyter can also be used as a "pseudo IDE", but it has only very basic features. Jupyter also comes free with Anaconda.

PyCharm is another popular free choice for beginners which ships with Anaconda.

PyCharm is for creating Python projects, .py files and other code. It's not for editing notebooks.

Running Anaconda Navigator



Jupyter is included. This is where you can create notebooks.

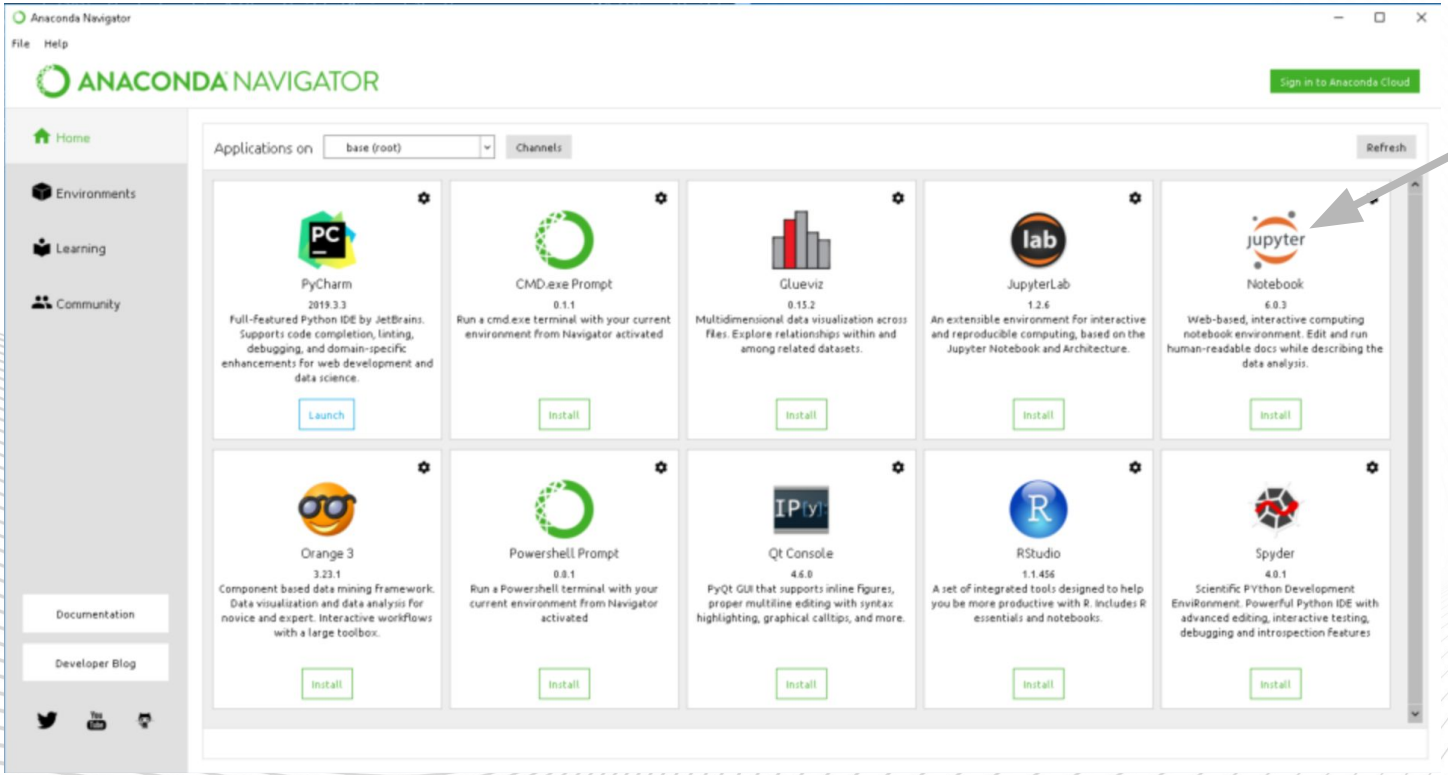
These files end in *.ipynb extension.

Spyder is a popular choice of first “IDE” also to create Python programs.

Opening Jupyter

Demo - open, create and save notebooks

Running Jupyter



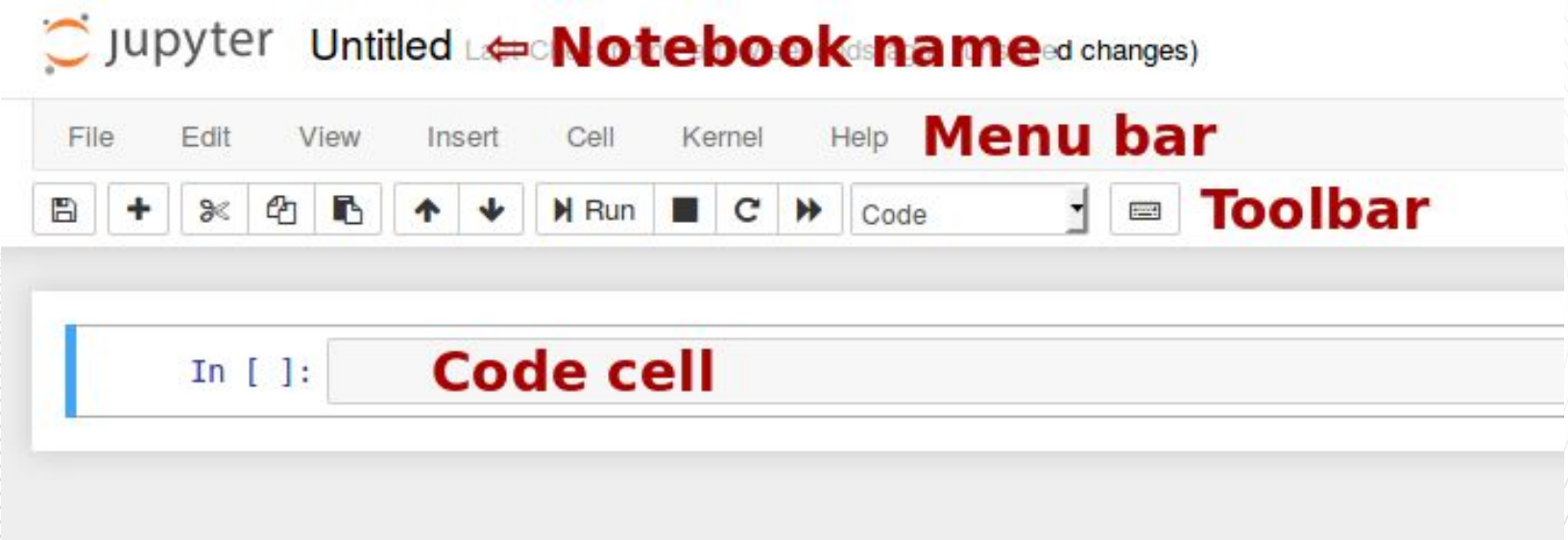
Run Jupyter now, and we will create our first notebook!

Running Jupyter

Create a new notebook here.

Select Python 3.

Running Jupyter

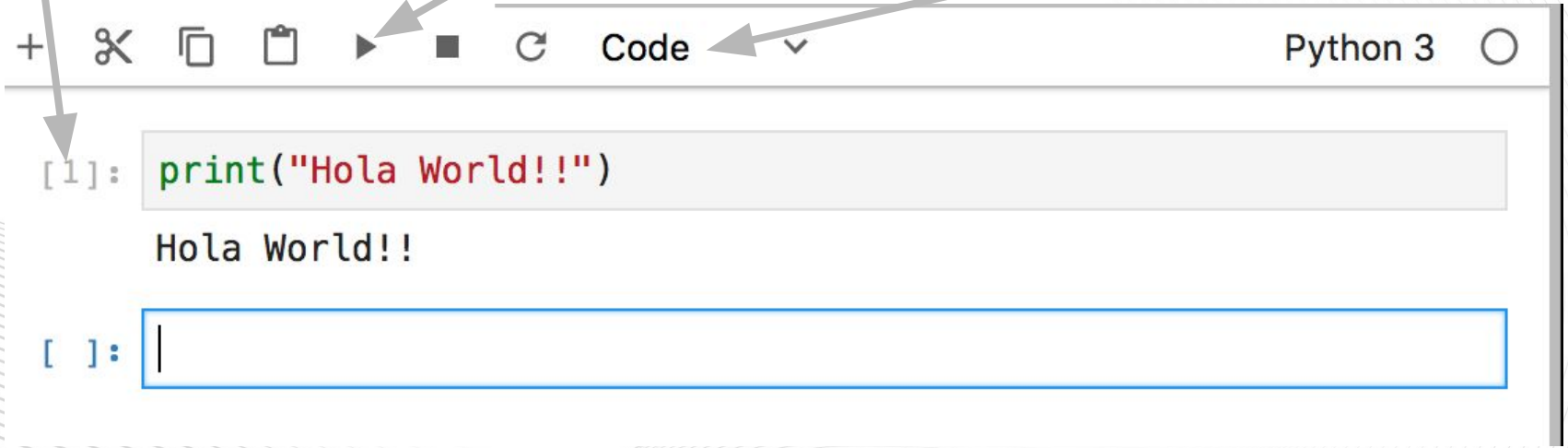


Running Jupyter

Change cell type to Markdown to add notes

Shows run order

Run selected cell (also Shift-Enter)



Intro to the terminal

Install using pip or conda

The terminal

Using the terminal is an important part of learning to be a software developer.

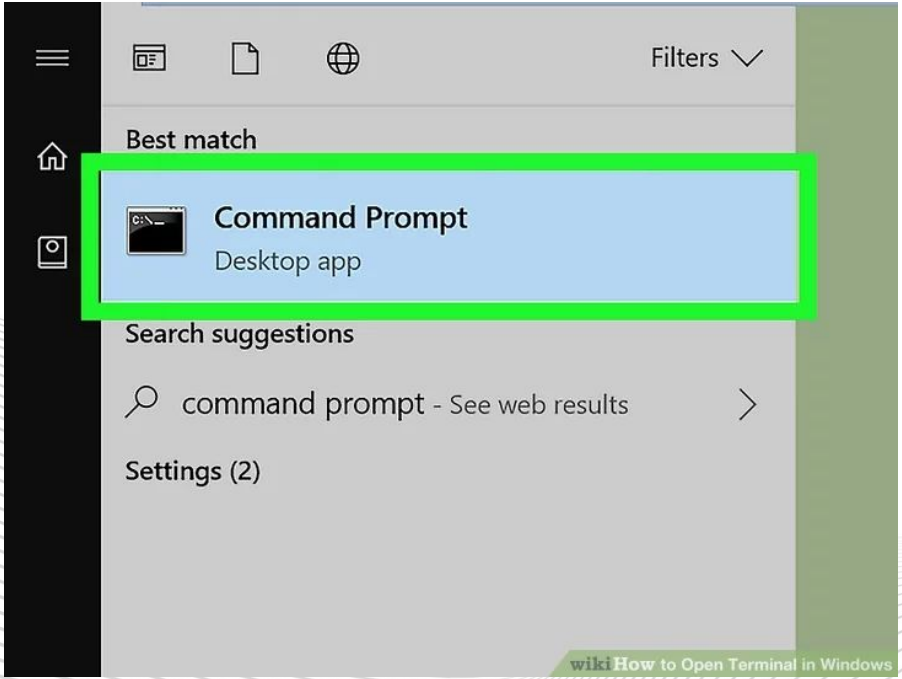
This is slightly different on Mac, Linux and Windows.

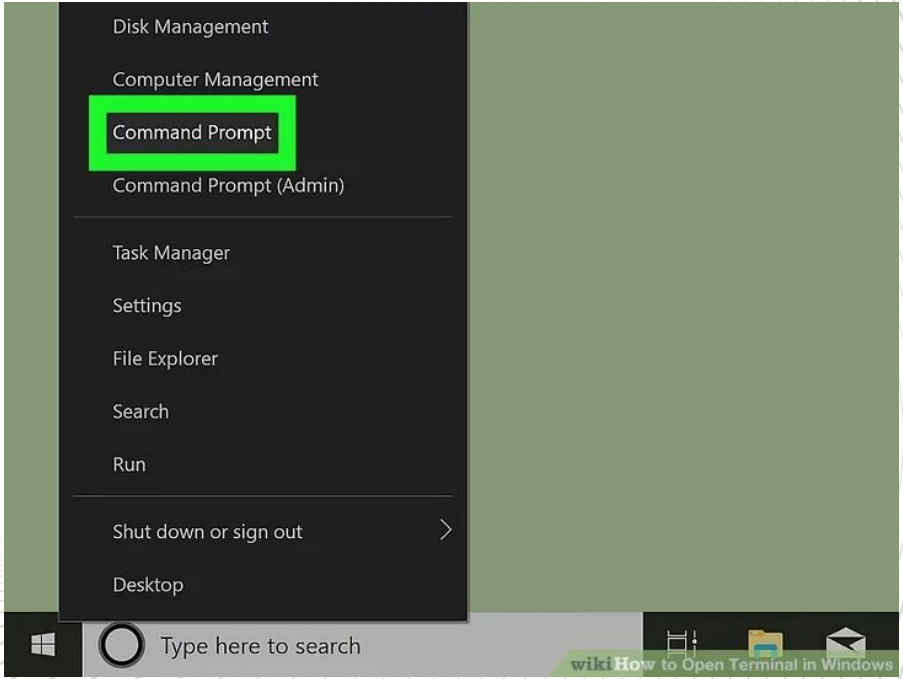
However, there is enough in common not to really affect how you use git.

In windows the terminal is also called the “command prompt”.

Whichever system you most use, it’s good to know them all so you are ready for any tools at work in the future.

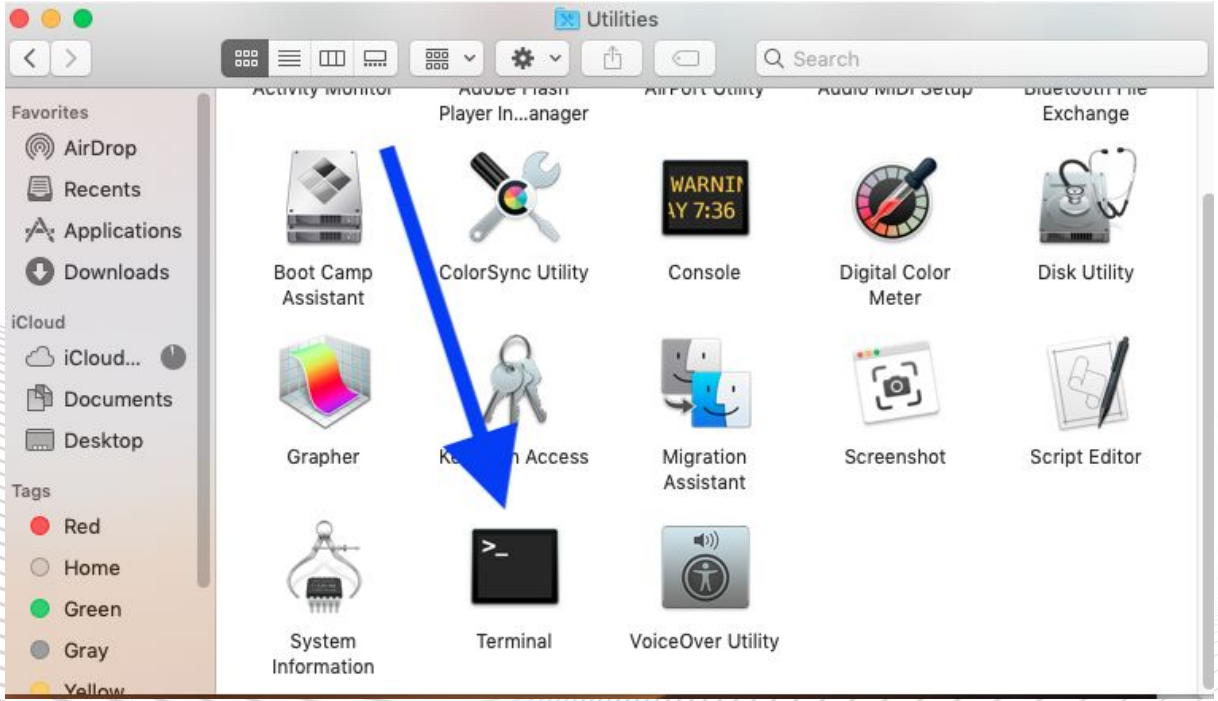
Opening the terminal





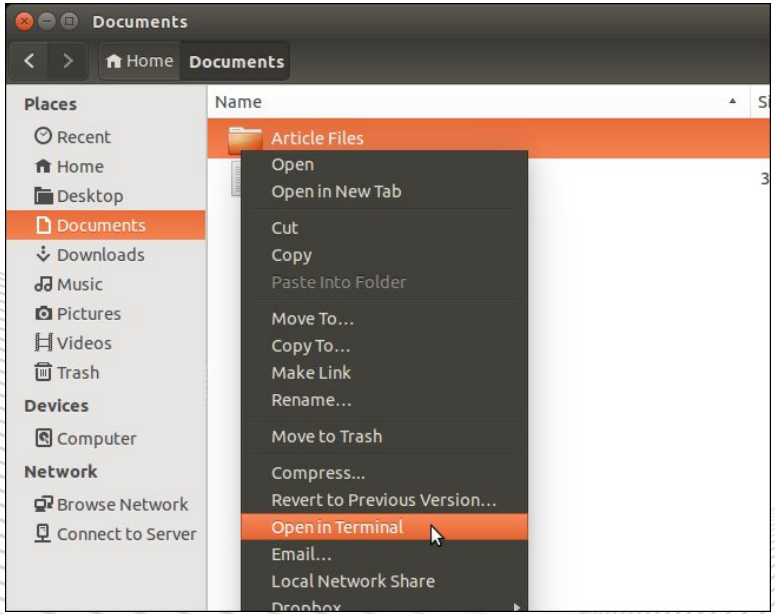
https://www.wikihow.com/Open-Terminal-in-Windows

Opening the terminal



You can also search the word “terminal” in searchlight.

Opening the terminal



Most linux distributions come with “open terminal here” type of option if you right click a folder.

Learning Libraries

Pandas, numpy, matplotlib...

Numpy + Pandas

This is a tool, born in the hedge-fund industry. It's used for analysing data, simple modeling, and is especially useful for simple modeling and visualization.

Matplotlib

This is a tool for visualizing data rapidly in python and jupyter.

Scikit-learn

This is a tool for making machine learning models ranging for simple linear regression, to advanced neural networks and nlp algorithms.

Numpy + Pandas

Open the terminal / command prompt

```
conda install numpy
conda install pandas
```

Matplotlib

Open the terminal / command prompt

```
conda install -c conda-forge matplotlib
```

Scikit-learn

Open the terminal / command prompt

```
conda install -c conda-forge scikit-learn
```

Future topics

Git, pip, venv....

What we offer

- Rapid training in Python, Git, Jupyter, Python libraries
- Live Python help via discord when you are stuck
- Marked mini projects in 2X monthly small group tutoring
- Job application mentoring and Interview preparation

We are currently offering this for half price: $20 pm

Just register on www.kikodo.io and use the code:

SkillsHalfPrice

A tool to keep track of the versions of your code

A tool for installing Python modules and tracking their versions

A tool for isolating your Python projects, especially when different projects need different versions of libs/modules

These will be the next webinars...