

INTRODUCTION TO PYTHON

19.03.2019



PYTHON

Python is an interpreted **high-level language** for **general-purpose programming**.

The language has a design philosophy that emphasize **code readability**.

Python is **open-source** and has a community-based development model.



WHAT IS A HIGH-LEVEL LANGUAGE?



In computer science, a **high-level programming language** is a programming language with strong abstraction from the details of the computer.



Level of abstraction



- Machine language
- Assembly
- C / C++
- Fortran
- Pascal
- Java
- Python

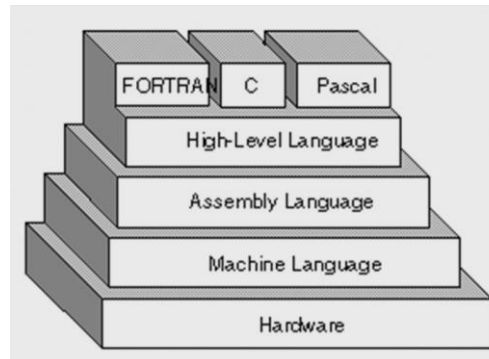


High Level



Low Level Language

- Superb performance
- Direct memory management
- Hard to learn



High Level Language

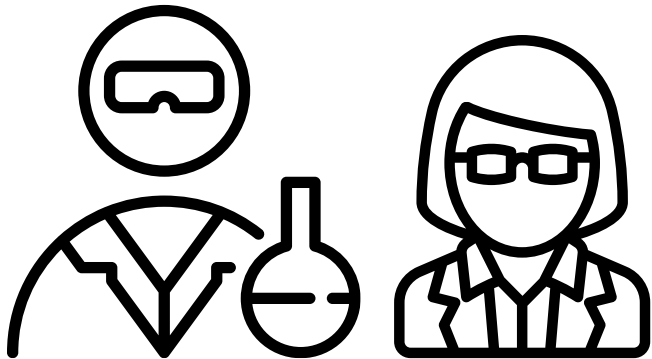
- High level of abstraction (Easy to read)
- Interpreted
- Object oriented and functional
- Large communities of user
- Slower in terms of performance



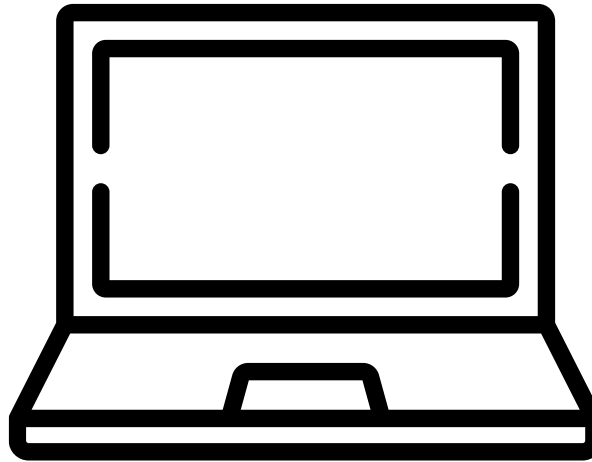
WHAT IS GENERAL PURPOSE?

A **general-purpose programming language** is a **programming language** designed to be used for writing software in the widest variety of application domains.

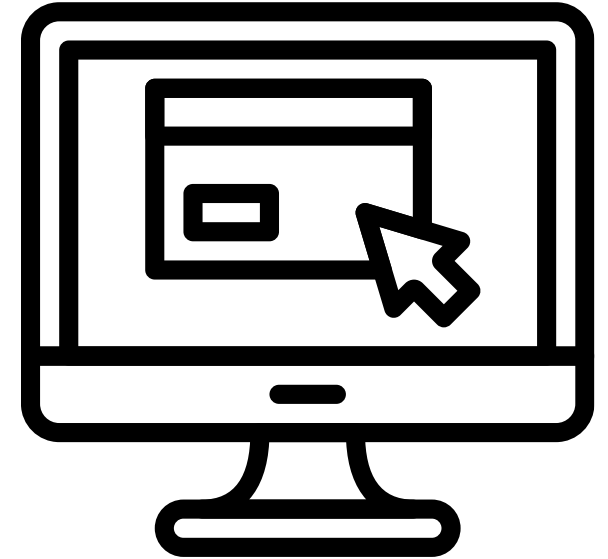
Scientific & numeric tasks



Desktop applications



Web applications



WHAT IS OPEN SOURCE SOFTWARE?



Open-source software is a type of software which source code is released under a license in which the copyright holder **grants users the rights to study, change, and distribute the software** to anyone and for any purpose.



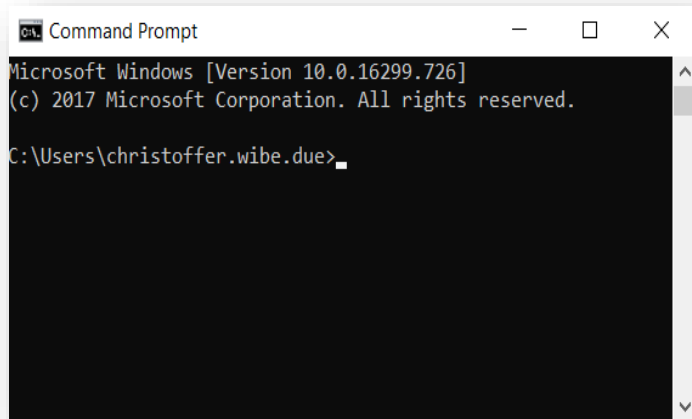
TOOLS

Python programing can take a variety of shapes depending on the tool being used.



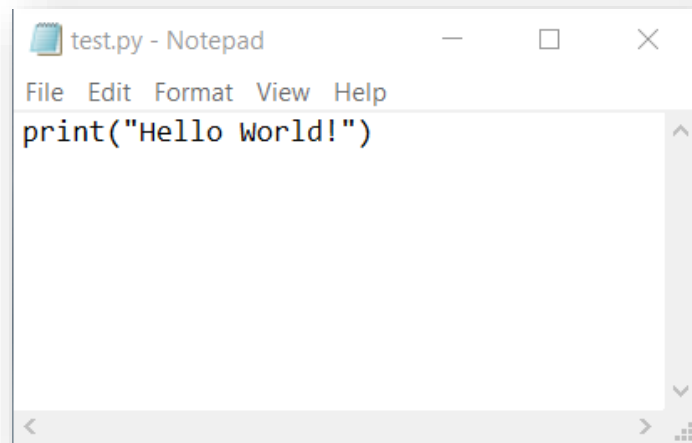
WHAT TOOLS CAN WE USE TO WRITE CODE?

CMD + Text editor



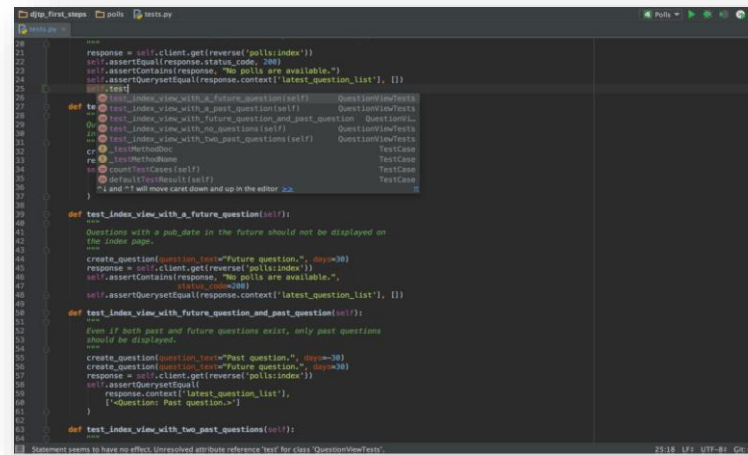
```
Microsoft Windows [Version 10.0.16299.726]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\christoffer.wibe.due>
```



```
test.py - Notepad
File Edit Format View Help
print("Hello World!")
```

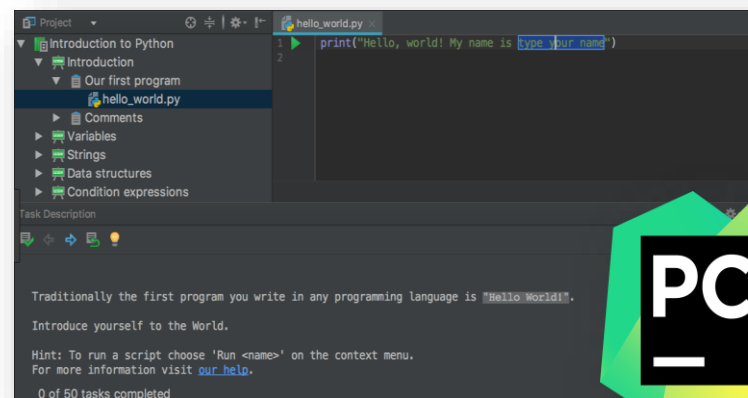
Integrated development environments (IDE)



```
def test_index_view_with_a_future_question(self):
    """
    Questions with a pub_date in the future should not be displayed on
    the index page.
    """
    create_question(question_text="Future question.", days=30)
    response = self.client.get(reverse('polls:index'))
    self.assertEqual(response.status_code, 200)
    self.assertQuerysetEqual(response.context['latest_question_list'], [])

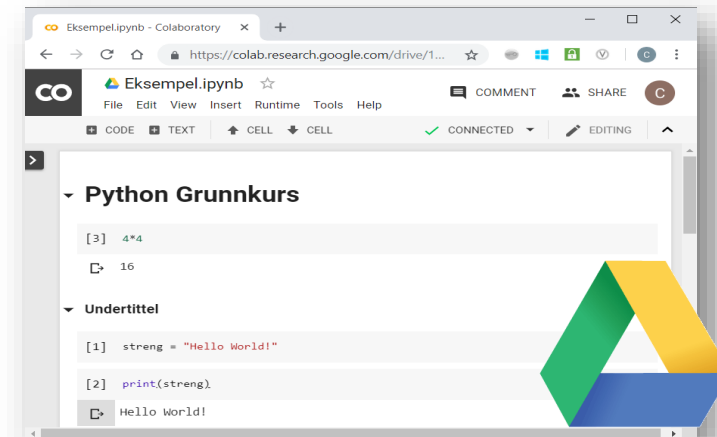
def test_index_view_with_future_question_and_past_question(self):
    """
    Even if both past and future questions exist, only past questions
    should be displayed.
    """
    create_question(question_text="Past question.", days=-30)
    create_question(question_text="Future question.", days=30)
    response = self.client.get(reverse('polls:index'))
    self.assertQuerysetEqual(
        response.context['latest_question_list'],
        ['question: Past question...']
    )

def test_index_view_with_two_past_questions(self):
    """
    Questions with a pub_date in the past should be displayed on the
    index page.
    """
    create_question(question_text="Past question 1.", days=-30)
    create_question(question_text="Past question 2.", days=-30)
    response = self.client.get(reverse('polls:index'))
    self.assertQuerysetEqual(
        response.context['latest_question_list'],
        ['question: Past question...']
    )
```



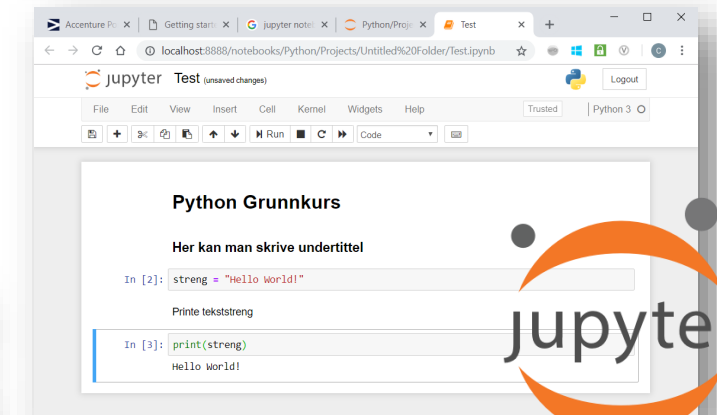
```
hello_world.py
1 print("Hello, world! My name is Type your name")
```

Notebooks



```
Eksempel.ipynb - Colaboratory
https://colab.research.google.com/drive/1...
Eksempel.ipynb
Python Grunnkurs
[3] 4*4
16
Undertittel
[1] streng = "Hello World!"
[2] print(streng)
Hello World!
```

Google Drive



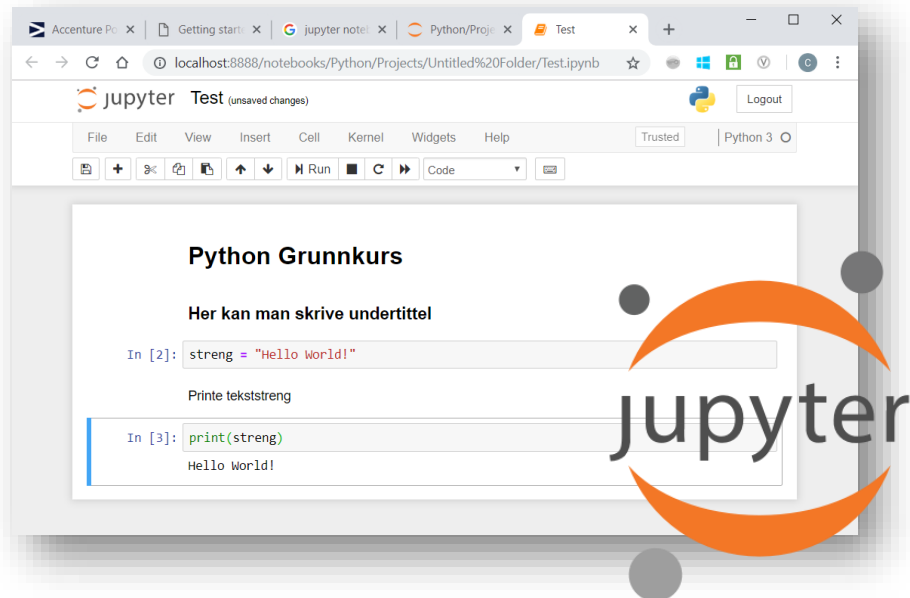
```
jupyter Test (unsaved changes)
Python Grunnkurs
Her kan man skrive undertittel
In [2]: streng = "Hello World!"
Printe tekststreng
In [3]: print(streng)
Hello World!
```


WHY NOTEBOOKS?

A **notebook** integrates code, and its corresponding output, into a **single document** that combines visualizations, narrative text, mathematical equations, and other rich media. The intuitive workflow promotes iterative and rapid development, making notebooks an increasingly popular choice.

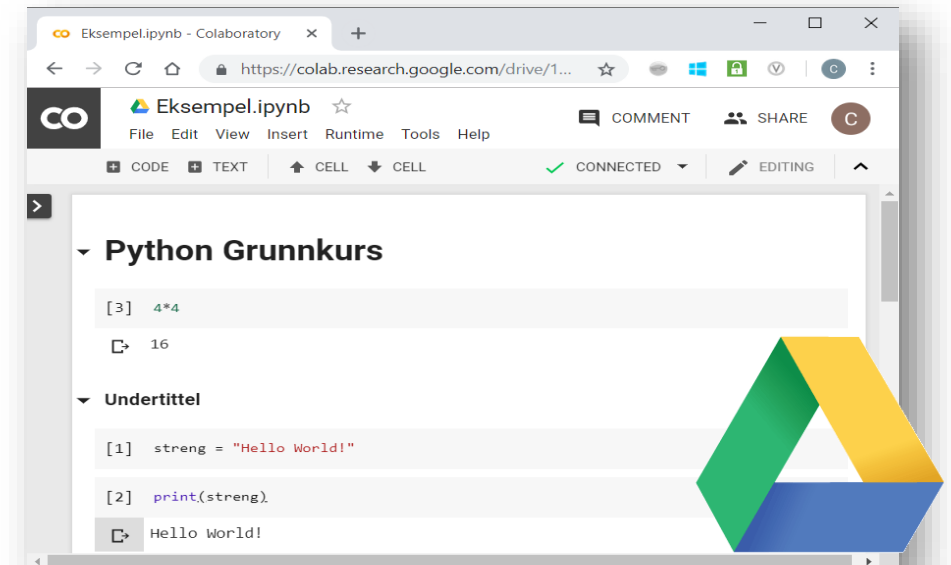
Jupyter Notebook

- Julia, Python, R compatible.
- Runs on your own laptop/computer
- Need installation of both Jupyter notebook as well as programming language.



Google Colaboratory

- Based on Jupyter Notebook, but only support for Python. (so far)
- Cloud based – no installation needed.
- Can make use of cloud GPU and TPU

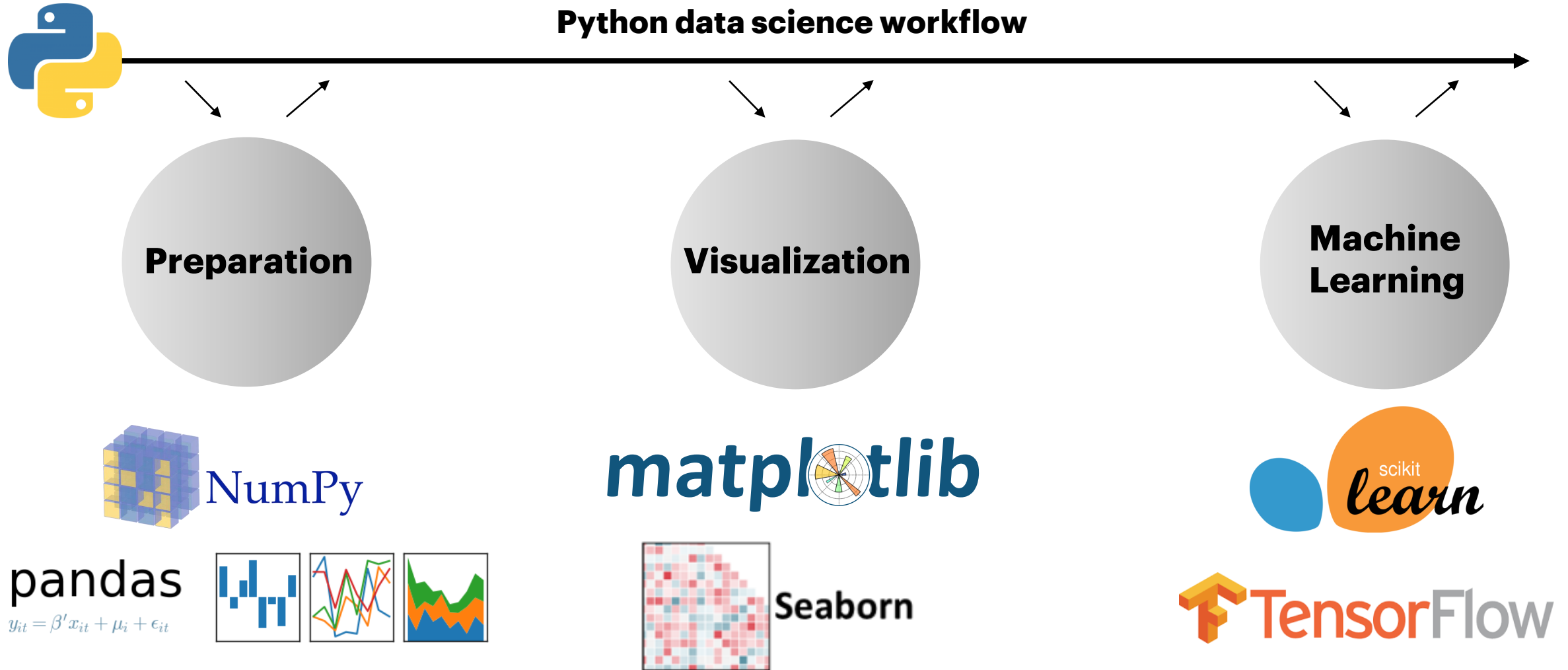


LIBRARIES

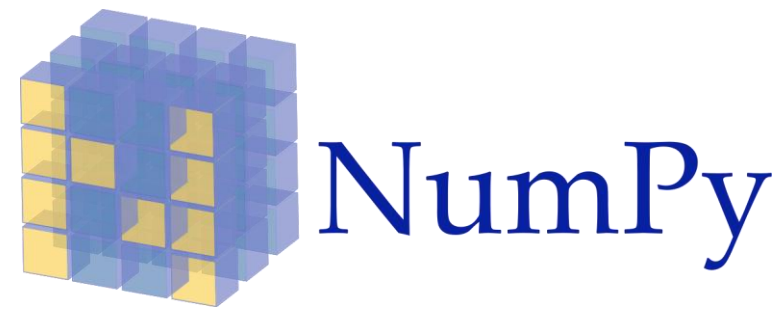


PYTHON DATA SCIENCE LIBRARIES

Python offers libraries to support all steps of the data science workflow.



NUMPY



“Good replacement for MATLAB”

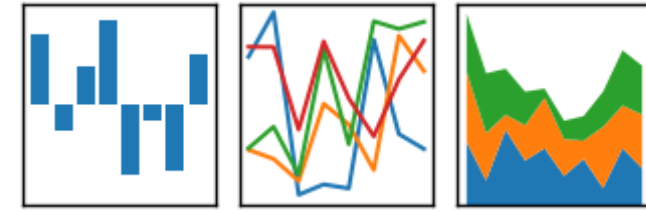


“Lightning quick calculations”

- NumPy is a linear algebra library for python.
- NumPy is the fundamental package for scientific computing with Python, where other libraries often rely on NumPy as one of the main building blocks.
- Most famous for its N-dimensional array data structure.
- The library is incredibly fast and has bindings to C libraries.

PANDAS

pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



“Python’s version of Excel”

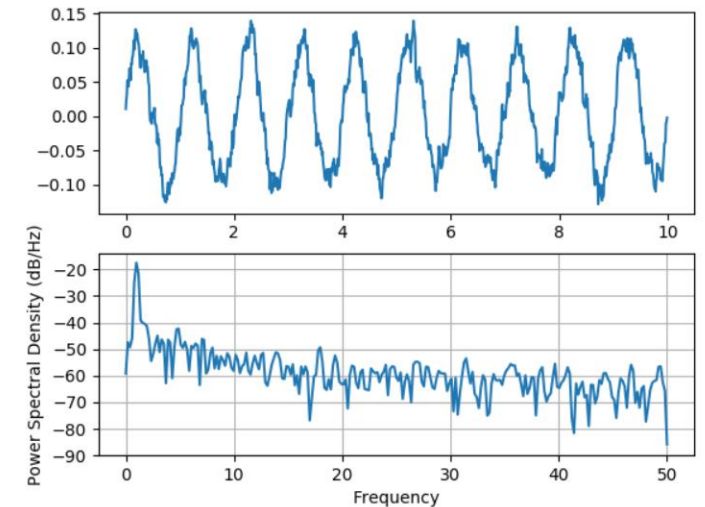
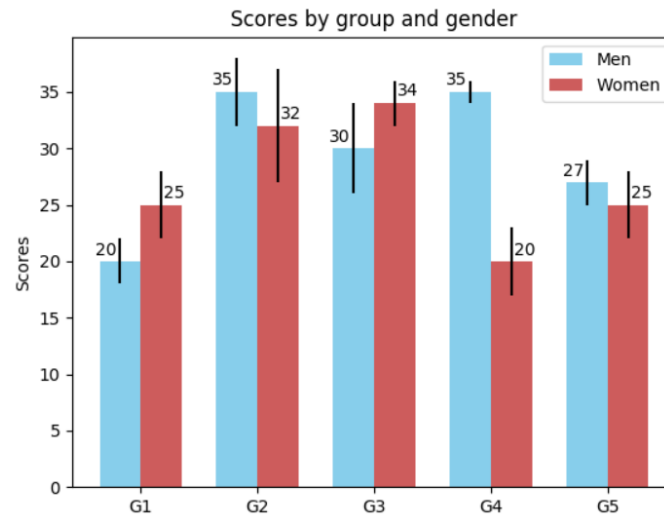
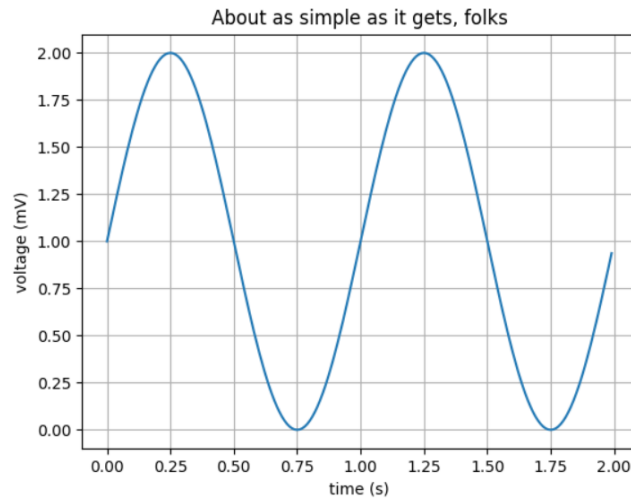
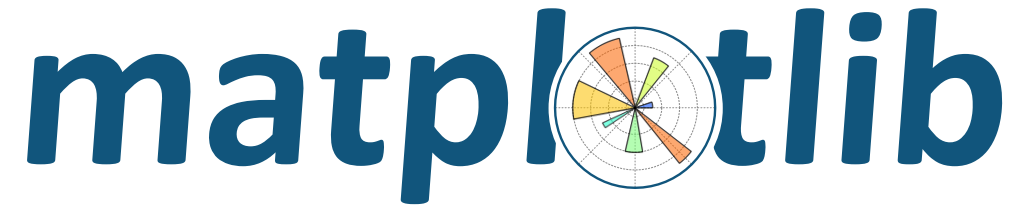


“Python’s version of R data frames”



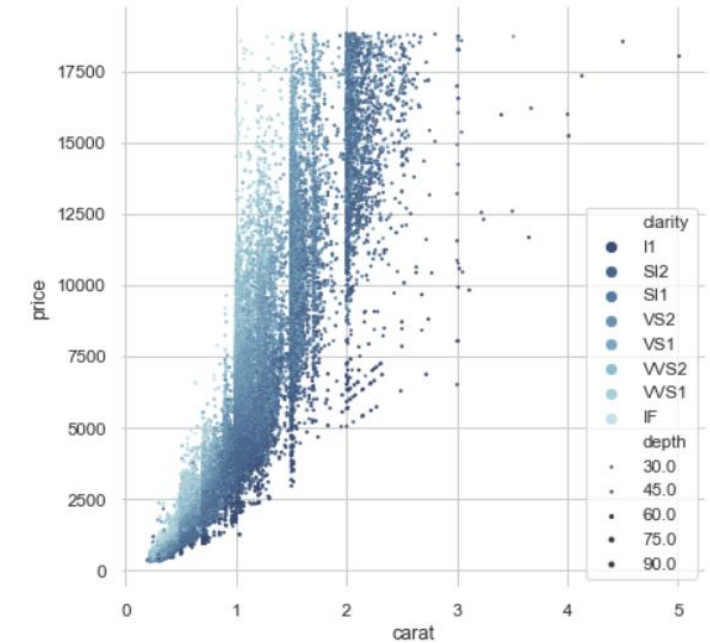
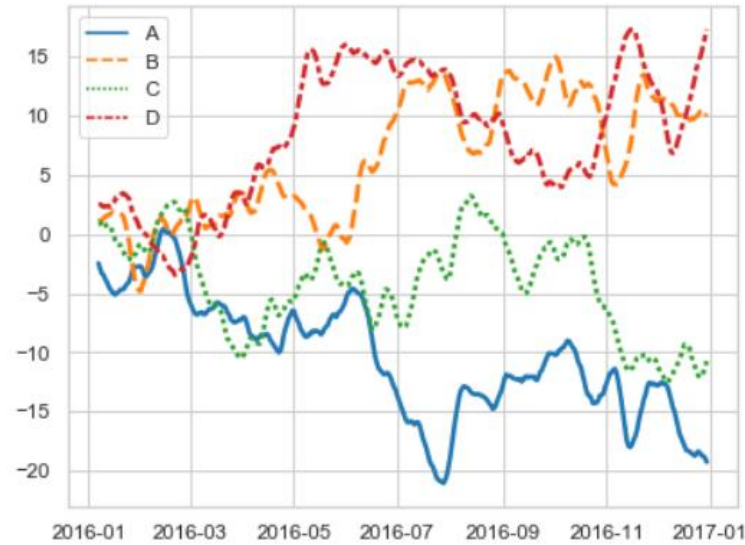
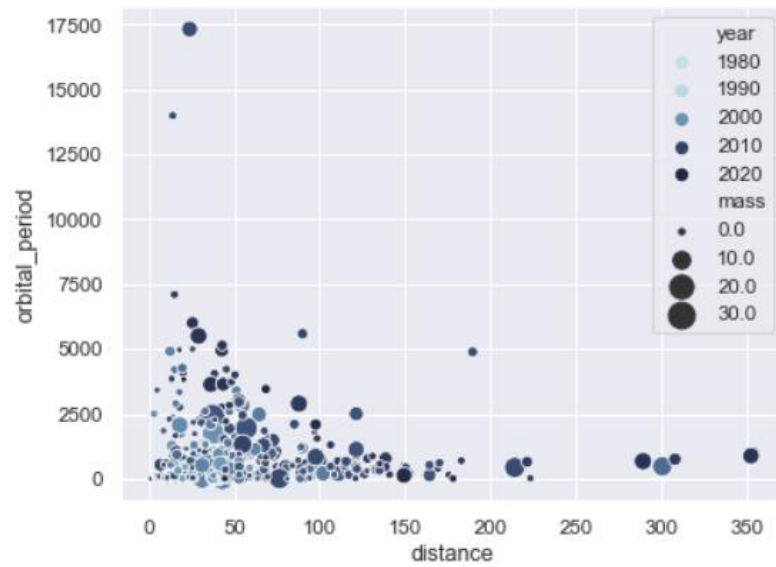
- Pandas is an open source library built on top of Numpy.
- The library allows for fast data cleaning, preparation and analysis using Python.
- Pandas also has built in visualization features.
- It can work with data from a large variety of sources

MATPLOTLIB



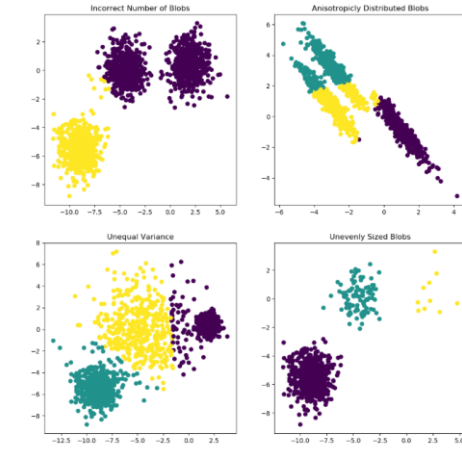
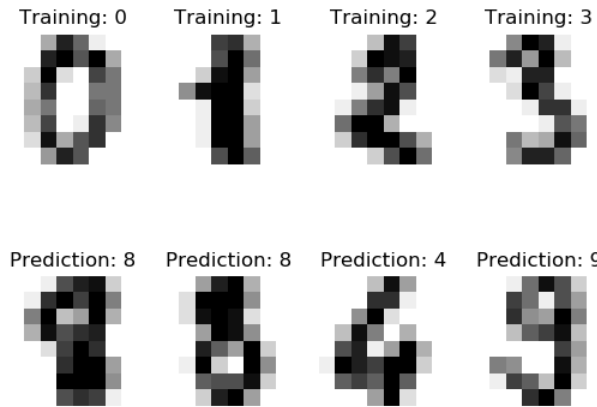
- Matplotlib is a Python plotting library which produces publication quality figures in a variety of hardcopy formats.
- Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

SEABORN



- Seaborn is a library for making statistical graphics in Python.
- Seaborn aims to make visualization a central part of exploring and understanding data
- Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets.
- An alternative to R's plotting capability with ggplot2.

SCIKIT LEARN



- SciKit Learn offers simple and efficient tools for data mining and data analysis.
- Built on NumPy, SciPy, and matplotlib.
- Contains a wide variety of machine learning models (classification, regression, clustering etc)

WHY LEARN IT?



REASONS TO LEARN PYTHON

Python is great programming language for beginners!

- It is easy to understand, with clear readable syntax and quick progression.
- It's free, with lots of educational resources and communities to help you out.
- From 2014 it was the most popular introductory programming language in higher education.

Basic programming knowledge makes you more attractive in an increasingly digital work environment

- Allow you to solve everyday problems in a more efficient way.
- In the process of learning a programming language you learn a lot about digital technologies. This knowledge will be applicable in many work situations throughout your career.

Python is the number one programming language for AI and machine learning

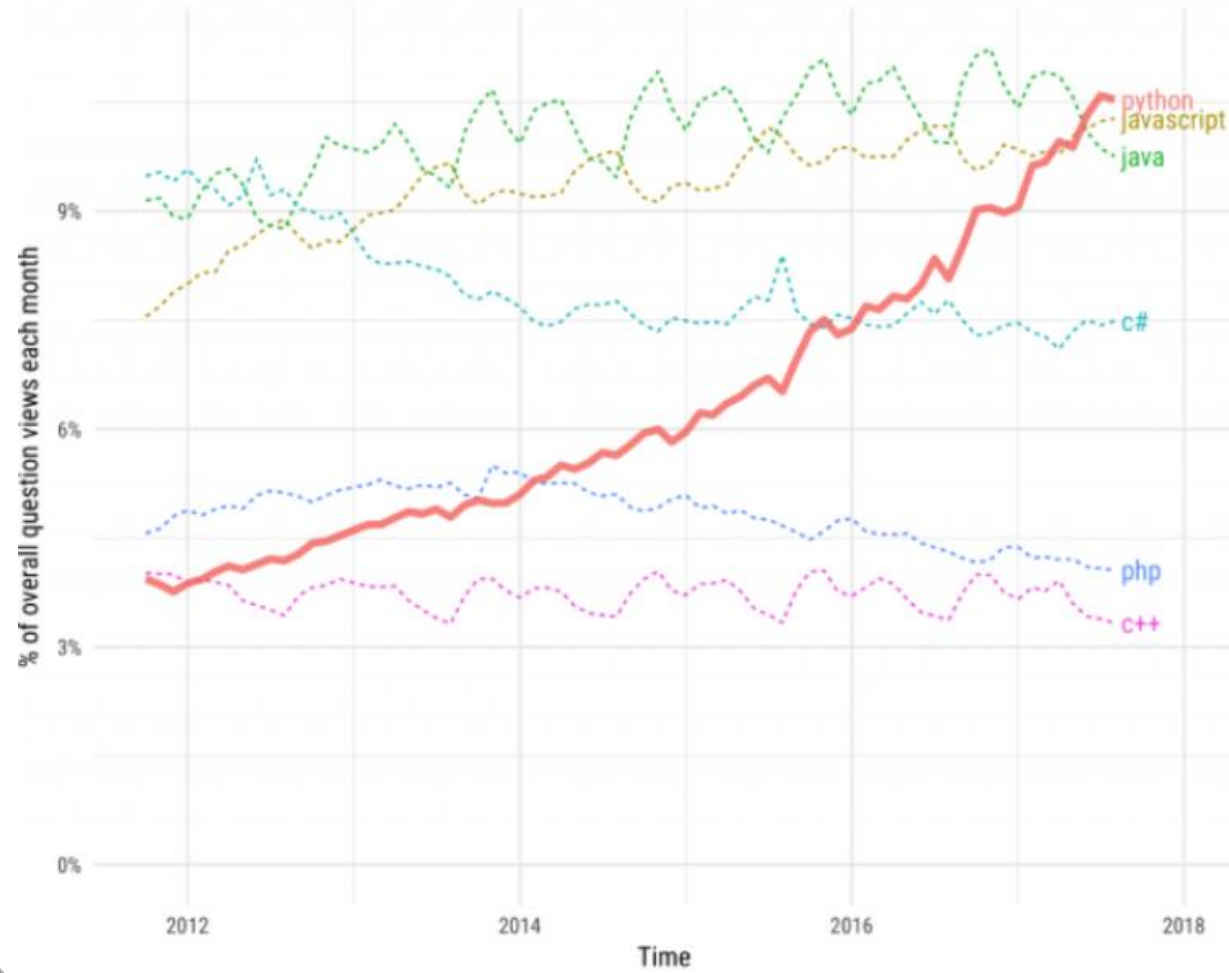
- AI and Machine Learning is all the buzz nowadays. Basic knowledge in Python allow you to use and experiment with these technologies.

EXAMPLE OF PRACTICAL PROBLEMS

- **Excel being to slow / unusable**
 - Excel sheets of more than ~20mb file size or CSV files of ~28mb often create problems with “normal laptop” specifications.
 - 1,048,576 is the maximum number of rows in excel.
- **Want faster workflow, or more opportunities**

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



LEARN MORE!

Links to webpages and courses that provide useful information.



USEFUL LINKS

Official pages

- Python main webpage
<https://www.python.org/>
- Numpy main webpage
<http://www.numpy.org/>
- Pandas main webpage
<https://pandas.pydata.org/>
- Matplotlib main webpage
<https://matplotlib.org/>
- Seaborn main webpage
<https://seaborn.pydata.org/>
- SciKit Learn
<http://scikit-learn.org/>

Good learning resources

W3 Schools
<https://www.w3schools.com/python/>

Learn Python
<https://www.learnpython.org/>

Online course material

Udemy courses:

- Complete Python Bootcamp: Go from Zero to Hero in Python 3
- Python for Data Science and Machine learning Bootcamp

Coursera:

- Python for Everybody
- Machine Learning (Stanford University via Coursera)