

# Introduction to Python

For Data Science and Scientific Computing

Pedro Bordignon

Wednesday, October 9th 2019



# Outline

- **Python:**
  - **What is it?**
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Methods
  - Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

# What is it?

- Interpreted, high-level, object-oriented with dynamic semantics programming language created in 1989 by:
- Guido Van Rossum (ex-BDFL, 1990-2018)
- Named after the *Monty Python* British comedy group



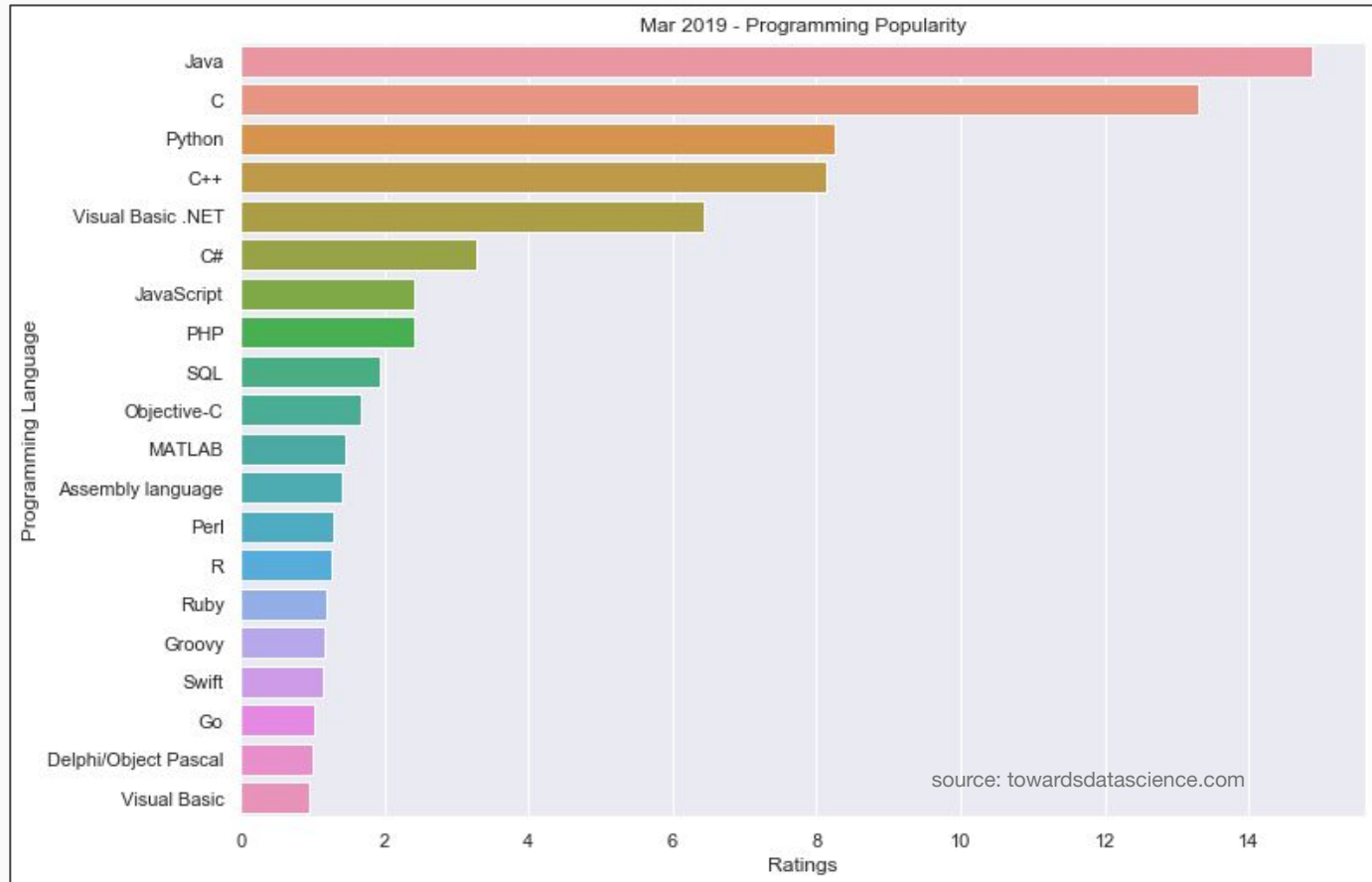
# Outline

- **Python:**
  - What is it?
  - **Why Python?**
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Methods
  - Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

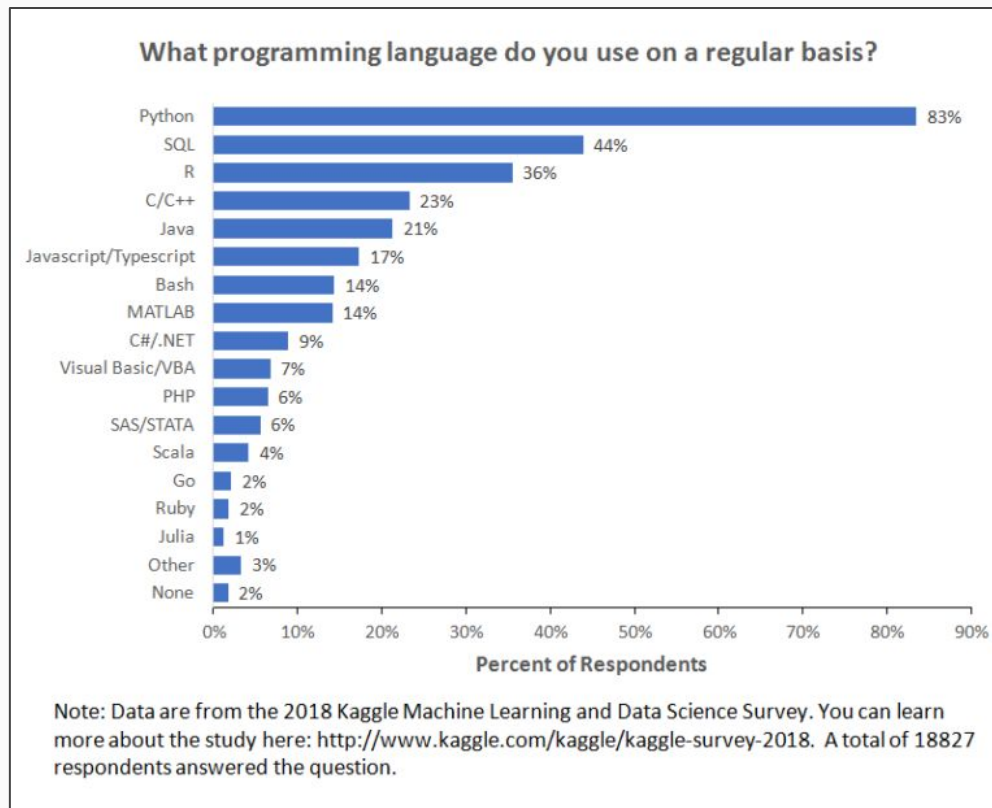
# Why Python?

- **Easy to learn**, very simple language (no garbage collection or variable declaration)
- **Open source**, with one of the biggest communities in the world (a lot of available and maintained packages)
- **Easy to read**, almost written as plain english (great for collaborations)
- **Extremely versatility**, you can analyse and visualize data, code websites or video games, do statistical work and machine learning

# Most used programming languages **Worldwide** (2019)



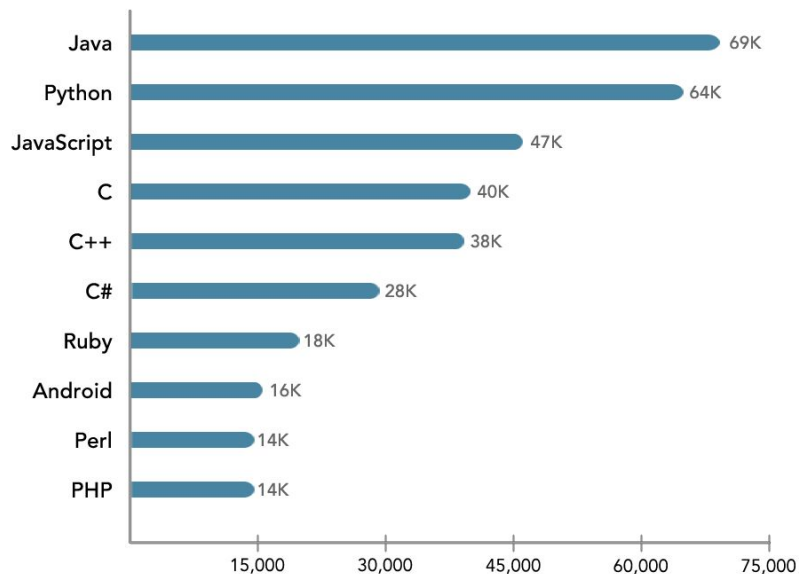
# Most used programming languages for Machine Learning and Data Science (2018)



# Most in-demand programming languages for **Jobs** (2019)

## Most in-demand programming languages of 2019

*Based on Indeed.com job postings in the USA - Feb 1, 2019*



*Image Source: CodingNomads*



# Outline

- **Python:**
  - What is it?
  - Why Python?
  - **Installation**
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Methods
  - Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

# Installing Python



Refer to [pdfs/python\\_training\\_setup\\_guide.pdf](#)

<https://www.python.org/downloads/>

<https://www.anaconda.com/distribution/>

<https://docs.conda.io/en/latest/miniconda.html>

# Outline

- **Python:**
  - What is it?
  - Why Python?
  - Installation
  - **Terminal use and scripts**
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

# Terminal use and Scripts.

Refer to [pdfs/python\\_terminal\\_and\\_scripts.pdf](#)

- Run from terminal or miniconda terminal
- Scripts always end in **.py**
- Use of text editors with syntax highlighting such as [Visual Studio Code](#) or [Atom](#)

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- **Jupyter Notebooks:**
  - **What and why?**
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

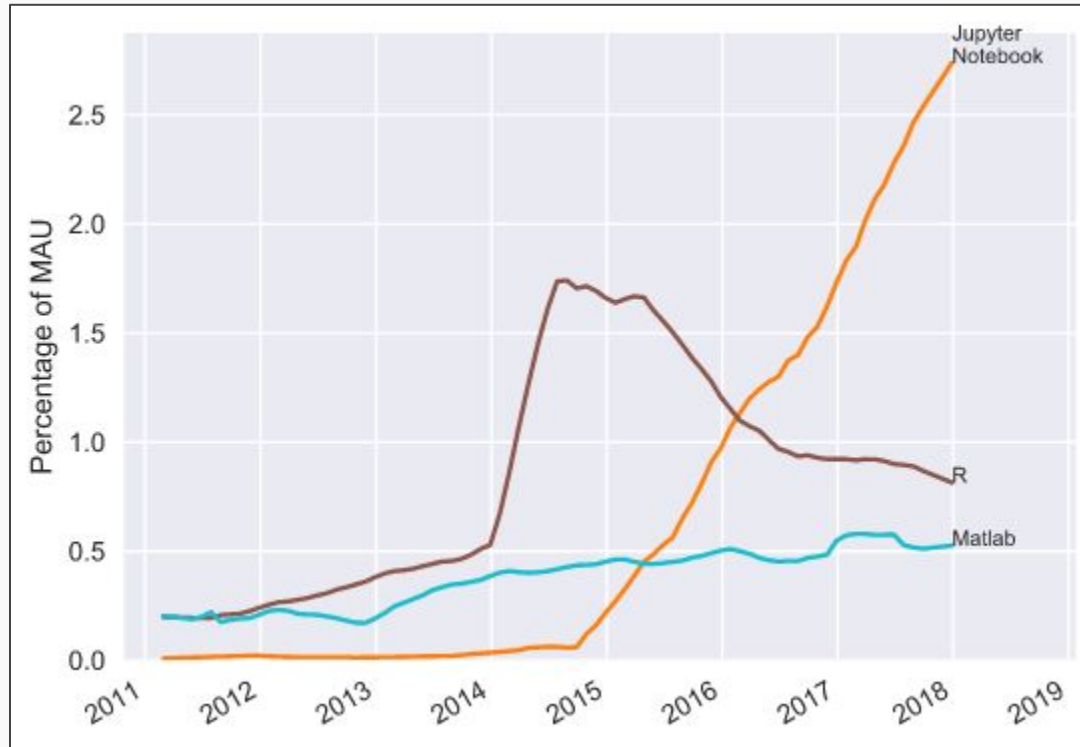
# Jupyter: What and Why?

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.

[Tutorial](#) | [Useful shortcuts](#)



# Jupyter: What and Why?



Source: GitHub

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- **Jupyter Notebooks:**
  - What and why?
  - **How?**
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources



# Installing and using Jupyter Notebook

Again refer to [pdfs/python\\_training\\_setup\\_guide.pdf](#)



# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- **Python Basics:**
  - **Data Types and Data Structures**
  - Variables and Statements
  - Functions and Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

# Data Types and Structures

## Data Types:

- Integer
- Float
- String
- Boolean

## Data Structures:

- List
- Dictionary
- Tuple
- Set
- DataFrame

Refer to [notebooks/python\\_basics\\_part1.ipynb](#)

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- **Python Basics:**
  - Data Types and Data Structures
  - **Variables and Statements**
  - Functions and Packages
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

# Data Types and Structures

- Variables
- Loops (**for** and **while**)
  - break/continue
- if/elif/else
  - and/or
- Commenting:
  - **First rule of programming: Always Comment!**
- Errors and Exception Catching.

Refer to [notebooks/python\\_basics\\_part1.ipynb](#)

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - **Functions and Packages**
- Python for Data Science:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

# Functions and Packages

- **Functions**

- **methods** are also functions but within a **Class** object. Classes will not be covered in this training (but are what make python and object-oriented programming language)!
- On how to call a function from a script, refer to [pdfs/python\\_terminal\\_and\\_scripts.pdf](#)

- Best **Packages** for data science and scientific computing:

- **Jupyter**: <https://jupyter.org/>
- **Numpy**: <https://numpy.org/>
- **Scipy**: <https://www.scipy.org/>
- **Pandas**: <https://pandas.pydata.org/>
- **Matplotlib**: <https://matplotlib.org>
- **Seaborn**: <https://seaborn.pydata.org/>
- **sklearn**: <https://scikit-learn.org/>
- **TensorFlow**: <https://www.tensorflow.org/>

Refer to [notebooks/python\\_basics\\_part2.ipynb](#)

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- **Python for Data Science and Scientific Computing:**
  - **Numpy, Scipy and Pandas**
  - **Matplotlib and Seaborn**
  - **Scikit-learn and TensorFlow**
- Data Cleaning and Visualisation Example:
  - Scores for MAW data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources



# Python for Data Science and Scientific Computing:

## Tutorials:

- [Jupyter](#): (Notebooks interface to handle and visualize data)
- [Numpy](#): (Mostly written in C, methods for mathematics and data handling)
- [Scipy](#): (Based on Numpy, scientific computations and methods)
- [Pandas](#): (Based on Numpy, offers Dataframes and eases data manipulation)
- [Matplotlib](#): (Python's core graphics library, can plot anything from basics graphs to 3D)
- [Seaborn](#): (Beautifies and simplifies matplotlib plotting, many other packages do so as well)
- [sklearn](#): (Based on Numpy and Matplotlib, offers more advanced statistical tools)
- [TensorFlow](#): (Recent packages with amazing tools for machine learning, notably deep-learning)

Refer to [notebooks/python\\_basics\\_part2.ipynb](#)

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science and Scientific Computing:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- **Data Cleaning and Visualisation**  
**Example:**
  - **Scores for MAWS CTQ data questionnaires.**
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - Python Learning Resources

## MAWS Data Cleaning and Visualisation Example:

Refer to [notebooks/CTQ\\_MAWS\\_A1.ipynb](#)

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science and Scientific Computing:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAWS CTQ data questionnaires.
- **Further Ressources and Trainings**
  - **Python Integrations**
  - [Stack Overflow and Google](#)
  - [Python Learning Resources](#)

# Python Integrations

- [Rpython](#) and/or [rpy2](#) : with R
- [Cython](#) : with C
- [Jython](#) : With Java

**BUT** best of all is to save your data to **CSV** or **JSON** formats, which all these languages can also read!

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science and Scientific Computing:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAWS CTQ data questionnaires.
- **Further Ressources and Trainings**
  - Python Integrations
  - **Stack Overflow and Google**
  - [Python Learning Resources](#)

# Stack Overflow and Google



## The Third rule of programming: Google it!

### Does Python have a string 'contains' substring method?

Asked 9 years, 2 months ago · Active 13 days ago · Viewed 3.8m times

I'm looking for a `string.contains` or `string.indexof` method in Python.

3602 I want to do:

```
if not somestring.contains("blah"):
    continue
```



457

python string substring contains

share

edited May 26 '17 at 18:02



Peter Mortensen

14.5k · 19 · 89 · 118

asked Aug 9 '10 at 2:52



Blankman

103k · 277 · 677 · 1061

10 Answers

active oldest votes

You can use the [in operator](#):

5735

```
if "blah" not in somestring:
    continue
```



share improve this answer

edited Nov 11 '15 at 23:30

answered Aug 9 '10 at 2:56



Michael Mrozek

127k · 20 · 147 · 156

161 Under the hood, Python will use `__contains__(self, item)`, `__iter__(self)`, and `__getitem__(self, key)` in that order to determine whether an item lies in a given contains. Implement at least one of those methods to make `in` available to your custom type. – [BallpointBen](#) Aug 17 '18 at 7:02

17 Just make sure that somestring won't be None. Otherwise you get a `TypeError: argument of type 'NoneType' is not iterable`. – [Nan Zhong](#) Oct 10 '18 at 22:44

5 FWIW, this is the idiomatic way to accomplish said goal. – [Trenton](#) Nov 13 '18 at 21:41

4 For strings, does the Python `in` operator use the Rabin-Carp algorithm? – [Sam Chats](#) Dec 18 '18 at 20:23

1 This is inconsistent and ugly in code like `".so." in filename or filename.endswith(".blah")`. – [Kaz](#) Feb 12 at 20:24

# Outline

- Python:
  - What is it?
  - Why Python?
  - Installation
  - Terminal use and scripts
- Jupyter Notebooks:
  - What and why?
  - How?
- Python Basics:
  - Data Types and Data Structures
  - Variables and Statements
  - Functions and Packages
- Python for Data Science and Scientific Computing:
  - Numpy, Scipy and Pandas
  - Matplotlib and Seaborn
  - Scikit-learn and TensorFlow
- Data Cleaning and Visualisation Example:
  - Scores for MAWS CTQ data questionnaires.
- Further Ressources and Trainings
  - Python Integrations
  - Stack Overflow and Google
  - **Python Learning Resources**



# Learning Resources

<https://www.codecademy.com/catalog/language/python>

<https://learnpythonthehardway.org/book/>

<https://realpython.com/learning-paths/python3-introduction/>

[edX](#) or [coursera](#)

[Stackoverflow](#) and [Towards Data Science](#) on Medium

GOOGLE!

# The rules of programming

- Always comment
- Never repeat yourself
- Google it!
- You learn by doing

Thanks!

**pedrobordignon@hotmail.com**

**[https://github.com/pedrodcB/python\\_training](https://github.com/pedrodcB/python_training)**