Introduction to Python and Jupyter Notebook

Course: Introduction to AI (Part I)

Computer Science Bachelor - AAI

University of Applied Studies Rosenheim, Germany

Technische
Hochschule
Rosenheim

About the history ...

- In late 1980s Guido Van Rossum started to work on Python at the National Research Institute for Mathematics and Computer Science in the Netherlands
- In 2001 Fernando Pérez started developing IPython (= *Interactive Python*)
    - *shell*/ *console* based
- From 2005 onward the IPython team has tried to develop a *web-based* notebook system
    - The idea of a *notebook* was adapted from *Mathematica notebooks* and *Maple worksheets*
    - The Mathematica notebooks were created as a front end or GUI in 1988 by Theodore Gray.
- 2014 The **Project Jupyter** started as a sping-off from IPython

... more details on **https://www.datacamp.com**

/community/blog/ipython-jupyter

What is Python?

Python ([ˈpʰyːtɔn]) is an interpreted, object-oriented, high-level programming language with dynamic semantics
Python: Dynamic programming language which supports several different programing paradigms:

- Procedural programming
- Object oriented programming
- Functional programming
- Standard: Python byte code is executed in the Python interpreter (similar to Java) → platform independent code

TIP: Use Python 3 (Support for Python2 has ended in 2020!)
Note: The Name *Python* is not influenced by the snake - it is more triggered by *Monty Python's Flying Circus*

Python Resources

Getting Started with Python:

- **https://www.codecademy.com/learn/python**
- **http://docs.python-guide.org/en/latest/intro/learning/**
- **https://learnpythonthehardway.org/book/**
- **https://www.codementor.io/learn-python-online**
- **https://websitesetup.org/python-cheat-sheet/**

Python Reference:

- **https://docs.python.org/3.5/reference/**

Python Statements

Python is an imperative language based on statements. That is, programs in Python consists of lines composed of statements. A statement can be:

- a single expression
- an assignment
- a function call
- a function definition
- a statement; statement

Expressions

Numbers

```
In [12]:
1 + 4
```

```
Out[12]:
```

```
5
```

```
In [13]:
6+6
```

```
Out[13]:
```

```
12
```

This is text!

Strings

```
In [14]:
a = "apple"
b = 'orange'
print(a+b)
```

```
appleorange
```

Other things to know

- type(a)
- if ... else ...
- for n in range(10)

```
In [ ]:
```

Lists and Dicts

Python has three very useful data structures built into the language:

- dictionaries (hash tables): {}
- lists: []
- tuples: (item, ...)

In [15]:

```
A = [1, 2, 3]
B = {"apple": "a fruit", "banana": "an herb", "monkey": "a mammal"}
```

In [16]:

```
print(A[1])
print(A[1:])
print(A[:1])

print(B["apple"])
```

```
2
[2, 3]
[1]
a fruit
```

Comprehensions

"List comprehension" is the idea of writing some code inside of a list that will generate a list. Consider the following:

```
In [17]:
X = [x ** 2 for x in range(10)]
print(X)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Defining Functions

In [18]:

```
def plus(a, b):
    return a + b

plus(1,2)
```

Out[18]:

```
3
```

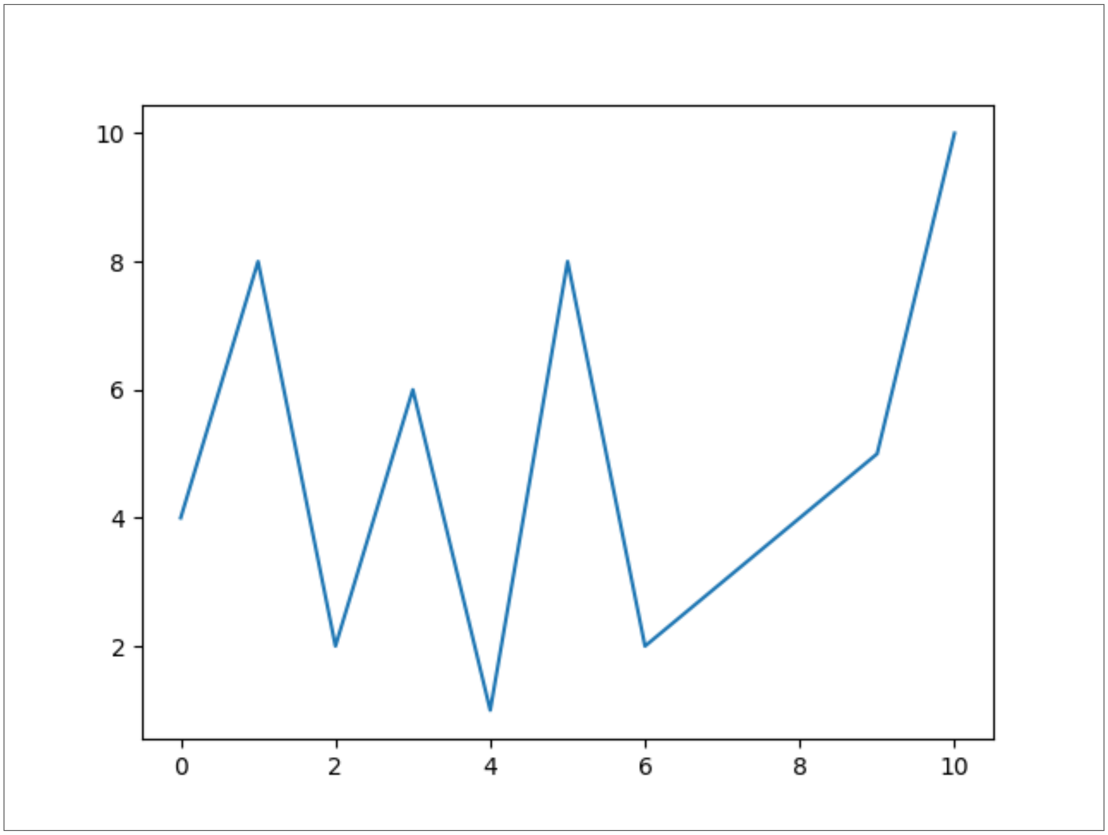
External Libs

In [19]:

```
                %matplotlib notebook

import matplotlib.pyplot as plt
from math import sin


plt.plot([4, 8, 2, 6, 1, 8, 2, 3, 4, 5, 10])
```



Out[19]:

```
[<matplotlib.lines.Line2D at 0x7f6f6ec63
5b0>]
```

What is a “Notebook”?

- A notebook integrates various elements (so-called *Cells*) and its output into a single document
- *Cells* can have different types
    - code (pretty_printed)
    - visualizations
    - narrative and formatted text
    - mathematical equations
    - and other media (since it is HTML you can even embed YouTube videos)

A notebook is an *HTML document with interpreted code*

Cells in Notebooks

- Each section in a notebook is called a **Cell**
- Cells can hold text (such as Markdown, HTML, or LaTeX) or Code
- *Click* the Cell Menu at the top of this page to set what Cells can do.
- *Double-click* (or *ENTER*) a Cell to edit it
- Run a code cell using Shift-Enter or pressing the **Run*** button in the toolbar above
- Markdown will show with all the Markdown Codes until you Run them.
- Code runs and displays an output below the cell.

Hello!

```
In [20]:  print("hello")
```

```
hello
```

In [21]:
```
print("hello again")
```

```
hello again
```

In [22]:
```
print("")
```

```

```

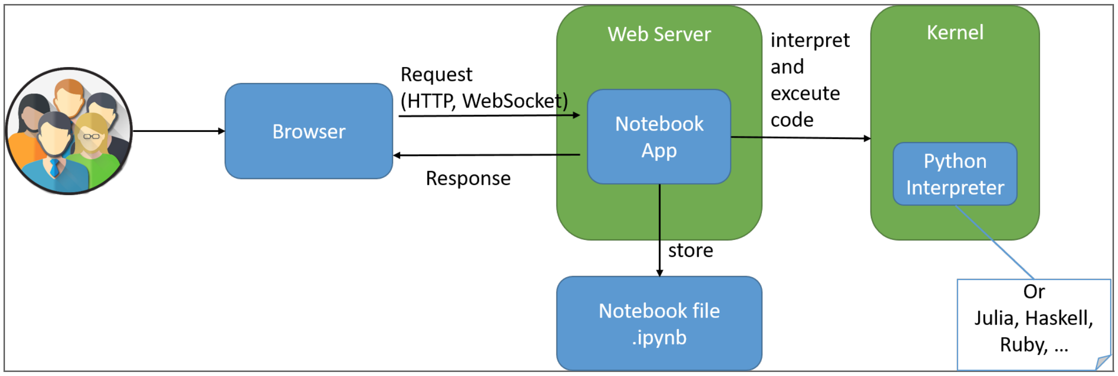
This is a cell!

In [23]:
```
# here can go something
```

Architecture

A notebook runs as a web app

- A web server hosts the notebook web app
- A kernel is used to interpret and to execute cells (code, Markdown, ...)
- The web app manages the notebook file



How To Install Jupyter Notebook

- The classical way
  - Install Python from **https://www.python.org/**
  - `pip3 install jupyter`
- The lazy way
  - Install Anaconda Python distribution: **https://www.anaconda.com/**
- The modern way
  - Install docker
  - and do a pull with an notebook image
    - `docker pull jupyter/scipy-nodebook`
    - `docker pull tensorflow/tensorflow:la py3`
- the hosted way
  - or Google:

https://colab.research.google.com

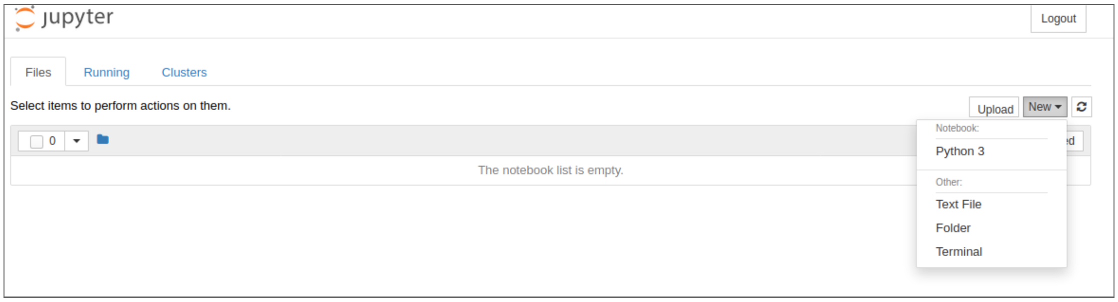
Run a Jupyter Notebook server

- From Python: `$ jupyter notebook`
- From Anaconda: `C:> jupyter notebook`
- With Docker image: `docker run --rm -it -p 8888:8888 -v "$(pwd):/notebooks" jupyter/scipy-notebook`

In all cases the server starts in working directory and listens to port 8888!
-> Open a browser and use address `http://localhost:8888`

Create a new Notebook

- You can create folder, files and notebooks via browser
- Go to "New" and create a new notebook



Select items to perform actions on them.

The notebook list is empty.

- And now? But what should I do with my new notebook?

A Notebook document

Notebook documents (or “notebooks”, all lower case) are documents produced by the Jupyter Notebook App, which contain both

- computer code (e.g. python):

```
In [24]:
print("Hello World!")
```

```
Hello World!
```

- and rich text elements (paragraph, equations, figures, links, etc…).

$$\frac{d}{dx}c^n = nx^{n-1}$$

Now...

... you have an environment

- to write code (Here: Python)
- to run code
- to check results (event visualize it!)
- and to add documentation!

Markdown

- **Markdown** is a *simplified* markup language for text.
- Use it for general text and simple graphics. You can read more about Markdown here: **https://www.markdowntutorial.com/**
- And there's a great cheat-sheet on Markdown here: **https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet**

(Double-click this cell to see an example of Markdown in action)

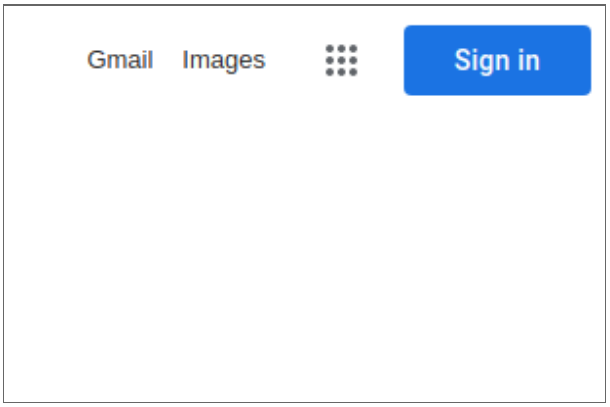
Hosted Environments

Azure Notebooks (**https://notebooks.azure.com/**) helps you to get started quickly on prototyping, data science, academic research, or learning to program Python:

- A data scientist has instant access to a full Anaconda environment with no installation.
- A teacher can provide a hassle-free Python environment to students.
- A presenter can give a like talk or webinar without asking attendees to spend 45 mins installing software.
- A developer or hobbyist can use Notebooks as a quick code scratchpad.

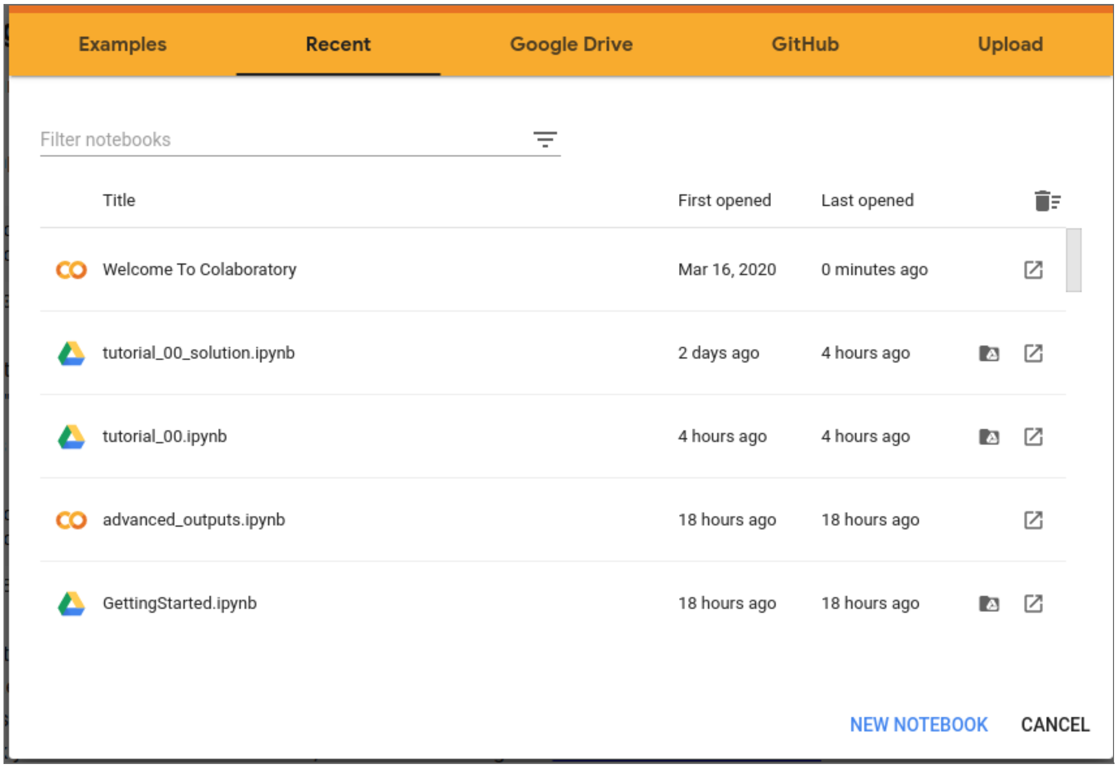
Setup an Google Account

- To sign in to Google Colab a Google Account is required
    - This any email address registered at Google
    - In case you do not want to use your existing; create a new one
    - Go to **https://google.com/** and *Sign In* on top right
    - From here: *Create one* or use an existing one



Google Colab

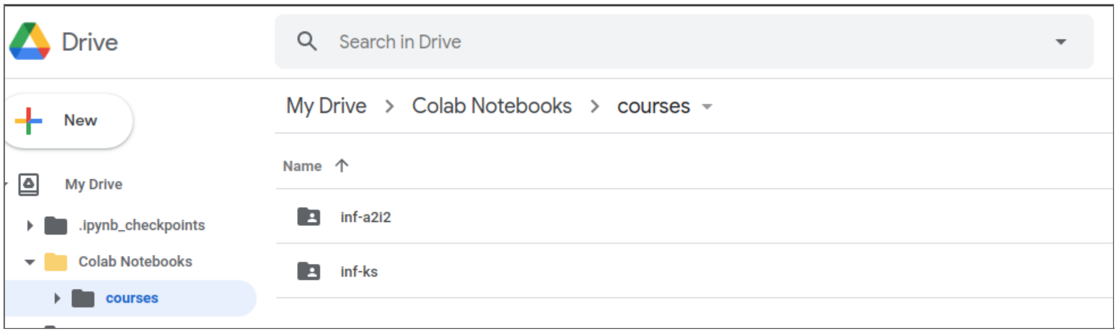
- You can manage your projects on Google Colab (**https://colab.research.google.com/**)
    - public or private visibility
- Projects upload from Github or file system



Title	First opened	Last opened
Welcome To Colaboratory	Mar 16, 2020	0 minutes ago
tutorial_00_solution.ipynb	2 days ago	4 hours ago
tutorial_00.ipynb	4 hours ago	4 hours ago
advanced_outputs.ipynb	18 hours ago	18 hours ago
GettingStarted.ipynb	18 hours ago	18 hours ago

Project Structure

- A project can contain
  - Jupyter Notebooks (.ipynb)
  - Files , e.g. Readme.md or Images.png
  - Folder, e.g. images
- Folders and Files can be shared



My Drive > Colab Notebooks > courses

Name ↑
inf-a2i2
inf-ks

References and Links

There's a lot more to learn about Notebooks!

- Learn more about Jupyter Notebooks here: **https://jupyter-notebook.readthedocs.io/en/stable/**
- A gallery of interesting Notebooks - **https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks**
- Learn about Python and Notebooks here: **https://github.com/jdwittenauer/ipython-notebooks/blob/master/notebooks/language/Intro.ipynb**
- Learn about R and Notebooks here: **https://www.datacamp.com/community/blog/jupyter-notebook-r**

```
In [ ]:
```