

SCIENTIFIC PYTHON.



Before we start...

The interactive part of this Powercourse requires a supplementary repository (which builds a large docker image) and some external datasets. Please take this moment to start cloning the repository and running `make start`.

Powercourse Python Data Science repository

<https://github.com/RineshRamadhin/Powercourse-Python-Data-Science>

Requirements: Docker, terminal, browser

Setup

We  **Data**

What now?!

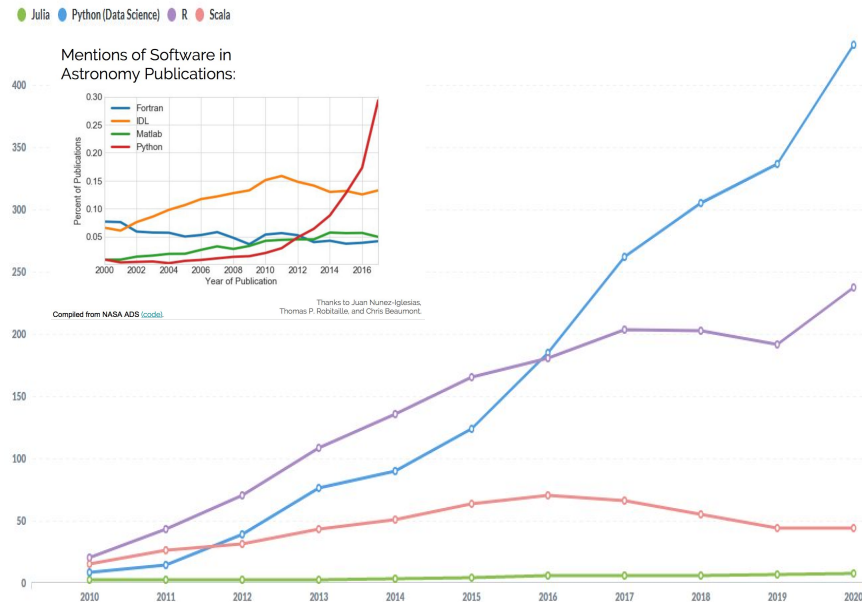
- Why Python?
- The scientific ecosystem
- Interactive Python (IPython)
- Data manipulations
- Interactive examples



Scientific Python

Some reasons

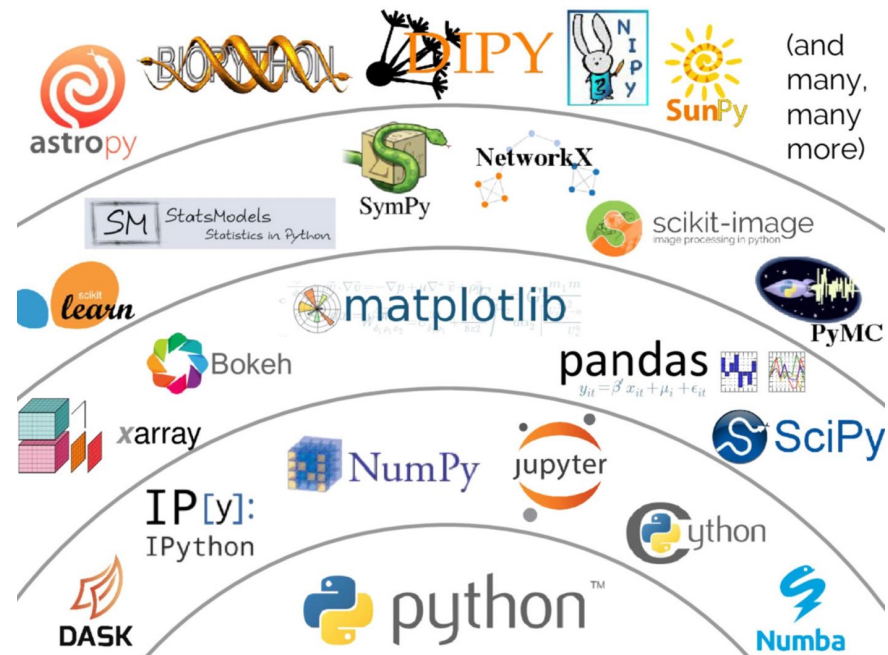
- Simplicity & dynamic nature
 - High-level, interpreted language
 - Platform independent
- Widely used in scientific and numeric computing
 - Large ecosystem, interoperability
- Scripting support
- Large community 🚀
 - Most popular for data science (by far)



Why Python?

Batteries Included

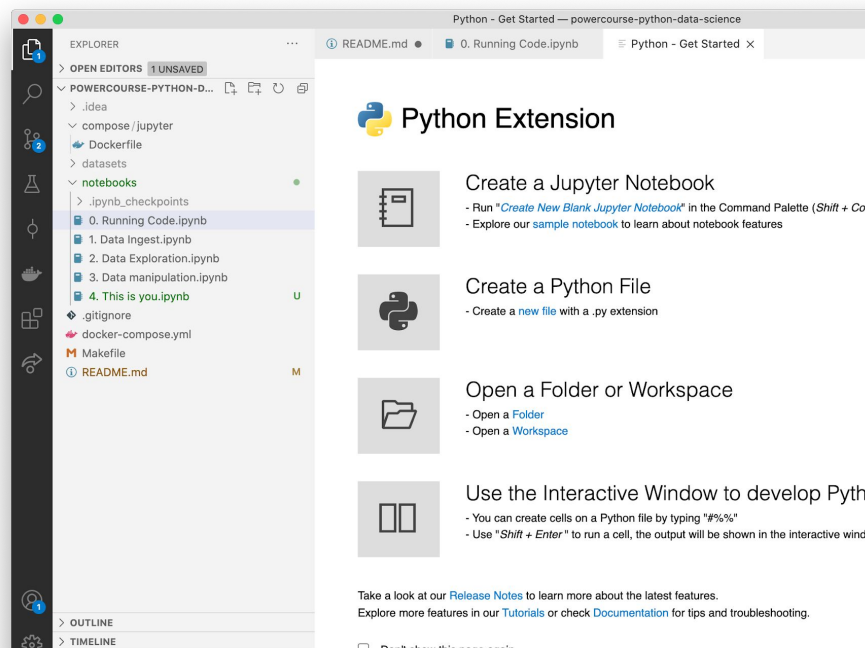
- Already comes with a lot of scientific requirements
 - Scraping, working with file systems, csv, etc.
- Giant Scientific Python (SciPy) Ecosystem
 - NumPy for array-computing, Pandas for DataFrames, Matplotlib for data visualization, Scikit-learn & TensorFlow for Machine Learning, etc.
- Data science specific tools...



The scientific ecosystem

You can do it yourself

- REPL (Read-eval-print loop)
- Feature rich
 - Syntax highlighting, tab completion, built-in docs, Unix support, history
- **Input & output caching**
- Google Colab, Jupyter notebook & Labs
- Support in IDE's and VSC
 - PyCharm, VS Code, GitHub, etc



Interactive Python (IPython)

print(getmembers(pd, isfunction))

- Python is powerful 💪
 - Data type manipulations
 - Grouping
 - Aggregation
 - Transformation
 - Filtration
 - Pivot Tables
 - Joining Data

```
# Reading data from a csv file:  
df = pd.read_csv('students.csv')
```

```
# Selecting rows where age is over 20  
df[df.age > 20]
```

```
# Selecting rows where name is not John  
df[df.name != "John"]
```

```
# Selecting rows where age is less than 10  
# OR greater than 70  
df[(df.age < 10) | (df.age > 70)]
```

```
data.loc[(data["Gender"]=="Female") & (data["Education"]=="Not Graduate") & (data["Loan_Status"]=="Y"), ["Gender","Education","Loan_Status"]]
```

```
# This function doubles the input value  
def double(x):  
    return 2*x
```

```
# Apply this function to double every value in  
a specified column  
df.column1 = df.column1.apply(double)
```

can also be supplied to

```
column2.apply(lambda x : 3*x)
```

```
# Applying to a row requires it to be called on the  
iFrame
```

```
mn'] = df.apply(lambda row:  
mn1'] * 1.5 + row['column2'],
```

```
# Specifying each value in the new column:  
df['newColumn'] = [1, 2, 3, 4]
```

```
# Setting each row in the new column to the same  
value:  
df['newColumn'] = 1
```

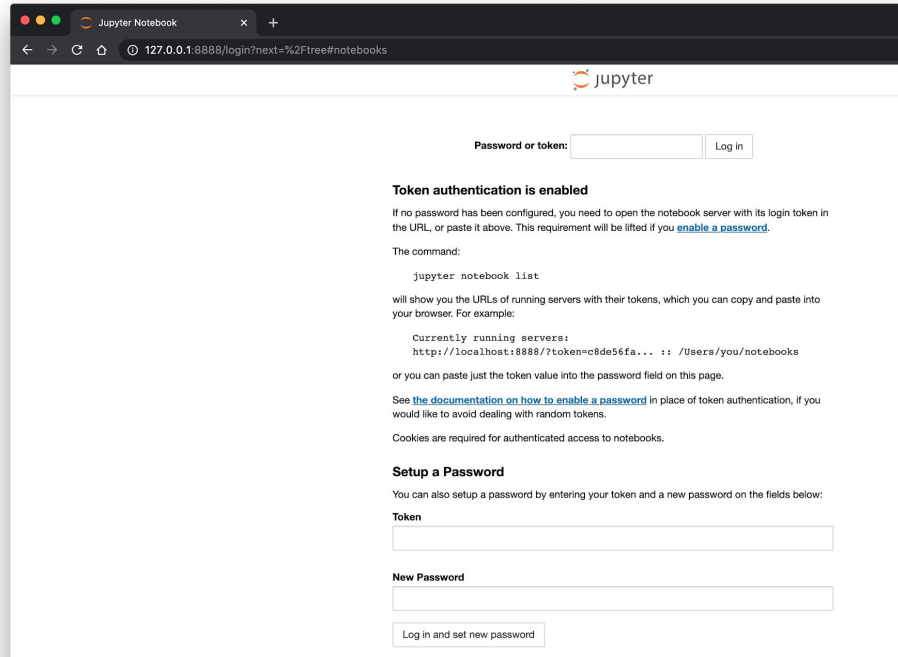
```
# Creating a new column by doing a  
# calculation on an existing column:  
df['newColumn'] = df['oldColumn'] * 5
```

```
companies.groupby('sector').filter(  
    lambda x: x['employees'].sum() > 1000000  
)[['name', 'employees']]
```

Data manipulations

Now do it yourself

- Repo ready?
- Google Colab also possible to try out.
 - <https://colab.research.google.com/notebooks/intro.ipynb>



Interactive examples

An orange circle containing the text "cool blue" in white, lowercase, sans-serif font.

cool
blue

QUESTIONS?

Not all at once.

An orange circle in the top right corner containing the text "cool blue" in white.

cool
blue

THANK YOU.
You were here.



alles voor een glimlach[😊]