First Steps with Python

Mon 28th October 2019 5pm - 6:30pm

PyData Bristol

coefficient

WHAT IS PYTHON?

What is Python?

- Created by Guido Van Rossum in 1991

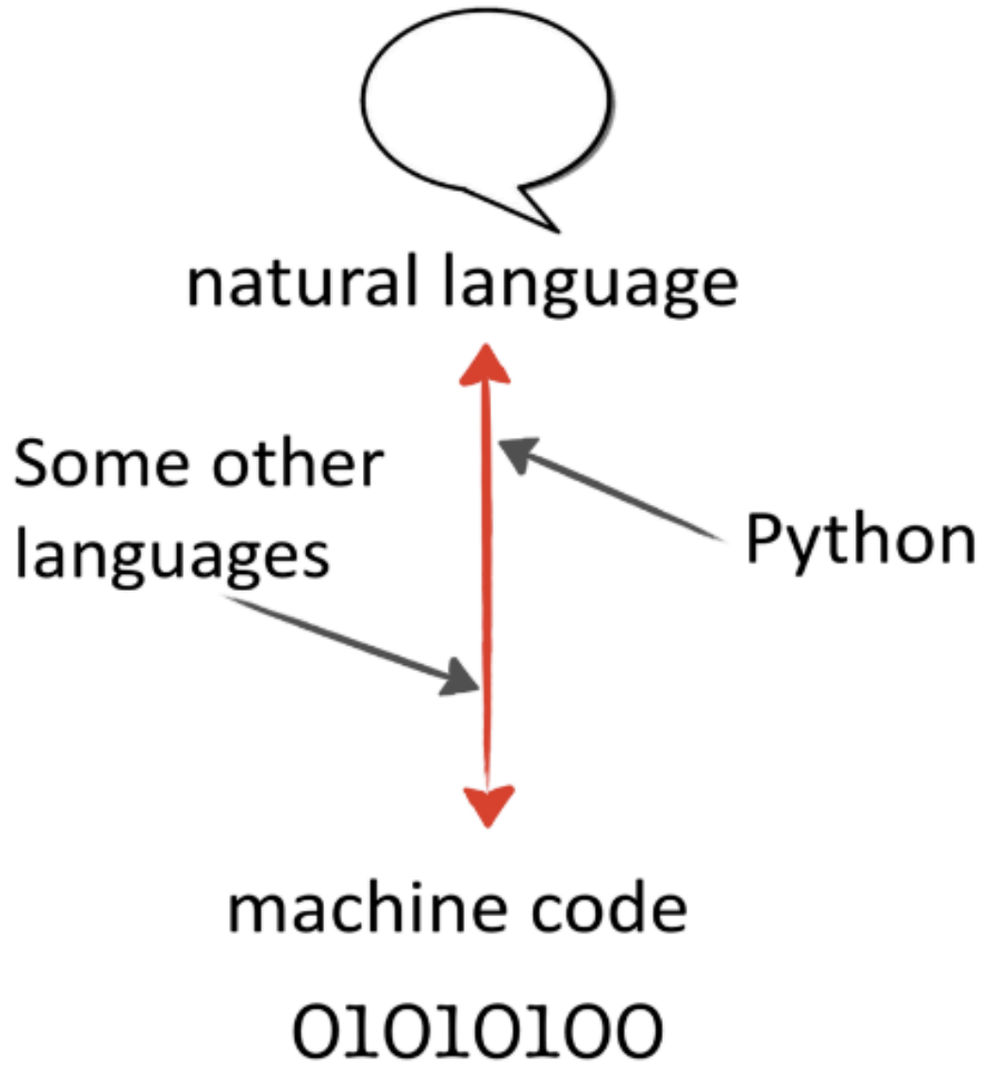
- Emphasises **productivity** and **code readability**

- The language is **easy to pick up** and learn

- **Readable code** means that almost anyone can pick up a piece of code and understand what it is doing

Why Python?

- **Human readable**
- **High level & general**
- **Cross-platform**
- **Free & open source**
- **Strong & friendly community**



Why Python?

- **Simple** & **clean** syntax
- Extremely **fun** to develop in :)
- Basically **everything** can be done with Python: many purposes outside data science
- Things can be done **quickly**!
    - For example a program that takes you weeks to write in C++, might take you a day in Python.

Where is Python used?

- Everywhere!
- Industry
- Academia & research
- Web Development / Web Apps
- Art & music
- Game Development
- Windows Applications

Powered by Python

Powered by Python

EVE ONLINE

SurveyMonkey

Atlassian

citi

Gartner

Google

Honeywell

EVERNOTE

NASA

BNP PARIBAS

DISQUS

TOYOTA

INDUSTRIAL LIGHT & MAGIC
A LUCASFILM COMPANY

The Washington Post

Eventbrite

the ONION

splunk>

LAB:

TOOLS OF THE TRADE

The Python Shell (February 1991)

- Search your computer for "**Anaconda Prompt**". You are now using the command line interface (CLI)!

- A python shell is similar to a Command Line Terminal and it can be launched by typing:
  - **python**

```
(base) ➜  ~ python
Python 3.6.6 | packaged by conda-forge | (default, Jul 26 2018, 09:55:02)
[GCC 4.2.1 Compatible Apple LLVM 6.1.0 (clang-602.0.53)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 2
3
>>> x = 3
>>> x
3
>>> quit()
```

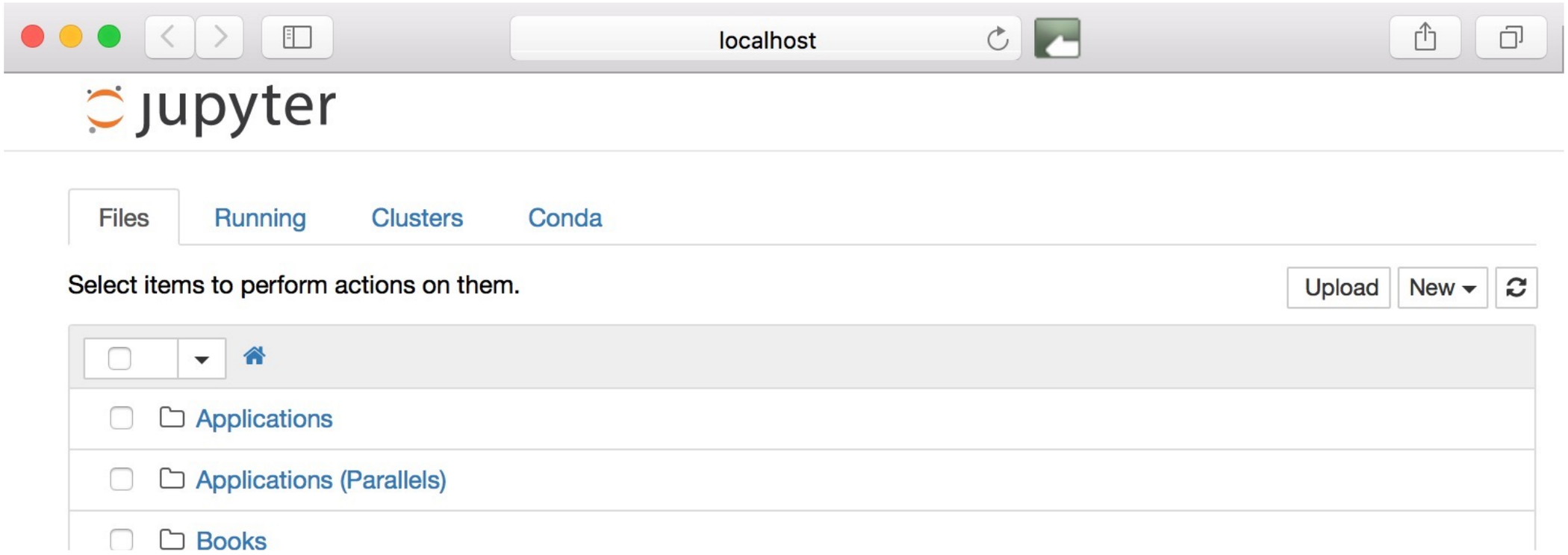
IPython Shell (2001)

- An **IPython shell** is more interesting than a plain terminal, providing syntax colouring and shortcuts to interact with our code. It can be launched with:
  - **ipython**

```
(base) ➜  ~ ipython
Python 3.6.6 | packaged by conda-forge | (default, Jul 26 2018, 09:55:02)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.0.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: 1+1
Out[1]: 2

In [2]: x = "Hello"

In [3]: x
Out[3]: 'Hello'

In [4]: quit()
```

Jupyter Notebook (2014)

- Jupyter notebook is a web interface that let's us use formatting along side our code. It is the extremely common and very useful! You can launch it by typing:
  - **jupyter notebook**
- Click **Desktop** to navigate into that folder.
- On the top-right-hand-side click "New" then "Python 3".



LAB:

JUPYTER NOTEBOOK

LAB:
LET'S WRITE
SOME PYTHON

Python as a calculator

Try each line inside **separate** Jupyter Notebook cells (type Shift-Enter to run):

```
1 + 1
```

```
50 - 2 * 4
```

```
8 / 5
```

```
3 ** 2
```

"Hello World"

- Compare this typical "Hello World!" program in Python...

```
print("Hello World")
```

- ...with the same program in C++

C++

```cpp
#include<iostream>
using namespace std;

int main()
{
    cout << "Hello World!";
    return 0;
}
```

Variables

- Variables stores values under a specified name:

```
cup = "coffee"
print(cup)

cup = 5
print(cup)

cup = 3.14
print(cup)
```

Operators

- Try the following:

```
# This is a comment

# Comparing numbers
2 > 1

# Testing for equality
2 == 2

# "Boolean comparisons"
(8 > 0) and (3 != 3)
(5 >= 5) or (3 != 3)
```

Lists

- Lists are collections of objects

```
my_list = ["a", "b", "c", "d"]

type(my_list)

my_list

my_list[1:3]
```

- Python starts counting from zero!

```
my_list[0]
```

Lists

- Lists can be changed

```
my_list
my_list[3] = "changed!"
my_list
```

Dictionaries

- Dictionaries combine keys with values in pairs
- Like in a dictionary, you can search the keys to obtain their corresponding value

```
my_dict = {'x': 'hello','y': 10}
my_dict
my_dict['x']
my_dict['z'] = [1, 2, 3]
my_dict
my_dict['z'][0]
```

Other data structures

- "**Data structures"** are a particular way of storing and organising data in a computer so that it can be used efficiently. Some examples include:
    - Tuples
    - Arrays
    - Matrices
    - Dataframes
- We will see some examples of these later.

PYTHON PROGRAMMING FUNDAMENTALS

Control structures

- **Control structures:** A block of programming that analyses variables and chooses a direction in which to go based on given parameters.

- The term flow control details the direction the program takes (which way program control “flows”). It determines how a computer will respond when given certain conditions and parameters. Some typical structures include:

  - **If** statement
  - **For** loop
  - **Functions**

if

- An **if** statement is a conditional structure that, if proved true, performs a function or displays information.

- Think of this as a decision that moves the flow of your program depending on the answer to a TRUE-FALSE question.

- In pseudocode:

```
IF a person is older than 18
THEN they can drive
ELSE they cannot drive
```

- In Python:

```
if age_person > 18:
    print("They can drive")
else:
    print("They cannot drive")
```

if

- Try this:

```
A = 10
B = 100
if A > B:
    print("A is bigger")
elif A == B:
    print("same")
else:
    print("B is bigger")
```

if

- Try this:

```
A = 10
B = 100

if A > B:
    print("A is bigger")
elif A == B:
    print("same")
else:
    print("B is bigger")
```

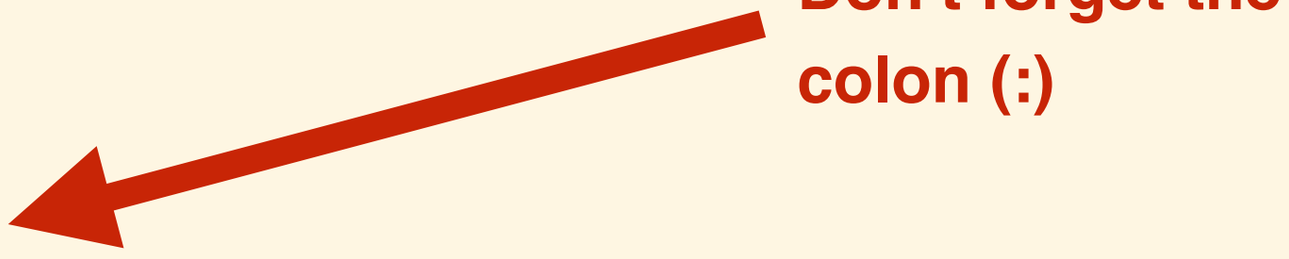
White space is very important in Python!

You can indent this with four spaces, or a tab (some text editors convert tabs to four spaces)

if

- Try this:

```
A = 10
B = 100
if A > B:
    print("A is bigger")
elif A == B:
    print("same")
else:
    print("B is bigger")
```



for loop

- A loop statement in programming performs a predefined set of instructions or tasks while or until a predetermined condition is met.

- Think of this as a repetitive action that has to be performed until further notice.

- In pseudocode:

```
FOR each user of a service in a list
   PRINT greet them
```

- In Python:

```
people = ["Boris", "Jeremy", "Jo"]

for person in people:
    print("Hello " + person)
```

for loop

- Try this:

```
for x in [1, 2, 3]:
    print(x)


for x in range(10):
    print(x**2)
```

Functions

- A function is a group of instructions used by programming languages to return a single result or a set of results.

- Functions are a convenient way to divide our code into useful blocks, providing us with order, making the code more readable and reusable.

- Here is how you define a function in Python:

```python
def function_name(input1, input2...):
    1st block of instructions
    2nd block of instructions
    ...
```

Functions

```
def square(x):
    return x**2


def add(x, y):
    return x + y

output = square(7)
print(output)
print(add(output, 3))
```

LAB:
WRITE A PYTHON PROGRAM

Activity: Write a Python program

- Write a Python script to solve one (or all!) of the following tasks:

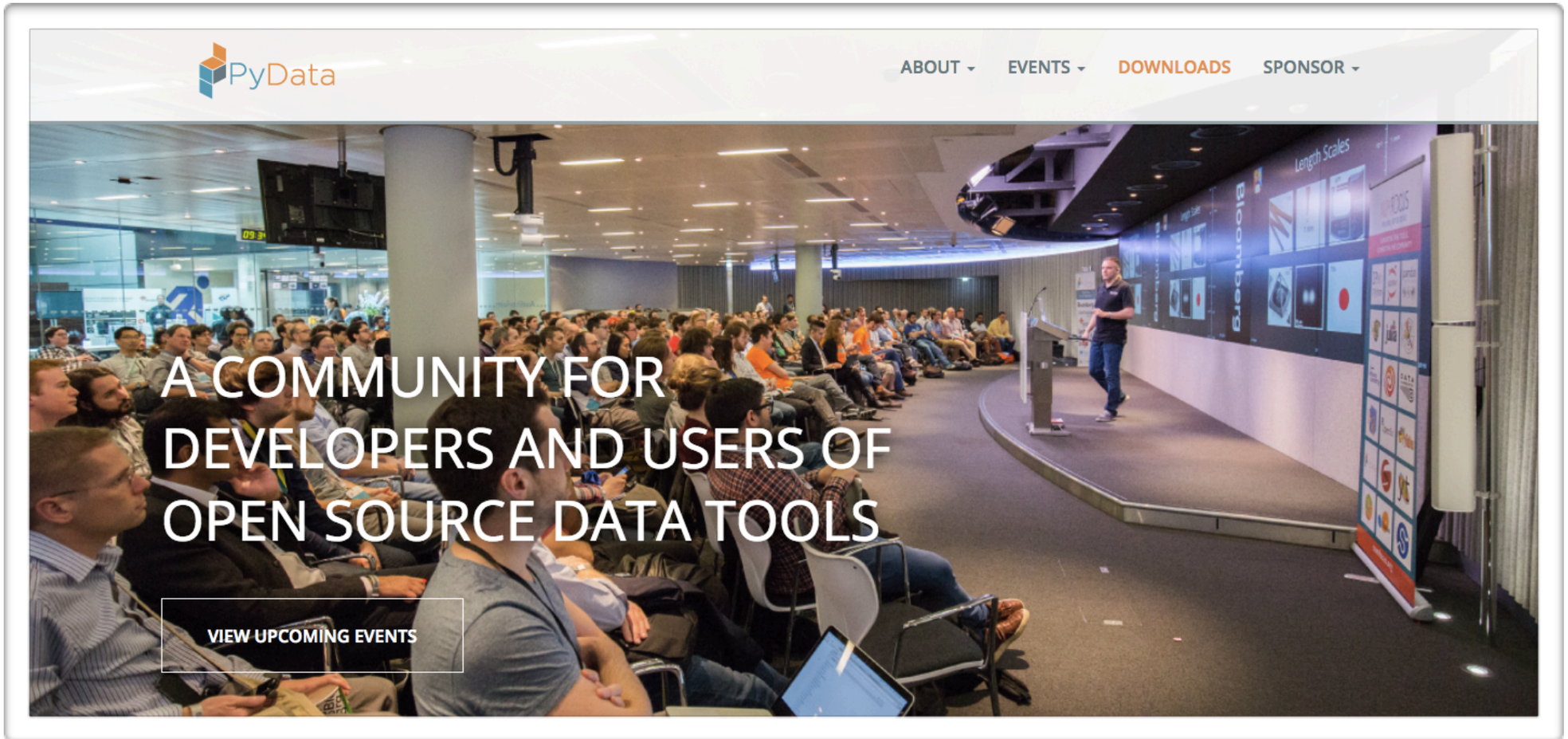
1. **Create a function that will calculate the area of a circle with radius r.**

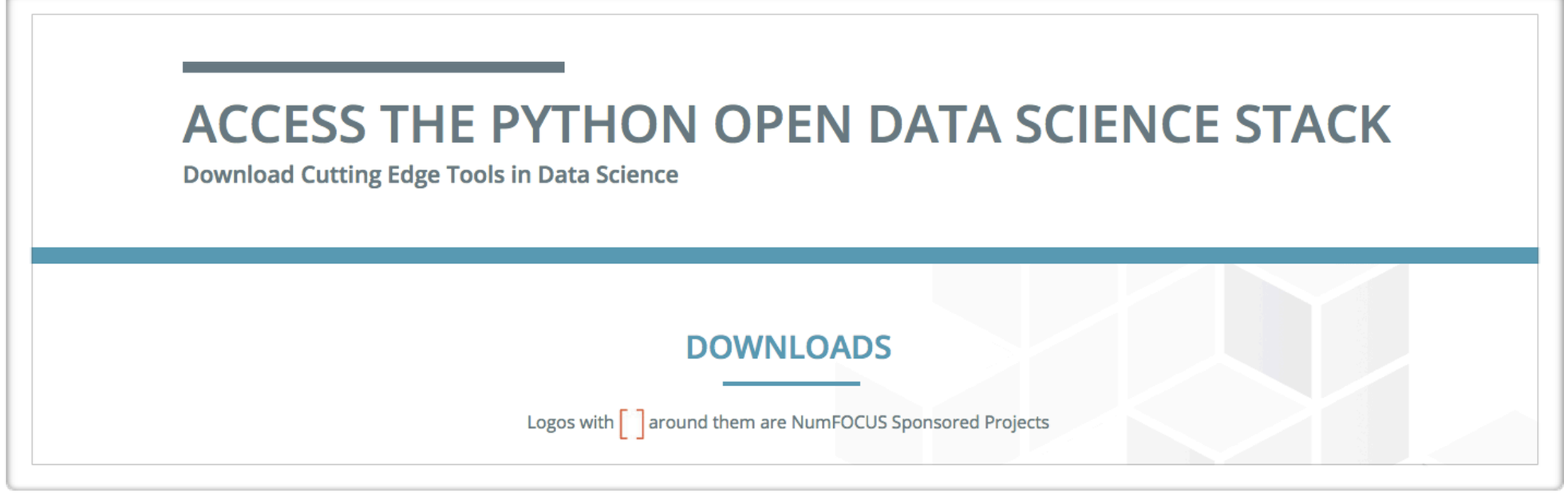
2. **For a given number, calculate and print the square of a number. If the given number is larger than 10 also calculate the cube.**

3. **List the letters in the sentence "Python is fun".**

LAB: PYTHON FUNDAMENTALS

The PyData Ecosystem





The PyData Ecosystem

FENICS'18
MARCH 21-23, 2018
Oxford, UK

PYDATA KAUNAS @ PYCON LITHUANIA
MAY 5, 2018

PYDATA EDINBURGH @ EUROPYTHON
JULY 25-29, 2018

PYDATA FLORENCE @ PYCON ITALY
APRIL 20-22, 2018

ROPENSCI UNCONF
MAY 21-22, 2018
Seattle, WA, USA

JULIACON
AUGUST 7-11, 2018
London, UK

PYDATA LONDON
APRIL 27-29, 2018

PYDATA AMSTERDAM
MAY 25-27, 2018

JUPYTERCON
AUGUST 21-24, 2018
New York, NY, USA

PYTHON IN ASTRONOMY
APRIL 30 - MAY 4, 2018
New York, NY, USA

PYDATA BERLIN
JULY 6-8, 2018

PYDATA CÓRDOBA
OCTOBER 1-2, 2018

PYDATA LOS ANGELES
OCTOBER 22-24, 2018

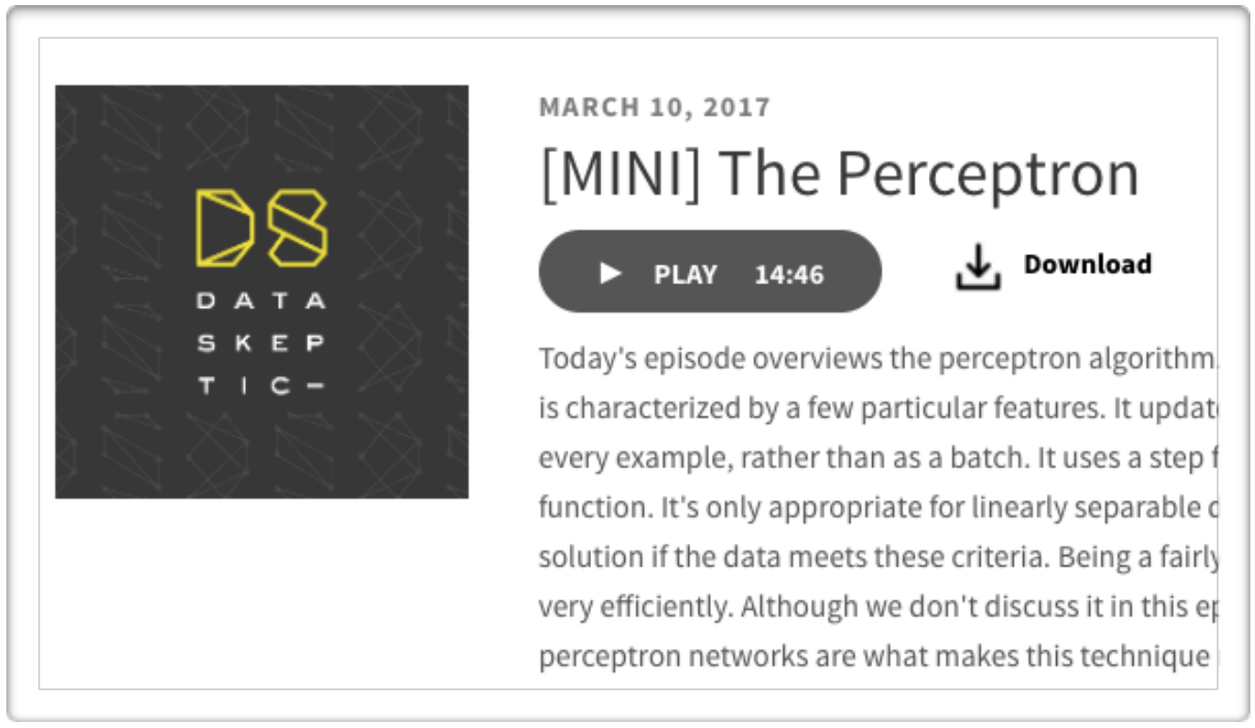
PYDATA KARLSRUHE & PYCON DE
OCTOBER 24-28, 2018

PYDATA WARSAW
NOVEMBER 19-20, 2018

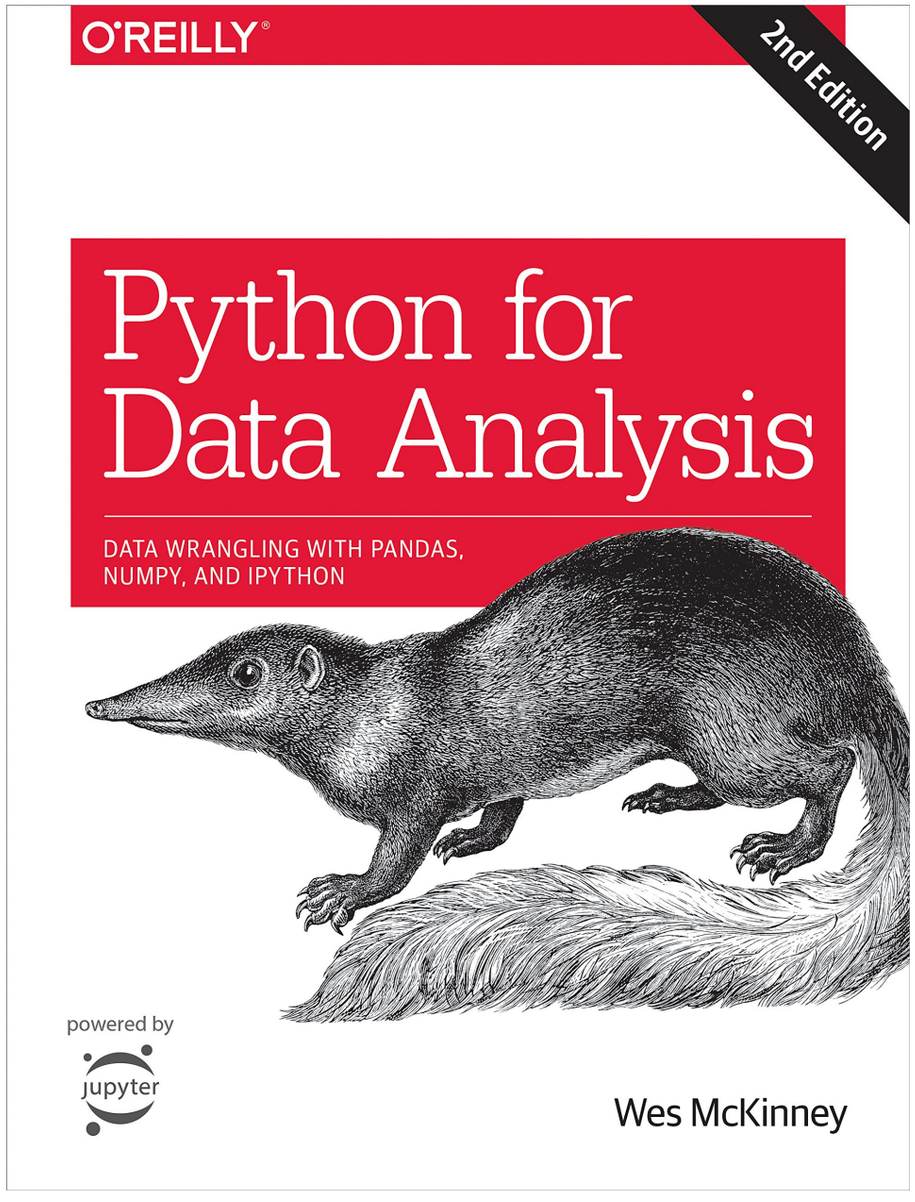
PYDATA NYC
NOVEMBER 2018

Learn More!

Data Skeptic Podcast



dataskeptic.com



kaggle.com/learn

THANK YOU

@john_sandall

PyData Bristol

coefficient