Python for Scientific Research

Bram Kuijper

University of Exeter, Penryn Campus, UK

March 6, 2019

Acknowledgements

- The workshop is funded by Exeter's researcher-led initiative award
- Big thanks to JJ Valletta as has he developed these lecture materials
- Big thanks to Deepak Kumar Panda and JJ for helping out today

UNIVERSITY OF EXETER | DOCTORAL COLLEGE

Researcher Development

Course Schedule

- Today, March 6: The basics of programming in Python
  - how to run Python code
  - data types
  - flow control
  - functions and modules
  - number crunching with `numpy` and `scipy`

Course Schedule

- Today, March 6: The basics of programming in Python
    - how to run Python code
    - data types
    - flow control
    - functions and modules
    - number crunching with `numpy` and `scipy`
- March 20th: Applying Python to simplify your life
    - number crunching with `numpy` and `scipy`
    - text manipulation
    - working with files
    - working with data using `pandas`
    - making graphs using `matplotlib`

Course Schedule

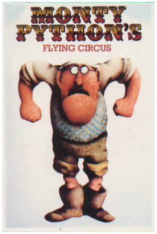
- Today, March 6: The basics of programming in Python
  - how to run Python code
  - data types
  - flow control
  - functions and modules
  - number crunching with `numpy` and `scipy`
- March 20th: Applying Python to simplify your life
  - number crunching with `numpy` and `scipy`
  - text manipulation
  - working with files
  - working with data using `pandas`
  - making graphs using `matplotlib`
- March 27th: Advanced subjects
  - object-oriented programming
  - automating tasks in MS-office
  - image manipulation
  - working on student-generated problems

Today's schedule

- 1300 - 1400: How to run Python
- 1400 - 1415: Break
- 1415 - 1500: Data types & flow control
- 1500 - 1515: Break
- 1515 - 1600: Functions & modules
- 1600 - 1615: Break
- 1615 - 1700: Numpy & scipy (if we get to it)

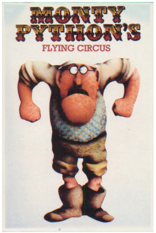
What is Python?

- A scripted, high-level programming language created by Guido Van Rossum and named after Monty Python's flying circus



What is Python?

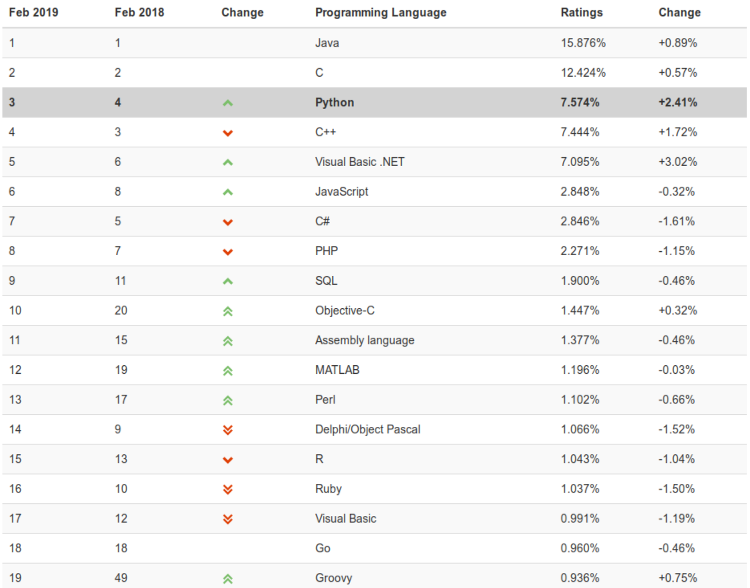
- A scripted, high-level programming language created by Guido Van Rossum and named after Monty Python's flying circus



- easy-to-use, highly standardized and with an emphasis on readability of code

Why use Python?

The TIOBE index is a measure of the popularity of programming languages:



Feb 2019	Feb 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.876%	+0.89%
2	2		C	12.424%	+0.57%
3	**4**	▲	**Python**	**7.574%**	**+2.41%**
4	3	▼	C++	7.444%	+1.72%
5	6	▲	Visual Basic .NET	7.095%	+3.02%
6	8	▲	JavaScript	2.848%	-0.32%
7	5	▼	C#	2.846%	-1.61%
8	7	▼	PHP	2.271%	-1.15%
9	11	▲	SQL	1.900%	-0.46%
10	20	▲▲	Objective-C	1.447%	+0.32%
11	15	▲▲	Assembly language	1.377%	-0.46%
12	19	▲▲	MATLAB	1.196%	-0.03%
13	17	▲▲	Perl	1.102%	-0.66%
14	9	▼▼	Delphi/Object Pascal	1.066%	-1.52%
15	13	▼	R	1.043%	-1.04%
16	10	▼▼	Ruby	1.037%	-1.50%
17	12	▼▼	Visual Basic	0.991%	-1.19%
18	18		Go	0.960%	-0.46%
19	49	▲▲	Groovy	0.936%	+0.75%

Why Python?

- It is free! No licence costs

Why Python?

- It is free! No licence costs
- Runs on all platforms (Mac, Windows, Linux)

Why Python?

- It is free! No licence costs
- Runs on all platforms (Mac, Windows, Linux)
- Because of it’s ease of programming (e.g no need to worry about memory allocation), Python minimises development effort

Why Python?

- It is free! No licence costs
- Runs on all platforms (Mac, Windows, Linux)
- Because of it's ease of programming (e.g no need to worry about memory allocation), Python minimises development effort
- A huge number of libraries, written by an active community

Why Python?

- It is free! No licence costs
- Runs on all platforms (Mac, Windows, Linux)
- Because of it’s ease of programming (e.g no need to worry about memory allocation), Python minimises development effort
- A huge number of libraries, written by an active community
- Python can “glue” together functions written in C/C++ and Fortran to speed things up (we can also call R and MATLAB functions)

Why Python?

- It is free! No licence costs
- Runs on all platforms (Mac, Windows, Linux)
- Because of it's ease of programming (e.g no need to worry about memory allocation), Python minimises development effort
- A huge number of libraries, written by an active community
- Python can "glue" together functions written in C/C++ and Fortran to speed things up (we can also call R and MATLAB functions)
- Compared to other high-level scientific languages such as MATLAB and R, Python offers a much wider range of additional functionality (e.g web and GUI development)

Horses for courses

- Python is becoming the de facto standard for exploratory and interactive scientific research

  BUT

Horses for courses

- Python is becoming the de facto standard for exploratory and interactive scientific research

BUT

- Python is no programming silver bullet

Horses for courses

- Python is becoming the de facto standard for exploratory and interactive scientific research

  BUT

- Python is no programming silver bullet
- Your application will ultimately dictate the tool (and a mixture of more than one language **is** ok). For example:

Horses for courses

- Python is becoming the de facto standard for exploratory and interactive scientific research

  BUT

- Python is no programming silver bullet
- Your application will ultimately dictate the tool (and a mixture of more than one language **is** ok). For example:
  - MATLAB excels at interfacing with hardware, e.g generating hardware description language (HDL) code to configure an integrated circuit board or connecting to a data acquisition card

Horses for courses

- Python is becoming the de facto standard for exploratory and interactive scientific research

  BUT

- Python is no programming silver bullet
- Your application will ultimately dictate the tool (and a mixture of more than one language **is** ok). For example:
  - MATLAB excels at interfacing with hardware, e.g generating hardware description language (HDL) code to configure an integrated circuit board or connecting to a data acquisition card
  - R is great for data wrangling and visualisation, and statistical modelling

Horses for courses

- Python is becoming the de facto standard for exploratory and interactive scientific research

  BUT

- Python is no programming silver bullet
- Your application will ultimately dictate the tool (and a mixture of more than one language **is** ok). For example:
  - MATLAB excels at interfacing with hardware, e.g generating hardware description language (HDL) code to configure an integrated circuit board or connecting to a data acquisition card
  - R is great for data wrangling and visualisation, and statistical modelling
  - C achieves the fastest runtimes (no wonder why Windows, Mac OS X, Linux have been coded in C (or flavors thereof), but coding simple things more difficult

Why do *you* want to learn Python?

Some reasons:

Why do *you* want to learn Python?

Some reasons:

- Python has a simple syntax and is widely used; ideal language for beginners

Why do *you* want to learn Python?

Some reasons:

- Python has a simple syntax and is widely used; ideal language for beginners
- Development time is much quicker than for compiled languages like C or Java

Why do *you* want to learn Python?

Some reasons:

- Python has a simple syntax and is widely used; ideal language for beginners
- Development time is much quicker than for compiled languages like C or Java
- Broad uptake: Python (and R) have replaced Perl as the key programming language in Bioinformatics

Why do *you* want to learn Python?

Some reasons:

- Python has a simple syntax and is widely used; ideal language for beginners
- Development time is much quicker than for compiled languages like C or Java
- Broad uptake: Python (and R) have replaced Perl as the key programming language in Bioinformatics
- Same for spatial applications: major GIS applications like ArcGIS use Python as their main scripting language
- Language of choice for machine learning (Tensorflow)

Why do *you* want to learn Python?

Some reasons:

- Python has a simple syntax and is widely used; ideal language for beginners
- Development time is much quicker than for compiled languages like C or Java
- Broad uptake: Python (and R) have replaced Perl as the key programming language in Bioinformatics
- Same for spatial applications: major GIS applications like ArcGIS use Python as their main scripting language
- Language of choice for machine learning (Tensorflow)

Python version 2 vs 3

- Many systems (e.g., Mac OS X) still use Python 2 as the default
- Python 3 differs in various ways from Python 2
- Often, Python 3 code cannot be run using a Python 2 interpreter and vice versa
- Python 2 is a legacy version and will ultimately be replaced by Python 3
- **Current course will focus on Python 3**

Different Python distributions

Various distributions of Python available:

Different Python distributions

Various distributions of Python available:

- Official version from python.org
    - Caveat: libraries and other tools need to be installed separately via `pip`

Different Python distributions

Various distributions of Python available:

- Official version from python.org
    - Caveat: libraries and other tools need to be installed separately via `pip`
- Enthought Canopy: Python, libraries & tools in single installer

Different Python distributions

Various distributions of Python available:

- Official version from python.org
  - Caveat: libraries and other tools need to be installed separately via `pip`
- Enthought Canopy: Python, libraries & tools in single installer
- Anaconda: Python, libraries & tools in single installer: **used in this course**

Different Python distributions

Various distributions of Python available:

- Official version from python.org
  - Caveat: libraries and other tools need to be installed separately via `pip`
- Enthought Canopy: Python, libraries & tools in single installer
- Anaconda: Python, libraries & tools in single installer: **used in this course**

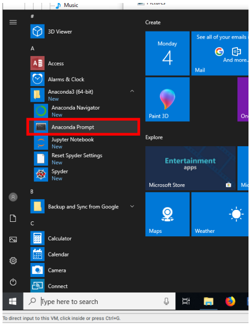
Testing small bits of Python code using the IDLE

IDLE: Integrated Development and Learning Environment

Testing small bits of Python code using the IDLE

IDLE: Integrated Development and Learning Environment

- Windows: Start Menu > Anaconda3 > Anaconda Prompt

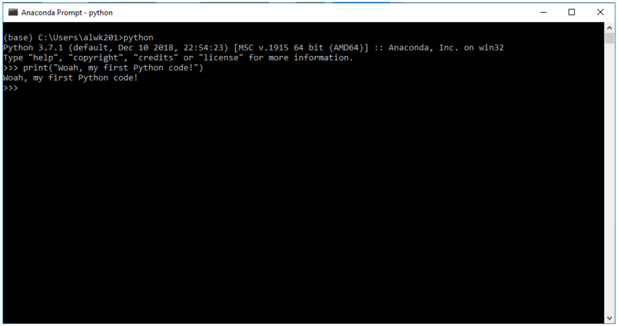


Testing small bits of Python code using the IDLE

- In the command-line prompt that appears, type `python`:

Testing small bits of Python code using the IDLE

- In the command-line prompt that appears, type `python`:
- You can type Python commands after the >>> mark:



Anaconda Prompt - python

```
(base) C:\Users\alwk201>python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Woah, my first Python code!")
Woah, my first Python code!
>>>
```

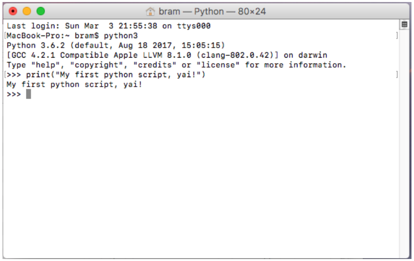
- For example, type `print("Any text you like")`

The IDLE interpreter on a Mac

- In Finder, go to Applications $>$ Utilities $>$ Terminal

The IDLE interpreter on a Mac

- In Finder, go to Applications > Utilities > Terminal
- Type `python3` (not `python`!) to invoke the IDLE



```
Last login: Sun Mar  3 21:55:38 on ttys000
MacBook-Pro:~ bram$ python3
Python 3.6.2 (default, Aug 18 2017, 15:05:15)
[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.42)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("My first python script, yai!")
My first python script, yai!
>>> 
```

- IDLE on Linux: open a terminal and type `python3`

IDLE: finding out how things work

- The IDLE prints the value of anything you type back to you

IDLE: finding out how things work

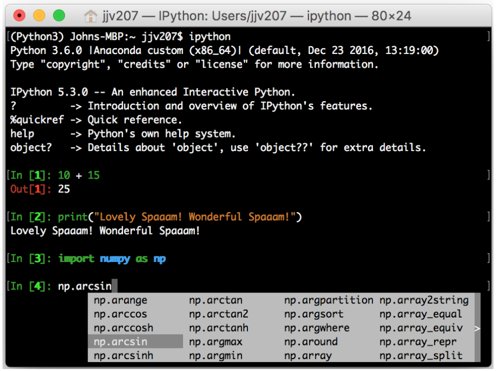
- The IDLE prints the value of anything you type back to you
- This makes the IDLE a great tool to test how commands work

IDLE: finding out how things work

- The IDLE prints the value of anything you type back to you
- This makes the IDLE a great tool to test how commands work
- Not useful for code that spans multiple lines

Executing Python code: IPython interpreter

- IPython is an interactive shell (similar to R Console), adding “frills” to the vanilla IDLE interpreter, such as:
  - syntax highlighting (making it easier to read code)
  - tab auto-completion (minimises typeos and lists available functions)



```
(Python3) Johns-MBP:~ jjv207$ ipython
Python 3.6.0 |Anaconda custom (x86_64)| (default, Dec 23 2016, 13:19:00)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: 10 + 15
Out[1]: 25

In [2]: print("Lovely Spaaam! Wonderful Spaaam!")
Lovely Spaaam! Wonderful Spaaam!

In [3]: import numpy as np

In [4]: np.arcsin
              np.arange       np.arctan       np.argpartition np.array2string
              np.arccos       np.arctan2      np.argsort      np.array_equal
              np.arccosh      np.arctanh      np.argwhere     np.array_equiv
              np.arcsin       np.argmax       np.around       np.array_repr
              np.arcsinh      np.argmin       np.array        np.array_split
```

Executing Python code: Spyder IDE

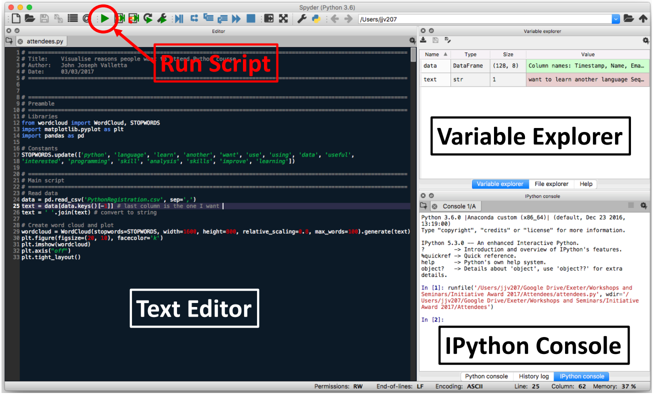
- Windows: Start Menu $>$ Anaconda3 $>$ Spyder

Executing Python code: Spyder IDE

- Windows: Start Menu $>$ Anaconda3 $>$ Spyder
- Mac: Applications $>$ Spyder

Executing Python code: Spyder IDE

- Spyder is an integrated development environment (IDE) for scientific computing, akin to RStudio and MATLAB
- One place to write, execute and debug code, and explore variables



```python
# ================================================================================
# Title:    Visualise reasons people want to attend Python course
# Author:   John Joseph Valletta
# Date:     03/03/2017
# ================================================================================


# ================================================================================
# Preamble
# ================================================================================
# Libraries
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

# Constants
STOPWORDS.update(['python', 'language', 'learn', 'another', 'want', 'use', 'using', 'data', 'useful',
'interested', 'programming', 'skill', 'analysis', 'skills', 'improve', 'learning'])

# ================================================================================
# Main script
# ================================================================================
# Read data
data = pd.read_csv('PythonRegistration.csv', sep=',')
text = data[data.keys()[-1]] # last column is the one I want
text = ' '.join(text) # convert to string

# Create word cloud and plot
wordcloud = WordCloud(stopwords=STOPWORDS, width=1600, height=800, relative_scaling=0.8, max_words=100).generate(text)
plt.figure(figsize=(20, 10), facecolor='k')
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout()
```

Name	Type	Size	Value
data	DataFrame	(128, 8)	Column names: Timestamp, Name, Ema…
text	str	1	want to learn another language Seq…

```
Python 3.6.0 |Anaconda custom (x86_64)| (default, Dec 23 2016,
13:19:00)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra
details.

In [1]: runfile('/Users/jjv207/Google Drive/Exeter/Workshops and
Seminars/Initiative Award 2017/Attendees/attendees.py', wdir='/
Users/jjv207/Google Drive/Exeter/Workshops and Seminars/Initiative
Award 2017/Attendees')

In [2]:
```

Running standalone Python scripts without the IDE

- Windows: don’t bother, use the Spyder IDE

Running standalone Python scripts without the IDE

- Windows: don’t bother, use the Spyder IDE
- Mac/Linux:
  - Write your code in text file, say `my_script.py`

Running standalone Python scripts without the IDE

- Windows: don't bother, use the Spyder IDE
- Mac/Linux:
  - Write your code in text file, say `my_script.py`
  - In a terminal, run:

```
python3 my_script.py
```