Introduction to Python

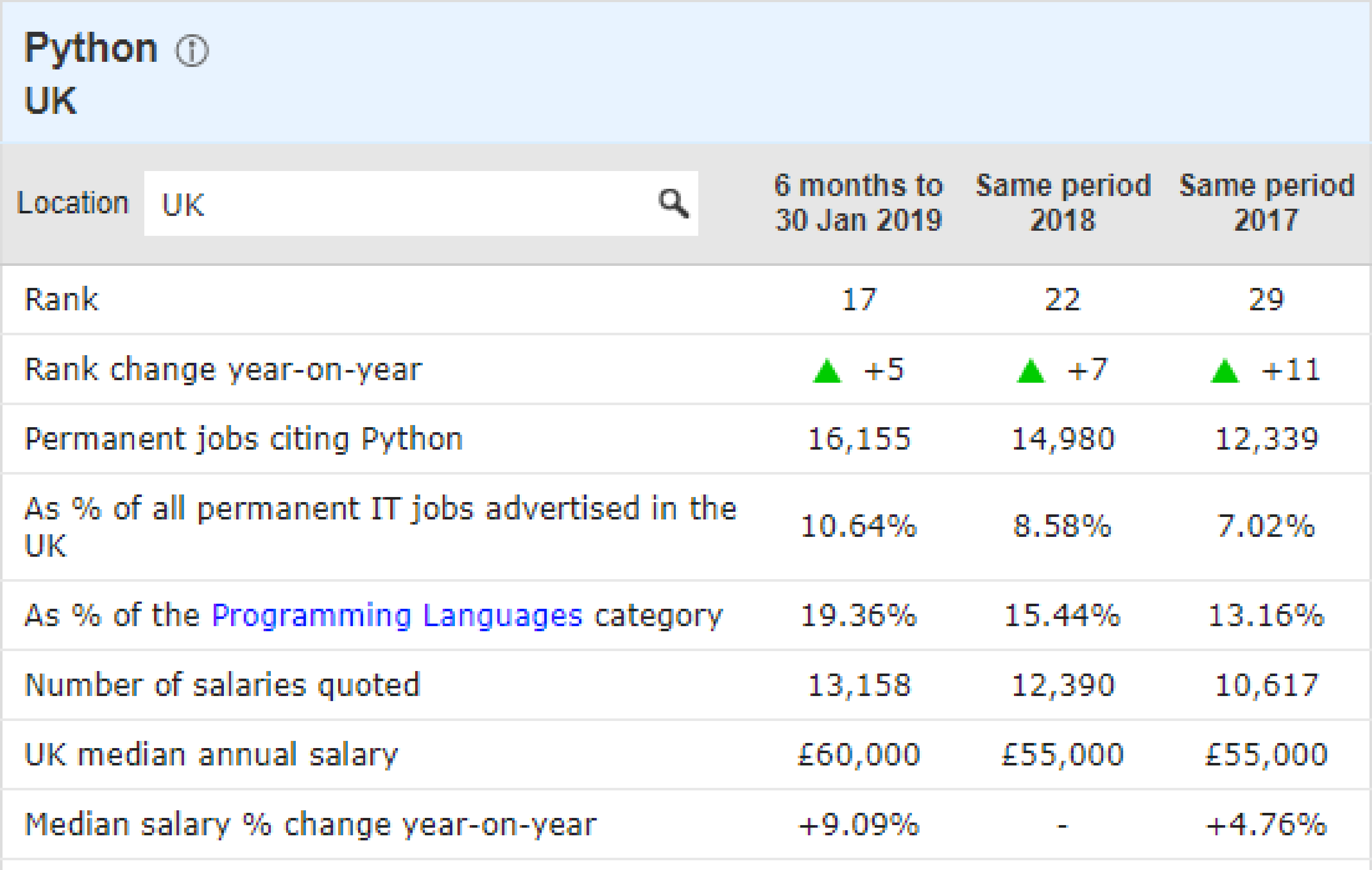
Content

- Why use Python?
- How to install Python
- Packages
- Python IDEs
- Python syntax

Why would you use Python?

- Completely open-source
- Relatively easy to learn
- Many packages for science and data analysis
- Used for things outside academia
  - Django – a web framework in Python

For those considering leaving academia



Python ⓘ

UK

Location UK	6 months to 30 Jan 2019	Same period 2018	Same period 2017
Rank	17	22	29
Rank change year-on-year	▲ +5	▲ +7	▲ +11
Permanent jobs citing Python	16,155	14,980	12,339
As % of all permanent IT jobs advertised in the UK	10.64%	8.58%	7.02%
As % of the Programming Languages category	19.36%	15.44%	13.16%
Number of salaries quoted	13,158	12,390	10,617
UK median annual salary	£60,000	£55,000	£55,000
Median salary % change year-on-year	+9.09%	-	+4.76%

Why avoid Python?

- Can be a slow language (lot of overhead)
- Lenient syntax, making it error-prone
- Memory allocated and is thus limited

Installing Python

- If you are coding in 2019, install Python 3!
    - Install Python
- Python integrated in Anaconda
    - Will give you most scientific packages
    - Install Anaconda

- Link to installation guide

Packages

- Collection of Python module
- Python module contains functions
  - May be written in Python or C
  - May be built-in or imported

Packages for science

- NumPy – arrays, linear algebra, Fourier transforms
- pandas – data structures, time series analysis
- Matplotlib and Seaborn – data visualization
- Scikit-learn – modelling and machine learning
- TensorFlow – deep neural networks
- OpenCV – computer vision

Jupyter

- Notebooks to write little bits of code
- Modular running of code (like Matlab)
- Inline visualization

IDEs

- IDE = integrated development environment
- Spyder – Console, script and variable view
  - Those used to Matlab will like this
- Pycharm – Powerful IDE with code completion
  - Part of range of JetBrains IDEs
- Eclipse – multifunctional IDE
  - Supports a large range of languages

Python syntax is unique

- No curly brackets
- No semicolons (unless you want to write two things on one line, which you shouldn’t)
- Indents matter

Python syntax example

```
int x = 1;
if( x > 0 ){
   printf('x is larger than 0');
}
```

```
x = 1
if x > 0:
    print(“x is larger than 0”)
```

- No curly brackets
- No semicolons
- Indents matter

- Implicit typing

Python’s superpower: list comprehension

- Shorten for-loops

```
[expression for item in list if conditional]
```

is equivalent to

```
for item in list:
    if conditional:
        expression
```

Example: I want to divide all even numbers in a list by 2

```
for item in list:
    if conditional:
        expression
```

```
for number in myList:
    if (number % 2) == 0:
        number = number/2
```

Example: I want to divide all even numbers in a list by 2

```
for number in myList:
    if (number % 2) == 0:
        x = number/2
```

```
[number/2 for number in myList if (number % 2) == 0]
```

Getting started with Python

- Tutorials for if you've never programmed before
- Tutorials if you can program in another language

Code Club

- Trying to follow a red thread
- Most lectures will be based on Python
- We’ll roughly follow https://mlcourse.ai/
- Should still be able to follow lectures if you miss one

25 February – 18:30
LGBT in STEM Wikipedia Edit-a-thon

Next week: Thomas Nowotny on GPUs