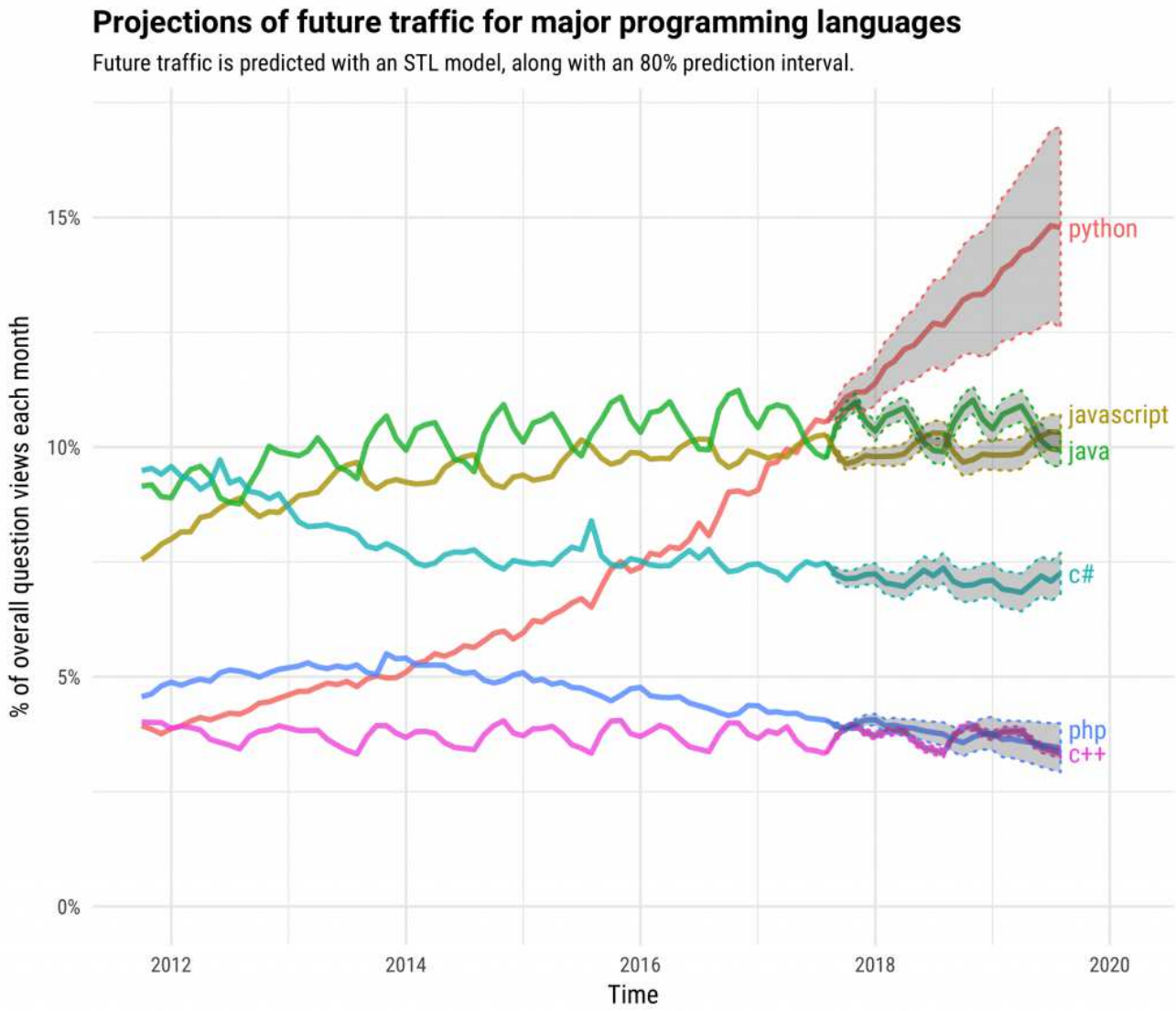
GSE Regional Conference 2018 - 25 April 2018

Why Everybody Loves Python

Arnout Veugelen - ABIS Training & Consulting

Track: Innovation as a Competitive Advantage

The Incredible Growth of Python



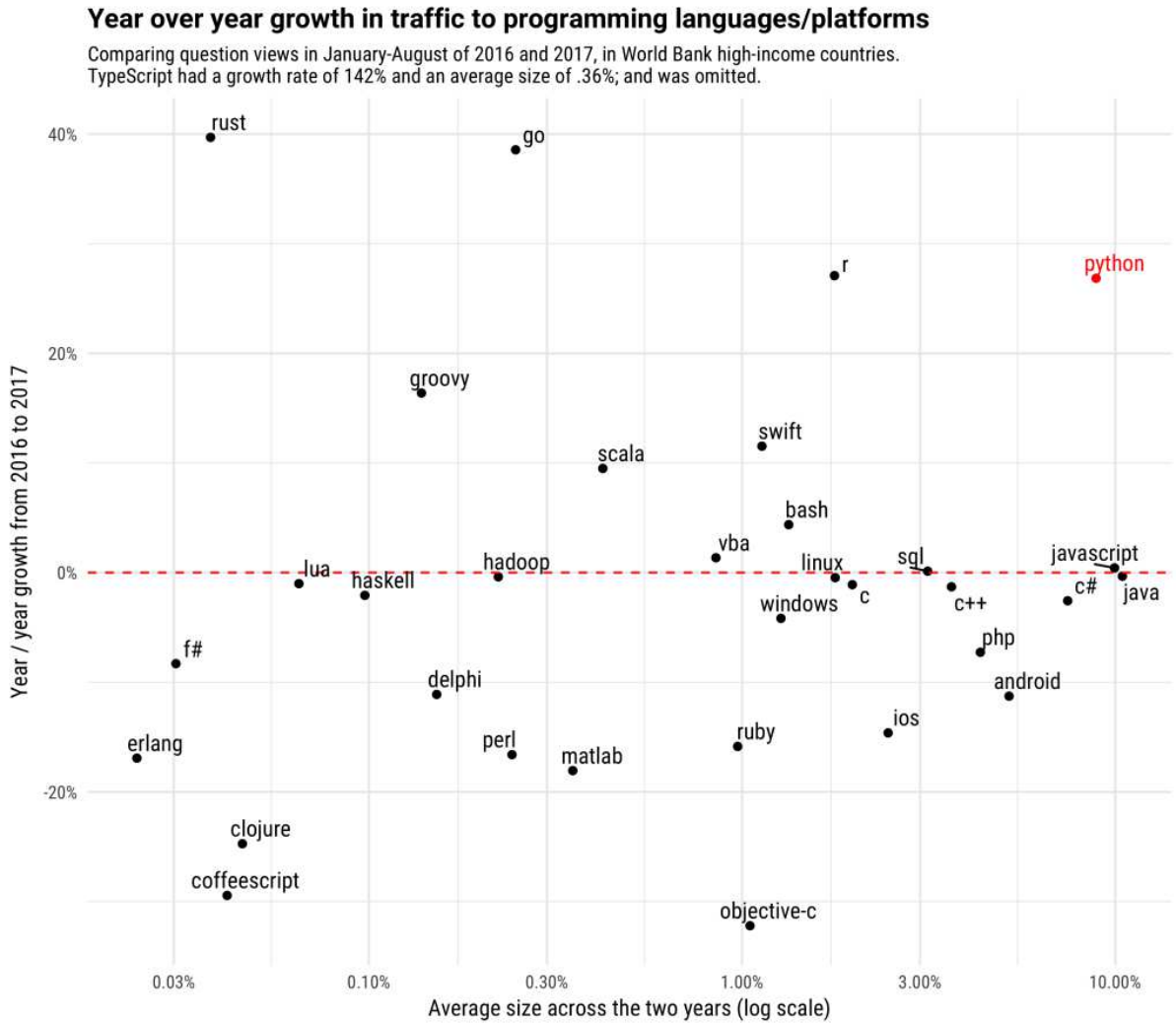
source: Stack Overflow - The Incredible Growth of Python (https://stackoverflow.blog/2017/09/06/incredible-growth-python/)

Ask a Famous Historian

- Benevolent Dictator For Life: Guido Van Rossum
- Created Python in 1989
- Initial goal: high-level language for system administration
- Named after Monty Python's Flying Circus

source: Doc Searls - via wikimedia (https://commons.wikimedia.org/wiki/File:Guido_van_Rossum_OSCON_2006.jpg)

The Incredible Growth of Python



source: Stack Overflow - The Incredible Growth of Python (https://stackoverflow.blog/2017/09/06/incredible-growth-python/)

The Incredible Growth of Python

Top Training Technologies at ABIS in 2017

1. Java
2. SQL
3. Linux / UNIX
4. Oracle
5. Python
6. Db2 for z/OS

(Based on student-days)

Why is Python so popular?

=> Python puts the developer first!

Comfortable syntax

In [2]:
```
print('Hello GSE')
```

```
Hello GSE
```

Versus Java:

```
class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello GSE!");
  }
}
```

Comfortable syntax

Versus COBOL:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION.
MAIN SECTION.
DISPLAY "Hello GSE!"
STOP RUN.
```

Versus REXX:

```
say 'Hello GSE!'
```

Comfortable syntax

In [3]:
```
values = [1, 17, 313, 28, 99, 12, 4238]

for item in values:
    if item % 3 == 0:
        print(item)
    else:
        print('spam!')
```

```
spam!
spam!
spam!
spam!
99
12
spam!
```

Comfortable syntax

Versus C:

```
#include <stdio.h>
int main() {
  int values[] = {0, 1, 17, 313, 28, 99, 12, 4238, EOF};
  int *item;
  for (item = values; *item!=EOF; ++item)
    if (*item % 3 == 0)
      printf("%d\n", *item);
    else
      printf("spam!\n");
  return 0;
}
```

=> Python's syntax is **concise**

Comfortable syntax

In [4]:
```
slogan = 'We are now no longer the knights who say NI'

for word in slogan.split():
    if len(word) % 2 == 0:
        print(word)
```

```
We
no
longer
NI
```

Versus Perl:

```
$slogan = "We are now no longer the knights who say NI";
print join "\n", grep { length($_) % 2 == 0 } split / /,$slogan;
```

=> Python's syntax is **concise**, yet **legible**

Comfortable syntax

- Dynamic typing
- No braces needed for code blocks
- No parantheses needed around a test
- No semicolons needed to indicate statements
- Automatic iteration over any 'collection' (lists, strings, files ...), no need for counters or EOF-pointers
- Mandatory (!) indentation

```
import this
```

The Zen of Python - by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

...

Readability counts.

...

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Several programming paradigms are supported:

- Procedural
- Object-Oriented
- Functional

In [5]:
```
slogan = 'We are now no longer the knights who say NI'
[word for word in slogan.split() if len(word) % 2 == 0]
```

Out[5]:
```
['We', 'no', 'longer', 'NI']
```

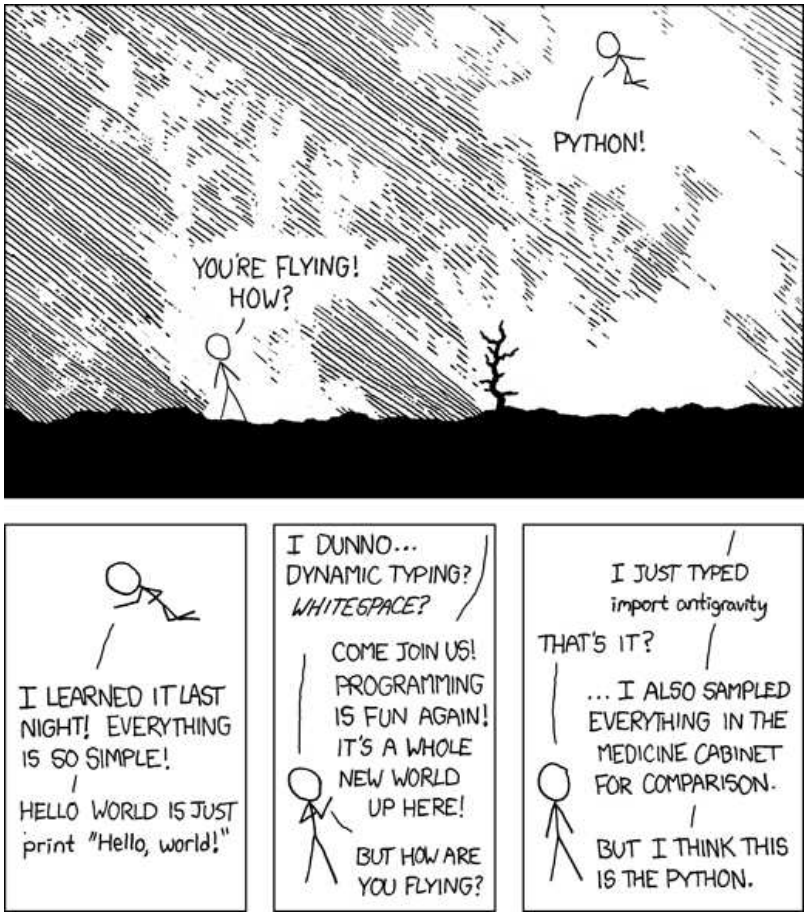
Portable across all major platforms:

- Windows
- Linux
- Mac
- iOS
- Android
- z/OS
- ...

Batteries are included

- Built-in object types with dedicated methods: strings, numbers, lists, dictionaries, files ...
- Built-in functions
- The standard library:
  - Included in the standard installiation
  - ~250 modules with a myriad of facilities:
    - Access to system functionality
    - Standardized solutions for various everyday problems
- An ecosystem of readily available third party packages for virtually anything (www.pypi.org)

```
import antigravity
```



source: xkcd (https://xkcd.com/353/)

Why is Python so popular?

=> Python puts the developer first!

- Comfortable syntax
- Several programming paradigms are supported
- Portable across all major platforms
- Batteries are included

Bonus points:

- Open Source
- Free!

Example: a simple web scraper

Goal: fetch all soccer results of this year's Belgian first division from www.sport.be

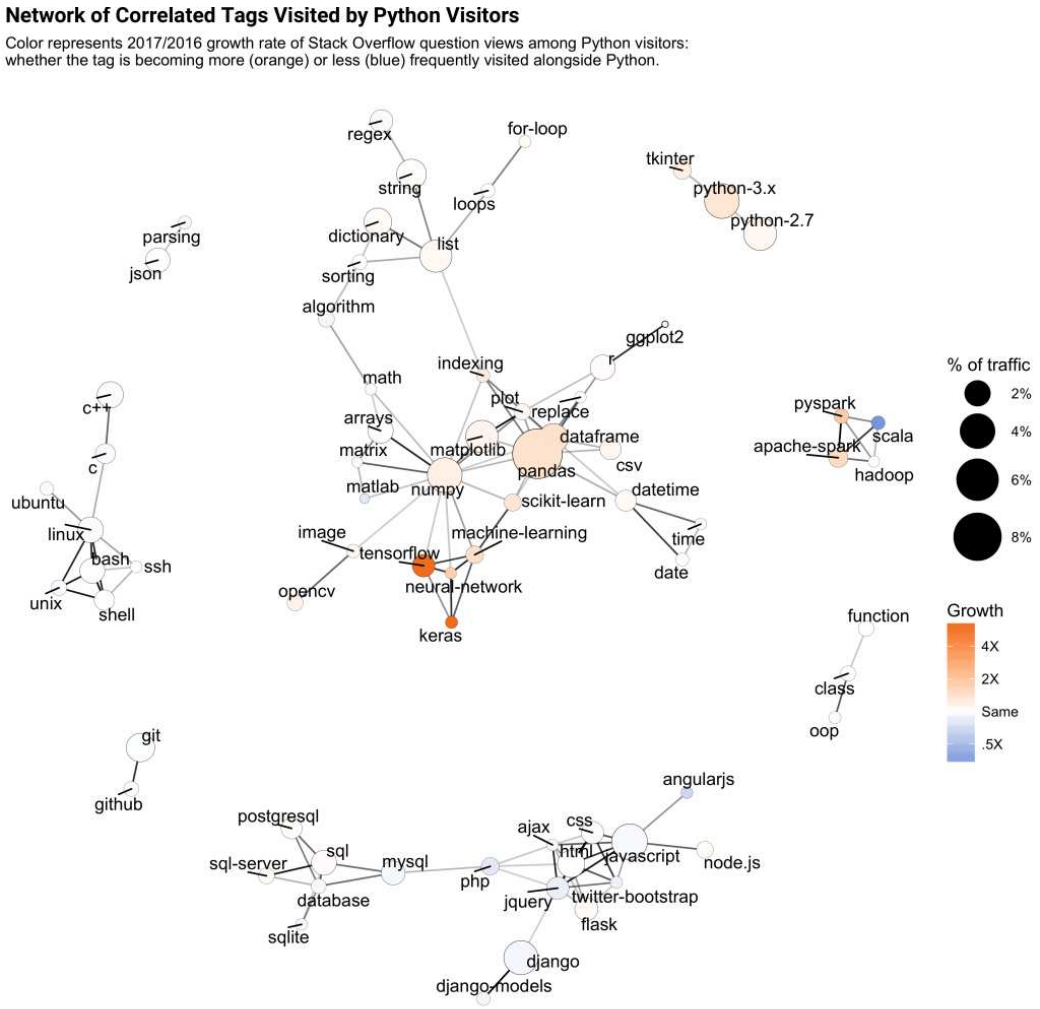
http://www.sport.be/nl/jupilerproleague/fixture.html?comp=38
(http://www.sport.be/nl/jupilerproleague/fixture.html?comp=38)

Example: a simple web scraper

In [6]:
```
from urllib.request import urlopen
from bs4 import BeautifulSoup

base_url = 'http://www.sport.be/nl/jupilerproleague/fixture.html?comp=38&period='

with open('soccerscores.csv', 'w') as outputfile:
    print('period', end=' ')
    for period in range(1,31):
        print(period, end=' ')
        url = base_url + str(period)
        html = urlopen(url)
        soup = BeautifulSoup(html, 'html5lib')
        table = soup.find(
                    name='table',
                    attrs={'class': 'table table-vertical-align table-calendar'})

        for row in table.find_all('tr')[1:]:
            date = row.find(name='span', attrs={'class': 'm_date'}).get_text()
            teams = row.find_all(name='div', attrs={'class': 'm_team'})
            team1 = teams[1].get_text()
            team2 = teams[3].get_text()
            score = row.find(name='div', attrs={'class': 'm_score'}).get_text()
            print(date, team1, team2, score, sep=',', file=outputfile)
```

```
period 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30
```

"Great, but how does that explain Python's recent growth?"



source: Stack Overflow - Why Is Python Growing so Quickly (https://stackoverflow.blog/2017/09/14/python-growing-quickly/)

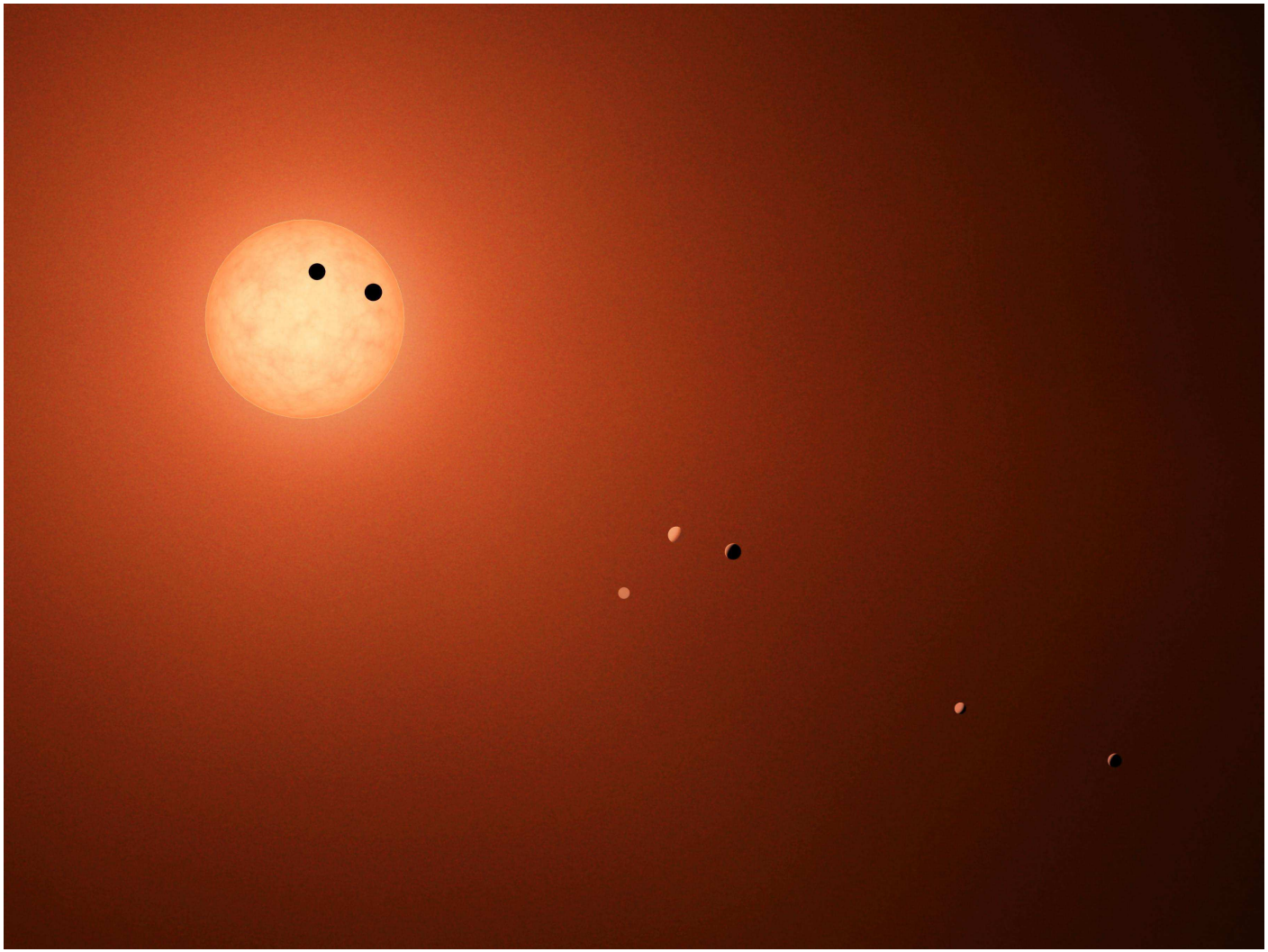
Python is becoming the number one language for data science

Initially, Python is used to glue together various tools used by scientists.

Gradually a rich ecosystem of Python packages was developed to replace non-Python tools.

Now, Python is at the forefront of ML and AI.

Python is becoming the number one language for data science



source: NASA (https://photojournal.jpl.nasa.gov/catalog/PIA21429)

"But isn't Python too slow for this kind of heavy data computing?"

In [8]:
```
L = [2, 'a', [23.5, 'eggs'], {'food': 'spam'}]
print(L[3]['food'])
```

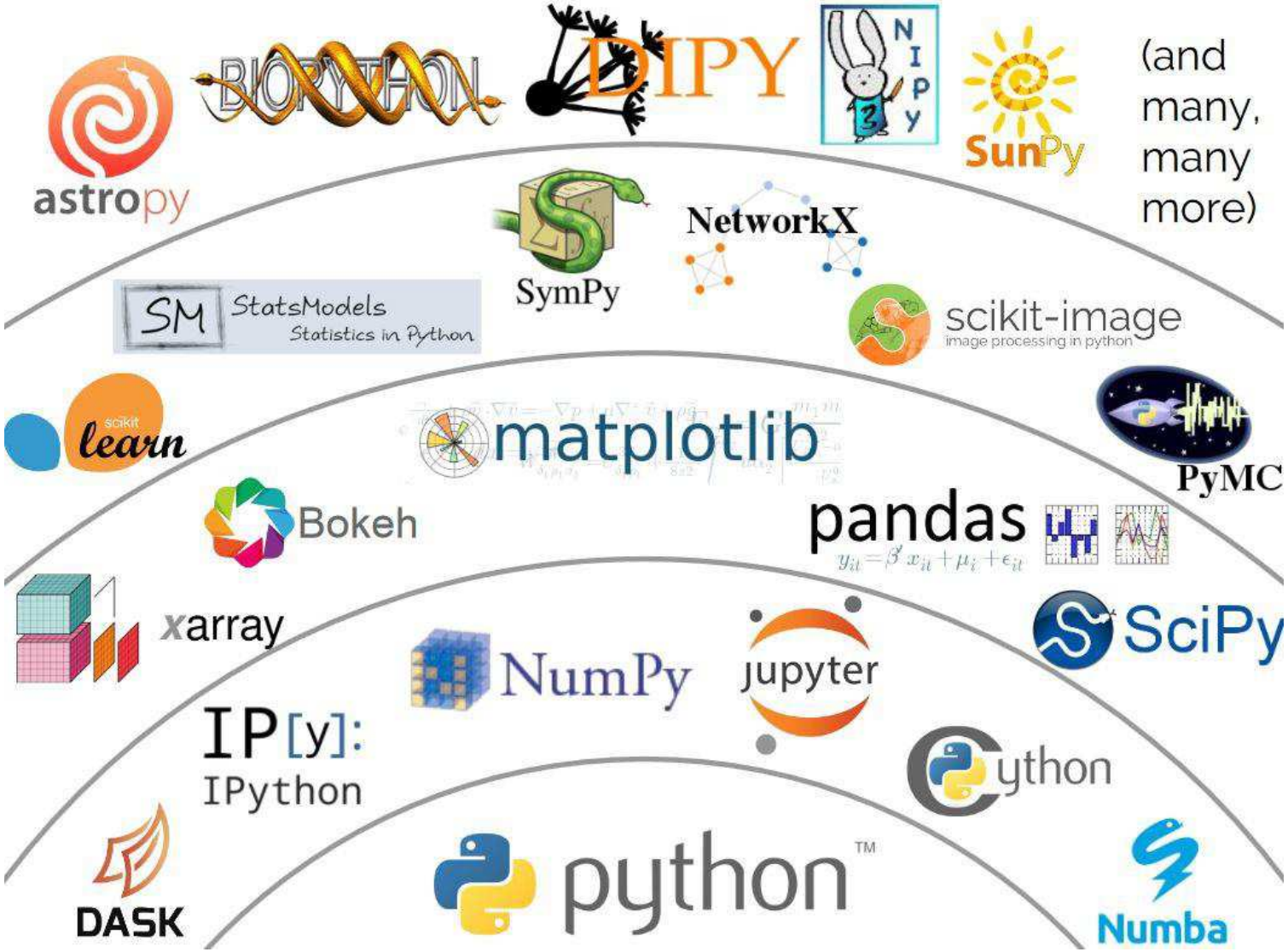
```
spam
```

Python's flexibility comes at a price in terms of execution speed.

- Guido Van Rossum: *"The user's time matters a lot more than the computer's time."*

- Mitigating the speed problem:
  - speeding up standard Python: multithreading, multiprocessing, asyncio
  - (semi-) compiled alternative implementations (Jython, PyPy)
  - **Use Python as front-end and main program flow, but dispatch some of the heavy lifting to other languages**

The Scientific Python Ecosystem



source: Jake Vanderplas (https://github.com/jakevdp/)

Ask me the questions, bridgekeeper.

And now for something completely different.

In [9]:
```
print(spam)
```

```
Check out our Python courses at www.abis.be
contact me: aveugelen@abis.be
```