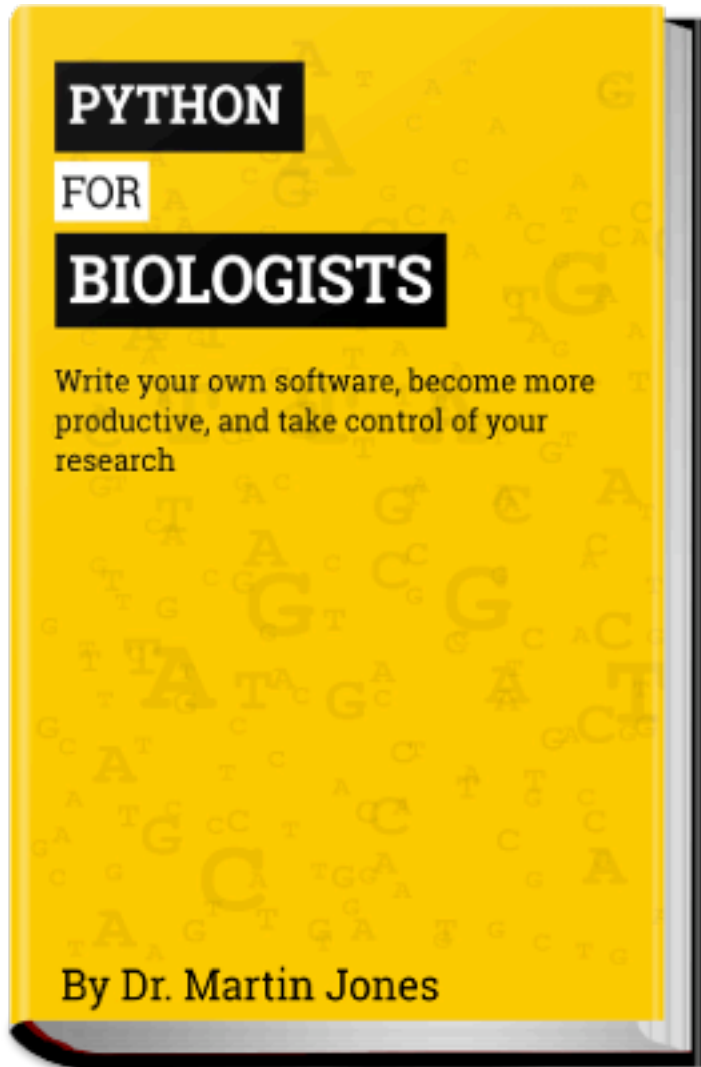




# Book



Much of the material is available for free on the website:

<http://pythonforbiologists.com/>

\$26 on amazon

All examples are pertinent to biologists, most are directly based on sequence data.

# Why is bioinformatics field moving from perl to python?

The switch happened between 2008 and 2010

What do biologists need?

- A tool to get science done
- Low barrier to entry and efficiency in writing code

How does python meet those needs?

- Emphasis on readability and simplicity
- Possibly easier to learn than perl\*
- Excellent scientific libraries of code to leverage
- Widely used
- “Trendy” factor

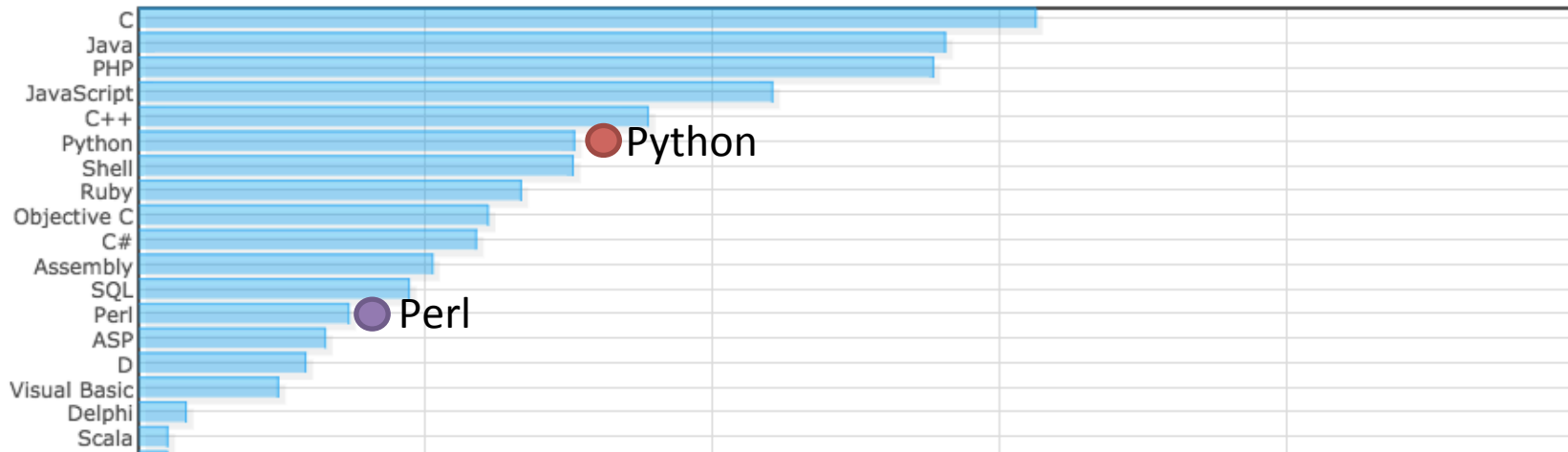
Some good news:  
If you learn python the leap to perl is very short.  
You'll probably already be able to read perl.

An Empirical Investigation into Programming Language Syntax, 2013, Stefik and Seibert, ACM Transactions on Computing Education

# Popularity according to Langpop.com, 2014

## Normalized Comparison

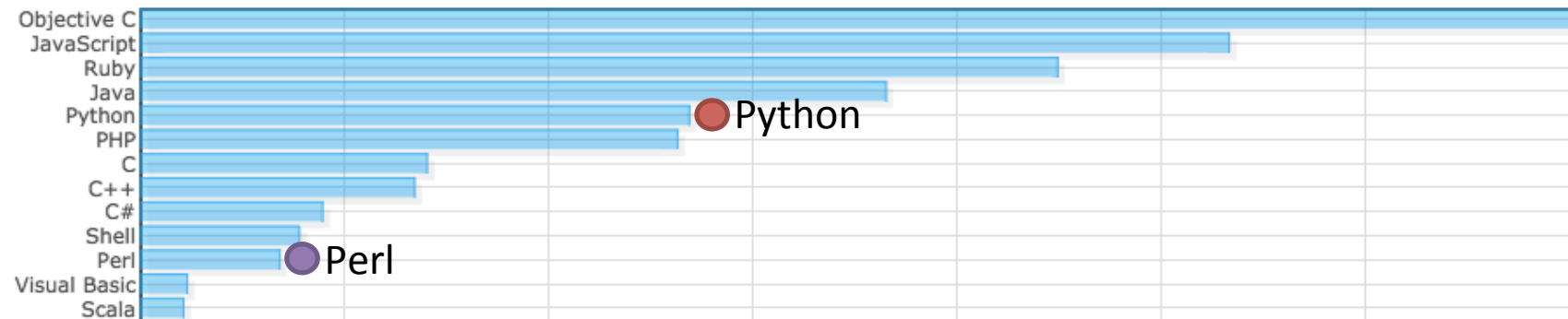
This is a chart showing combined results from all data sets, listed individually below.



## Github

Data from Github was obtained using the API to search here: <http://developer.github.com/v3/search/>

There is one downside to using this data: it favors open source projects with code that is visible on the internet.





- The debate is now python vs R
- Fortunately learning one language makes it much easier to learn a second language
- You start by learning to think like programmer



R:

- Focused on statistics and graphical models
- Preferred by data scientists, statisticians, quants
- Has always been centered in academics and research, less in industry
- See “Data Science Wars: R vs Python” in readings

# Python - History

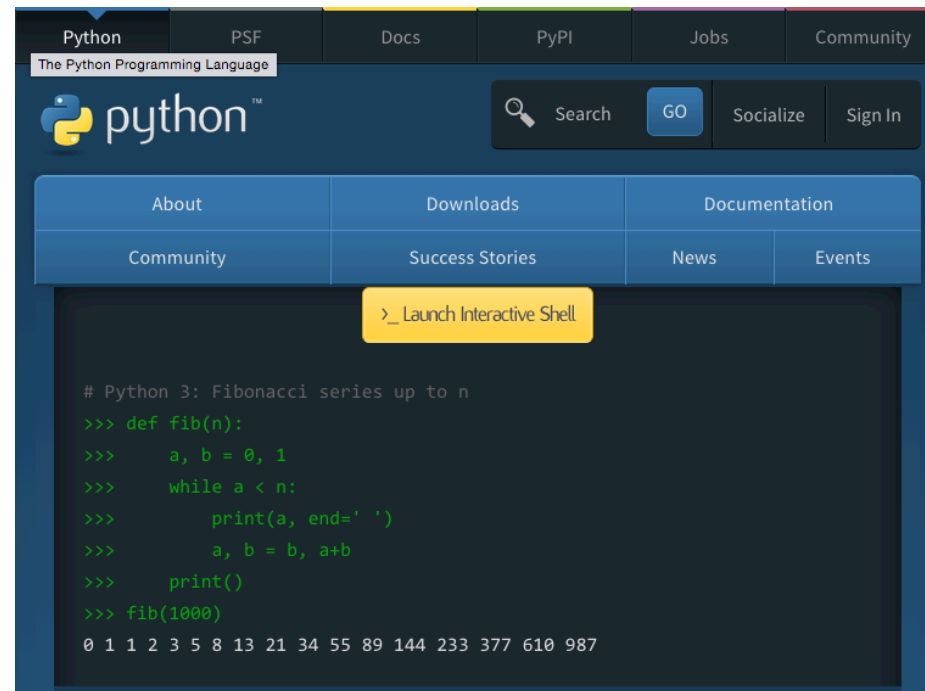
- Developed by the Python Software Foundation
- Created in 1989 by Guido van Rossum
- Named after Monty Python and the Flying Circus
- Design philosophy
  - Code readability
  - Consistency
  - Compact, uncluttered layout
    - English keywords instead of punctuation
    - Whitespace indentation is part of the syntax

# Python - Philosophy

- document "PEP 20 (The Zen of Python)" has 20 aphorisms, of which 19 are written down.
- Including:
  - Beautiful is better than ugly
  - Simple is better than complex
  - Complex is better than complicated
  - Readability counts
  - Special cases aren't special enough to break the rules.
  - If the implementation is hard to explain, it's a bad idea.

# Python – Resources

- Python Software Foundation
  - <http://python.org>
  - Downloads and installation instructions
  - Documentation (in > 70 languages)
  - Python for beginners
- Python Package Index
  - <http://pypi.python.org>
  - official repository for python software packages
  - more than 49,000 packages
  - Codename PyPI





# Python Versions

(aka the giant disaster)

- 2.7
  - The final 2.x version, released 2010 – no new major releases
  - Statement of extended support
  - Status quo
- 3.4
  - 3.0 released in 2008, 3.5 released in 2015
  - Shiny and new
  - Not much backward compatibility => less libraries, less documentation

**3.5.1 Newton Version**

(Other versions available)

# Python Versions

Try to stick with one version (3)

- Averting disaster if you need to use python 2 one day and you do any math at all
- Start all scripts with

```
from __future__ import division
```

- Fixes your division operator (/) to do more sensible things

# Packages

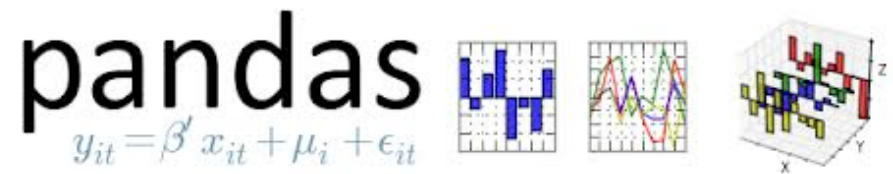
- SciPy
  - ecosystem of open-source software for mathematics, science, and engineering
- NumPy
  - Scientific computing package
  - Large multidimensional arrays
  - linear algebra, Fourier transform, and random number capabilities
- IPython
  - Interactive console
  - Can display inside a web browser



IP[y]:

# Packages

- Pandas
  - Data structures and analysis
  - Dataframe objects
- Biopython
  - [www.biopython.com](http://www.biopython.com)
  - Bioinformatics support
  - Support for 2.7 and 3.3



# Python on Newton

- When you use a new piece of software or language, the shell needs to know
  - Where is the software? (\$PATH variable)
  - Where are the library files for the software? (\$LD\_LIBRARY\_PATH variable)
  - Where are the manual pages? ( \$MAN\_PATH variable)
  - Etc.
- Newton uses modules (see readings)
- This is a convenient way to manage your environment and hides all those path issues
- It also allows different versions of software to be installed and for users to select their preferred version.

