Presented by:

Avirup Guha Neogi

Ashraf Eltaher

Abideen Idowu

b-it Bonn-Aachen International Center for Information Technology

Introduction

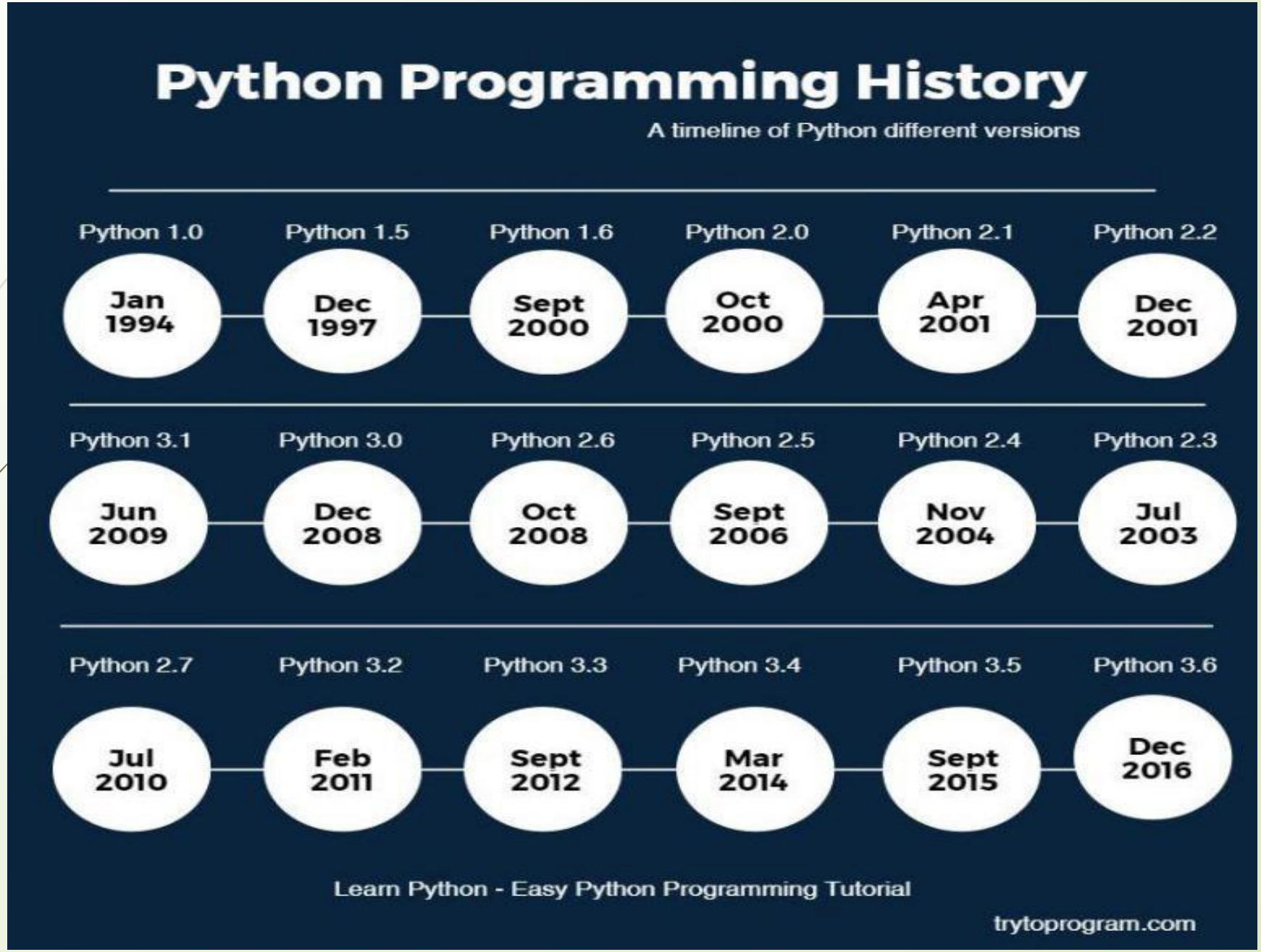
- Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.
- It is used for:
- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Features of Python

- Uses an elegant syntax, making the programs you write easier to read.
- Is an easy-to-use language that makes it simple to get your program working. This makes Python ideal for prototype development and other ad-hoc programming tasks, without compromising maintainability.
- Comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, reading and modifying files.
- Python's interactive mode makes it easy to test short snippets of code. There's also a bundled development environment called IDLE.
- Is easily extended by adding new modules implemented in a compiled language such as C or C++.
- Can also be embedded into an application to provide a programmable interface.
- Runs anywhere, including Mac OS X, Windows, Linux, and Unix, with unofficial builds also available for Android and iOS.
- Is free software in two senses. It doesn't cost anything to download or use Python, or to include it in your application. Python can also be freely modified and re-distributed, because while the language is copyrighted it's available under an open source license.


Python Programming History
A timeline of Python different versions
Python 1.0
Jan 1994
Python 1.5
Dec 1997
Python 1.6
Sept 2000
Python 2.0
Oct 2000
Python 2.1
Apr 2001
Python 2.2
Dec 2001
Python 3.1
Jun 2009
Python 3.0
Dec 2008
Python 2.6
Oct 2008
Python 2.5
Sept 2006
Python 2.4
Nov 2004
Python 2.3
Jul 2003
Python 2.7
Jul 2010
Python 3.2
Feb 2011
Python 3.3
Sept 2012
Python 3.4
Mar 2014
Python 3.5
Sept 2015
Python 3.6
Dec 2016
Learn Python - Easy Python Programming Tutorial
trytoprogram.com

Advantages of Python Programming Language

- Extensive Libraries
- Extensible
- Embeddable
- Improved Productivity
- IOT Opportunities
- Simple and Easy
- Readable
- Object-Oriented
- Free and Open-Source
- Portable

Disadvantages of Python Programming Language

- Speed Limitations
- Weak in Mobile Computing and Browsers
- Memory Consumption
- Database Access
- Runtime Errors

Some application of Python in Life Sciences

- Automate routine lab calculations with Python
- Find important motifs in genome sequences
- Search for gene promoter sequences
- Rotate a molecular bond
- Use object-oriented programming with Python to model diseases
- Mine interaction network data for patterns
- Create simulations of biochemical switches

Bioinformatics tools developed in Python

- Biopython: set of freely available tools for biological computation
- PyMOL: molecular visualization system
- PyCogent is a software library for genomic biology
- Galaxy: an open, web-based platform for data intensive biomedical research
- pygr: sequence and comparative genomics analyses, even with extremely large multi-genome data sets
- Biskit: facilitates the manipulation and analysis of macromolecular structures, protein complexes, and molecular dynamics trajectories
- Ruffus: a lightweight python module for running computational pipelines
- Pysam: for reading and manipulating Samfiles
- msatcommander: locates microsatellite (SSR, VNTR, &c) repeats within fasta-formatted sequence or consensus files
- glu-genetics: tools to store, clean, and analyze data generated by whole-genome or candidate gene association scans

Bioinformatics tools developed in Python contd…

- PySCeS provides a variety of tools for the analysis of cellular systems
- OpenAlea: odules to analyse, visualize and model the functioning and growth of plant architecture
- ETE assists in the automated manipulation, analysis and visualization of phylogenetic and other type of trees
- bx-python: allows for rapid implementation of genome scale analyses
- RSeQC: comprehensively evaluate high throughput sequence data especially RNA-seq data
- incf-omni: analysis and simulation construction of the nervous system
- genetrack: storing, querying and visualizing genomic interval oriented data
- chimerascan: detection of chimeric transcripts in high-throughput sequencing data

Python libraries and packages for Life Science Informatics

- These are the five most essential Data Science libraries you have to know:
  - Numpy
  - Pandas
  - Matplotlib
  - Scikit-Learn
  - Scipy

Numpy

- It is a Python C extension library for array-oriented computing.
- Helps you to manage multi-dimensional arrays very efficiently.
- NumPy is suited to many applications
  - Image processing
  - Signal processing
  - Linear algebra
  - A plethora of others
- Concept is a crucial part of data science, many other libraries are built on Numpy.

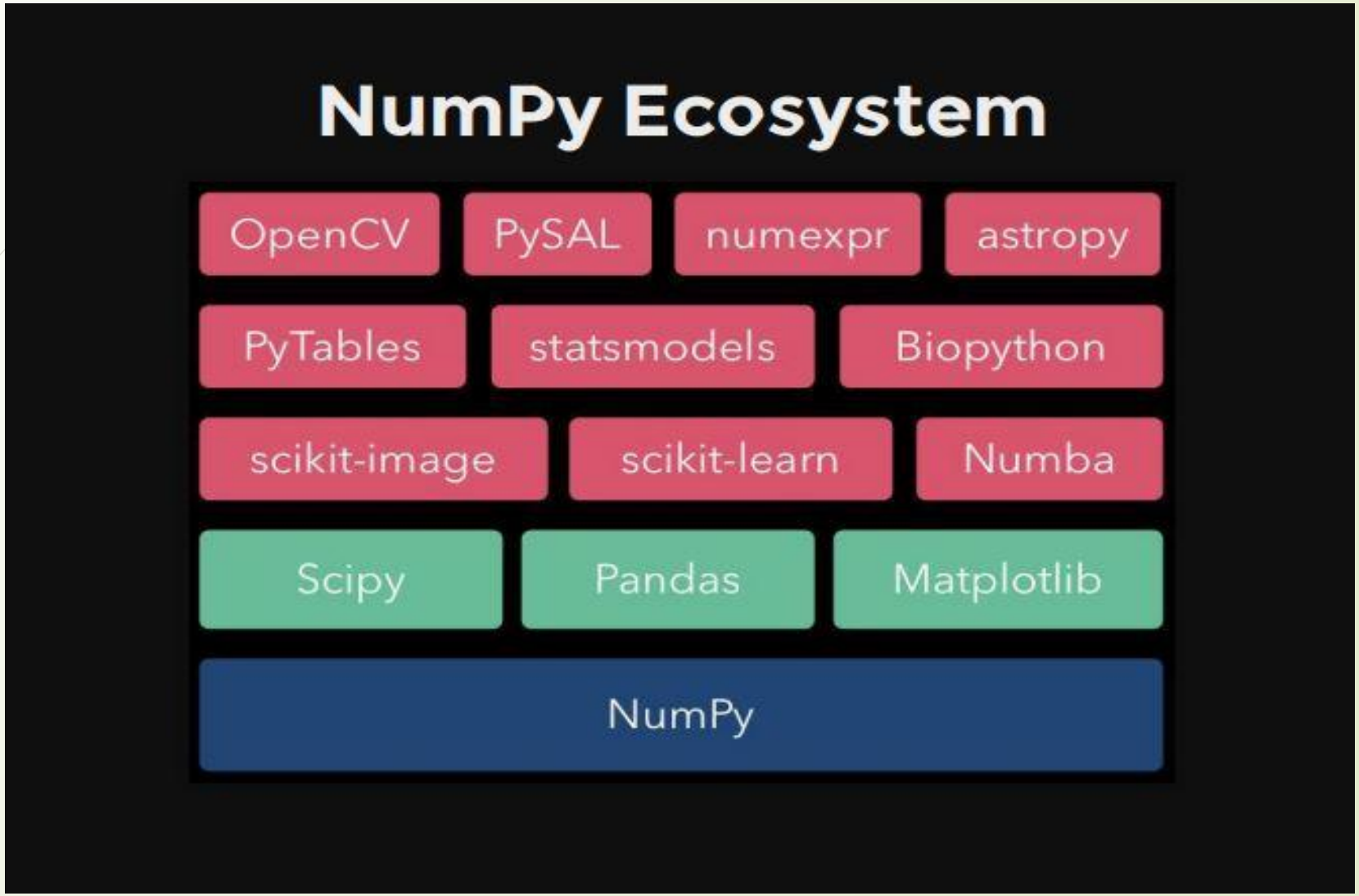


Fig 1: *NumPy as a foundation of the python scientific stack*
https://webvalley.fbk.eu/static/media/uploads/presentations/introductiontonumpy2.pdf

Pandas

- Pandas is an open source, BSD-licensed library
- High-performance, easy-to-use data structures and data analysis tools
- Offers data structures and operations for manipulating numerical tables and time series
- To analyze data, we like to use two-dimensional tables – like in SQL and in Excel. Originally, Python didn't have this feature.
- The "*SQL of Python*".
- Helps us to handle two-dimensional data tables in Python. In many senses it's really similar to SQL, though.

Pandas Contd…

- With pandas, you can load your data into data frames, you can select columns, filter for specific values, group by values, run functions (sum, mean, median, min, max, etc.), merge dataframes and so on.
- You can also create multi-dimensional data-tables.

```
In [6]: super_tree.head()
```

Out[6]:

	day	my_date	user_id	event_type
0	day_1	2017-12-01	1000007	sent_a_super_tree
1	day_1	2017-12-01	1000010	sent_a_super_tree
2	day_1	2017-12-01	1000011	sent_a_super_tree
3	day_1	2017-12-01	1000019	sent_a_super_tree
4	day_1	2017-12-01	1000022	sent_a_super_tree

Fig 2: A pandas dataframe

https://data36.com/python-libraries-packages-data-scientists/

Matplotlib

- Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats
- A plotting library for the Python programming language and its numerical mathematics extension NumPy.
- It provides an object-oriented API for embedding plots into applications using GUI toolkits like Tkinter, wxPython, Qt, or GTK+.
- Matplotlib is the best and most well-known Python data visualization library.

Matplotlib Contd…

- You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc.

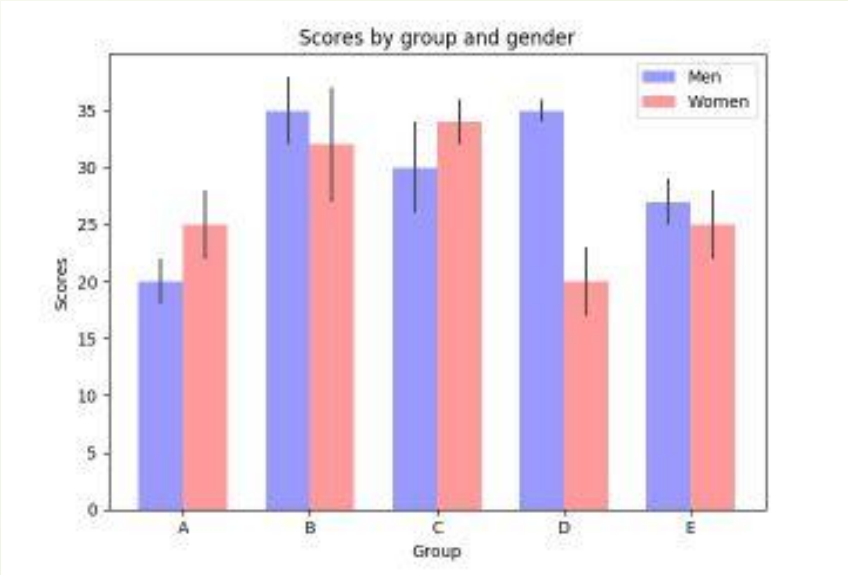


Fig 3: Bar chart Demo

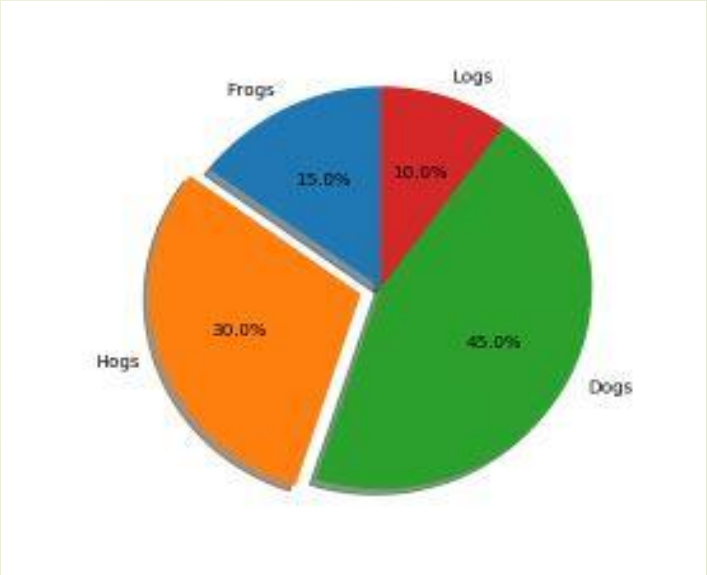


Fig 4: Pie Features

https://matplotlib.org/tutorials/introductory/sample_plots.html

Scikit-Learn

- Simple and efficient tools for data mining and data analysis
- It features various classification, regression, Model selection, preprocessing and clustering algorithms
- Built on NumPy, SciPy, and matplotlib
- Accessible to everybody, and reusable in various contexts

Scipy

- A free and open-source Python library used for scientific computing and technical computing
- Contains modules for optimization, linear algebra, integration, interpolation etc.
- It provides the core mathematical methods to do the complex machine learning processes in Scikit-learn.

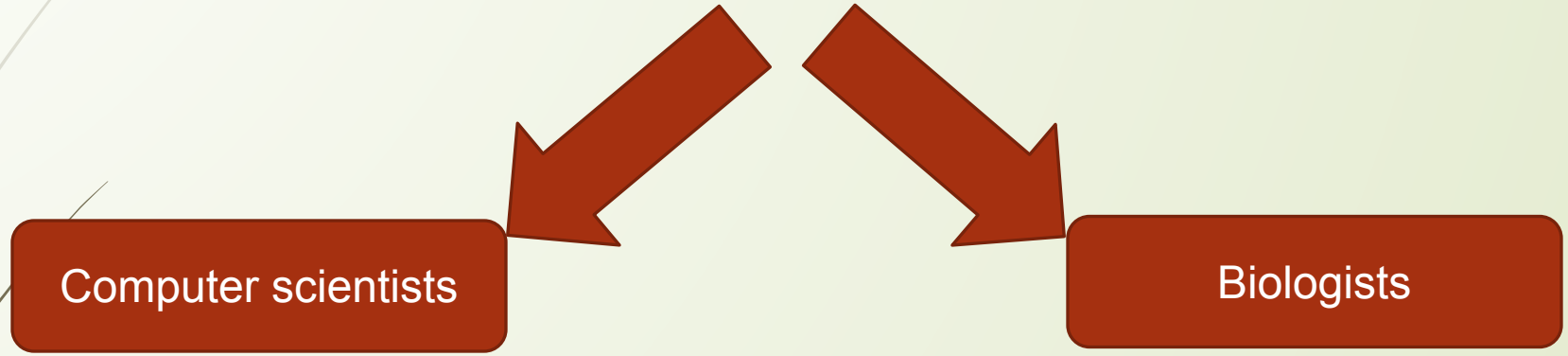
What is Bioinformatics?

Biology goes digital

What are the bioinformatician fields of expertise?

A new species of biologist is beginning to thrive

There are two paths to careers in bioinformatics, both of which require learning a new language.



must become fluent in the life science terminology of genetics, genomics and cellular biology

must pick up skills in data analysis, including statistics, logic and programming

Is this everything?

Absolutely not

The skill set needed by a bioinformatician **continues to evolve**. In the early days of the human genome project, it was sufficient for scientists to find **homologous genes of one organism** in the genome of another.

Now, bioinformaticians routinely compare **multiple genomes**, analyze **regions that don't code for DNA**, and incorporate a host of proteomic information in their analysis.

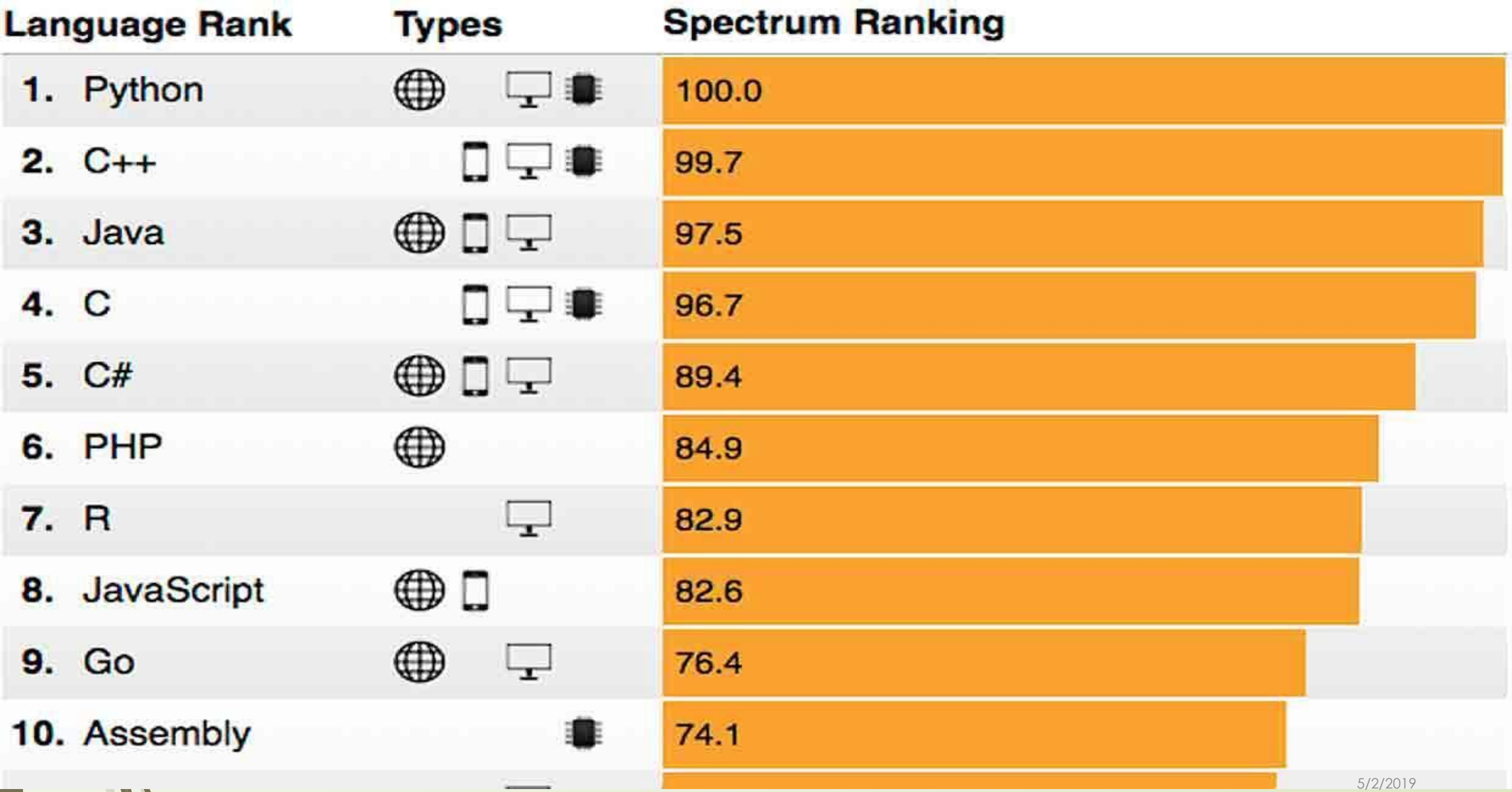
Both the **type** and **amount** of information continues to expand, as biological techniques continue to improve.

To be competent to the dynamic market , He must have the same level of dynamism.

Market data

Now, Let's be more practical!

Where does python stand among other programming languages?



https://hackernoon.com/top-3-programming-language-to-watch-out-in-2019-95995e81ad2b

Why python was chosen by the users?

- It is very handy in certain applications to play with attached hardware via an **interactive prompt** or dynamically reload scripts on the fly.

- Another reason for Python's increasing popularity may be seen in **R's small decline.**
  R is a language specialized for handling statistics and big data. As the interest in large data was owing to their applications in machine learning.
  However, the existence of high-quality Python libraries for both statistics and machine learning may be making **flexible** Python a more attractive jumping-off point than the more specialized R.

The PYPL popularity index

Worldwide, Apr 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	27.35 %	+5.2 %
2		Java	20.64 %	-1.8 %
3		Javascript	8.4 %	-0.2 %
4		C#	7.45 %	-0.5 %
5		PHP	7.18 %	-1.1 %
6		C/C++	6.08 %	-0.3 %
7		R	4.18 %	-0.1 %

created by analyzing how often language tutorials are searched on Google.

http://pypl.github.io/PYPL.html

Is Python the best?

Yes

or

NO

Does the popularity of Python translate into good jobs?

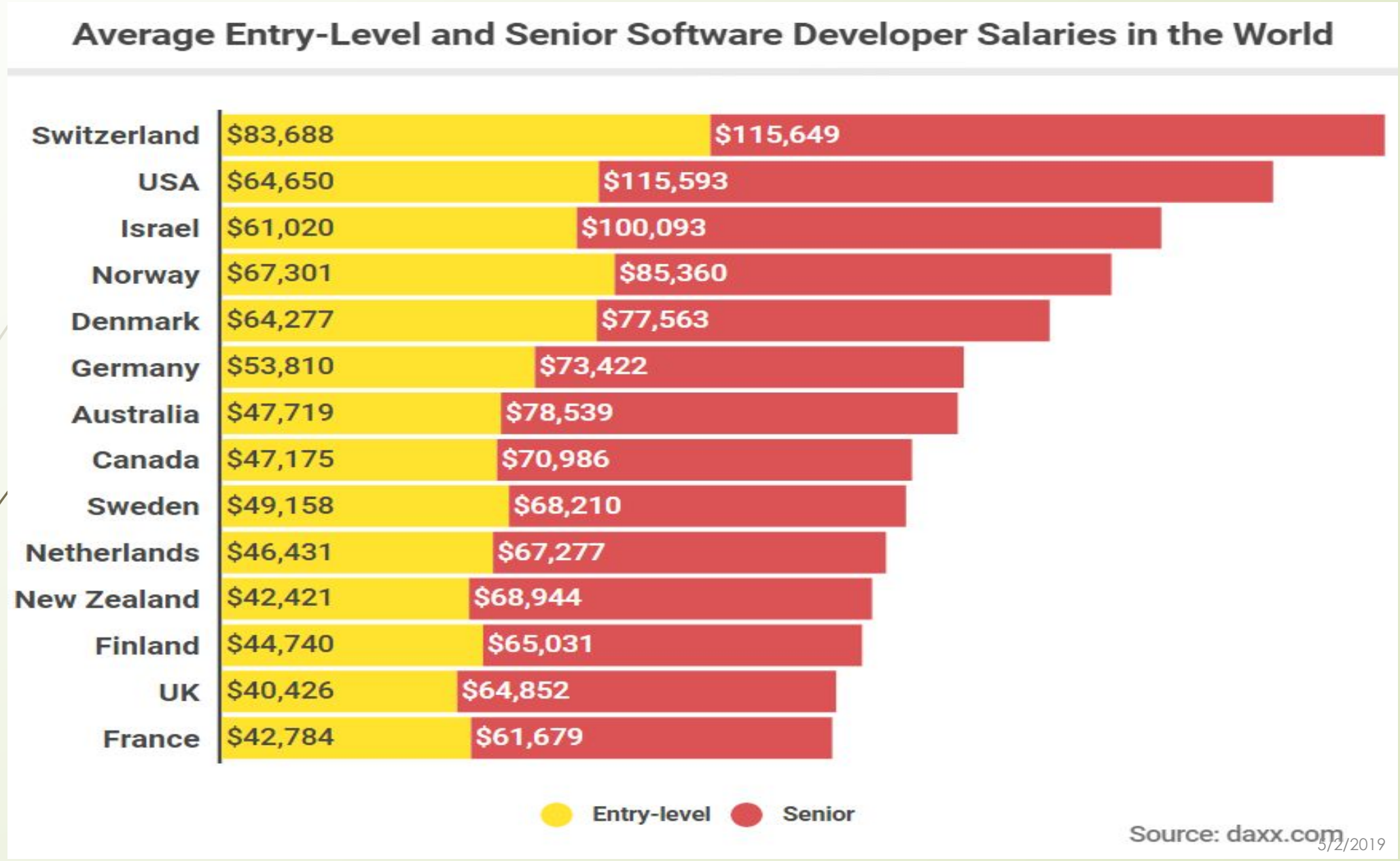
Conditioned yes

- Popularity is a double edged weapon.
- General usefulness of python makes it the **preferred software** in many market players.
- But general usefulness is not the only thing that translates into a vibrant job market. You also need to have a **relatively low supply** of people who have a skill compared to the number of companies hiring for that skill.
- And it is true that Python has **a lot of practitioners**. This is partially because it is a pretty easy language to understand and pick up.

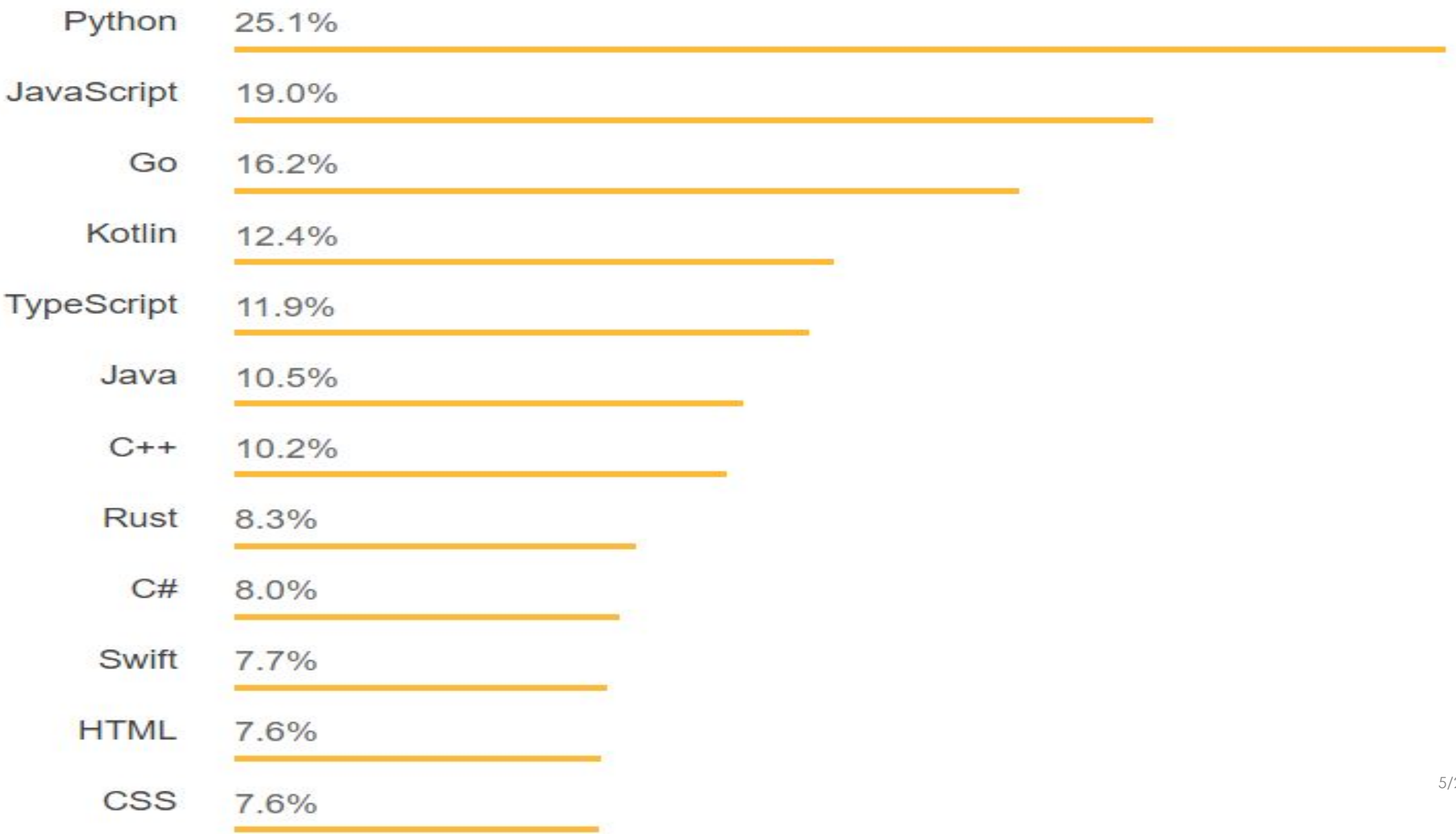
So does that mean that there is a glut of Python developers, driving down wages?

Well, not really

- If you look at the earning potential, of a Mid-level python engineer in the United States they can bring in over **$130,000** a year.
- Obviously, you can **earn more** in senior-level positions.
- Even though there are a lot of people who know Python, there's **a lot of demand** for Python skills which doesn't seem to be abating anytime


Average Entry-Level and Senior Software Developer Salaries in the World
Switzerland $83,688 $115,649
USA $64,650 $115,593
Israel $61,020 $100,093
Norway $67,301 $85,360
Denmark $64,277 $77,563
Germany $53,810 $73,422
Australia $47,719 $78,539
Canada $47,175 $70,986
Sweden $49,158 $68,210
Netherlands $46,431 $67,277
New Zealand $42,421 $68,944
Finland $44,740 $65,031
UK $40,426 $64,852
France $42,784 $61,679
Entry-level
Senior
Source: daxx.com

Most Wanted Languages



conclusion

- Python is well set to be a fundamental skill for the types of technologies which are going to **continue to grow** in the future. So I would say there's probably going to be a **very strong job market** for python programmers for the foreseeable future.

  As processors get stronger and programmer's time becomes more and more powerful, high-level languages like Python will become even more popular.

- But even though The choice of programming language totally depends upon the **purpose** of what it is going to be used for and also largely due to the ease of learning and adopting.

  So go ahead and make your choices based on your personality and market demands.

Take Home Message

- **Remember!**

Biology goes digital

But to be truly competitive, it's important to add another layer of skills on top of your programming skills to be able to grow in the job market and truly solve the problems of tomorrow.