# WHY PYTHON

The reason behind immense popularity of python programming language across the globe is the features it provides. Have a look at the features of python language.

(1) Python Supports Multiple Programming Paradigms

(2) Python Has Large Set Of Library and Tools

(3) Python Has a Vast Community Support

(4) Python is Designed For Better Code Readability

(5) Python Contains Fewer Lines Of Codes

**Future Technologies Counting On Python**

(1) Artificial Intelligence

(2) Big Data

(3) Networking

**However the three top reasons people may decide to learn Python could be:**

1. Machine learning

2. Demand for Python Developers

3. Our favorite framework - Django

In my opinion, Python is especially more popular amongst startups, both for back ends and data analytics.

- **Python Developer**: A Python developer has the following roles:

  1. Use server-side logic
  2. Develop software related to asset management
  3. Write and implement software solutions that integrate different systems
  4. Identify and suggest various opportunities to improve efficiency and functionality
  5. Write reusable and testable code
  6. Support new projects and implement solutions from a Level-3 perspective
  7. Integrate data storage solutions
  8. Implement data security and protection

# WHY PYTHON

## Language Popularity

__TIOBE__: The **TIOBE** Programming Community index is an indicator of the popularity of programming languages. TIOBE programming community index is a measure of popularity of programming languages, created and maintained by the TIOBE Company based in Eindhoven, the Netherlands. TIOBE stands for The Importance of Being Earnest

Following last year's finding by developer hub Stack Overflow that **Python** was the world's fastest-growing **programming language**, **Python** has now claimed fourth place in the TIOBE index for the first time. Aug 7, 2018.

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | | Java | 16.881% | +3.92% |
| 2 | 2 | | C | 14.966% | +8.49% |
| 3 | 3 | | C++ | 7.471% | +1.92% |
| 4 | 5 | ∧ | Python | 6.992% | +3.30% |
| 5 | 6 | ∧ | Visual Basic .NET | 4.762% | +2.19% |
| 6 | 4 | ∨ | C# | 3.541% | -0.65% |
| 7 | 7 | | PHP | 2.925% | +0.63% |
| 8 | 8 | | JavaScript | 2.411% | +0.31% |
| 9 | - | ∧∧ | SQL | 2.316% | +2.32% |
| 10 | 14 | ∧∧ | Assembly language | 1.409% | -0.40% |
| 11 | 11 | | Swift | 1.384% | -0.44% |
| 12 | 12 | | Delphi/Object Pascal | 1.372% | -0.45% |
| 13 | 17 | ∧∧ | MATLAB | 1.366% | -0.25% |
| 14 | 18 | ∧∧ | Objective-C | 1.358% | -0.15% |
| 15 | 10 | ∨∨ | Ruby | 1.182% | -0.78% |
| 16 | 9 | ∨∨ | Perl | 1.175% | -0.82% |
| 17 | 16 | ∨ | Go | 0.996% | -0.65% |
| 18 | 15 | ∨ | R | 0.965% | -0.80% |
| 19 | 13 | ∨∨ | Visual Basic | 0.922% | -0.89% |
| 20 | 21 | ∧ | PL/SQL | 0.702% | -0.51% |

The latest TIOBE index.
Image: TIOBE

TIOBE predicts "it is very likely that Python will enter the top 3 and even might become the new number 1 in the long run".

*Compiled by : Michael Devine. Software consultant & Trainer*
*https://sites.google.com/view/enhance-skill/home*

# WHY PYTHON

As a first language Python has many positives. Python code is relatively easy to read and understand, supports multiple programming paradigms, has a wide range of software libraries that can be dropped into code, and can be scaled to large applications. Obviously every language also has its cons, and Python is criticized for its slow performance, being a poor choice for mobile development and limitations in how it handles database access.

## Redmonk Programming Ranking

Python was also placed at number 3 in the 2018 **Redmonk Programming Ranking**s, which order languages by the number of pull requests for code repositories on GitHub and tags on questions on the programming Q&A site StackOverflow. The RedMonk language rankings are compiled based on the number of pull requests for code repositories on the hosting service GitHub and tags on questions on the programming Q&A site StackOverflow.
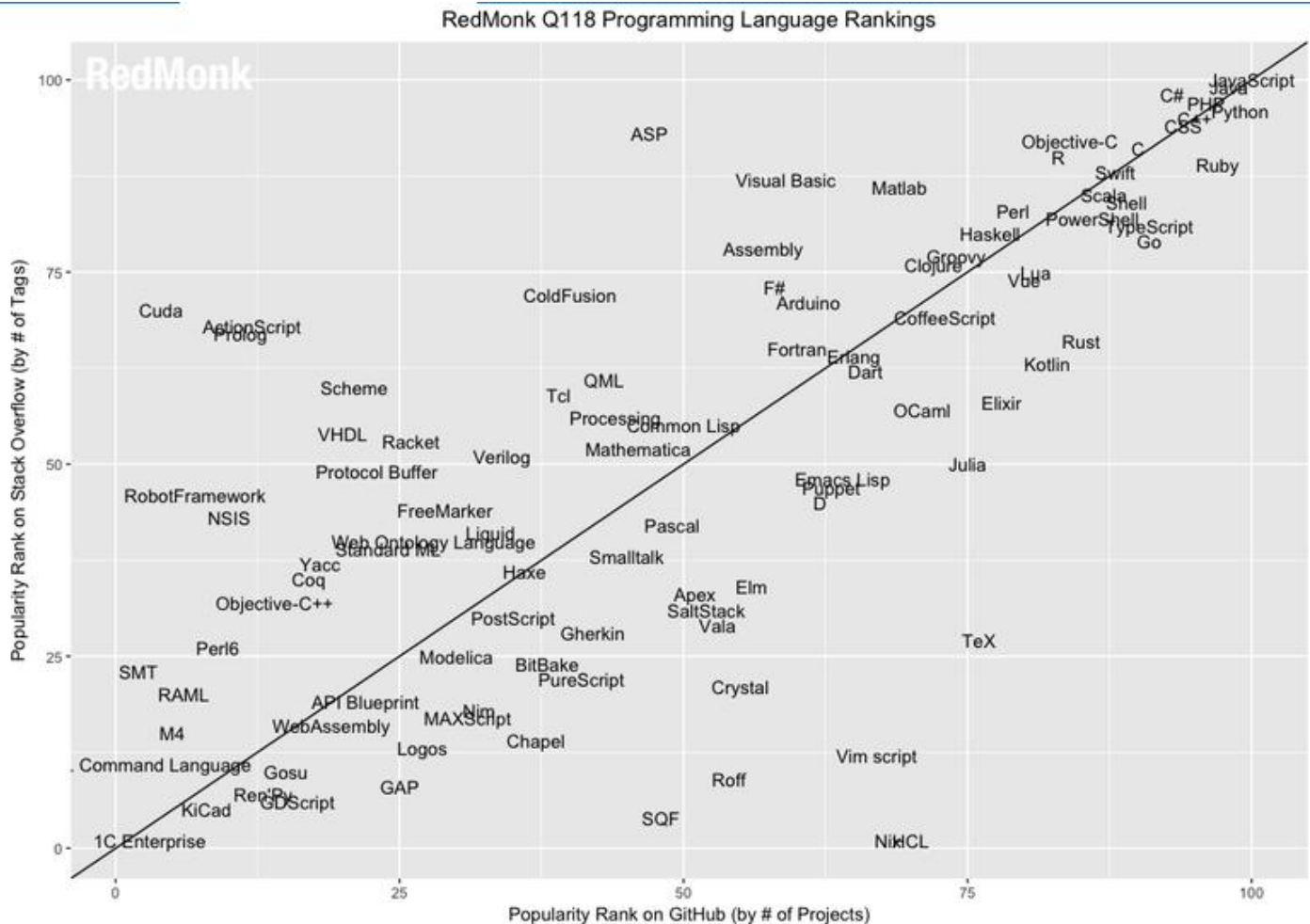
# WHY PYTHON

## Job Trends

The Tremendous Python Career Opportunities in 2018 7: 14 Feb, 2018 in Python Tutorials by DataFlair Team:

Job boards like Indeed and Naukri offer around 20,000 to 50,000 job listings for Python and this shows that Python career opportunities in India is High. Python Careers are good to go with. The below screen shot from indeed job trends how job trends in Python compared to other languages.
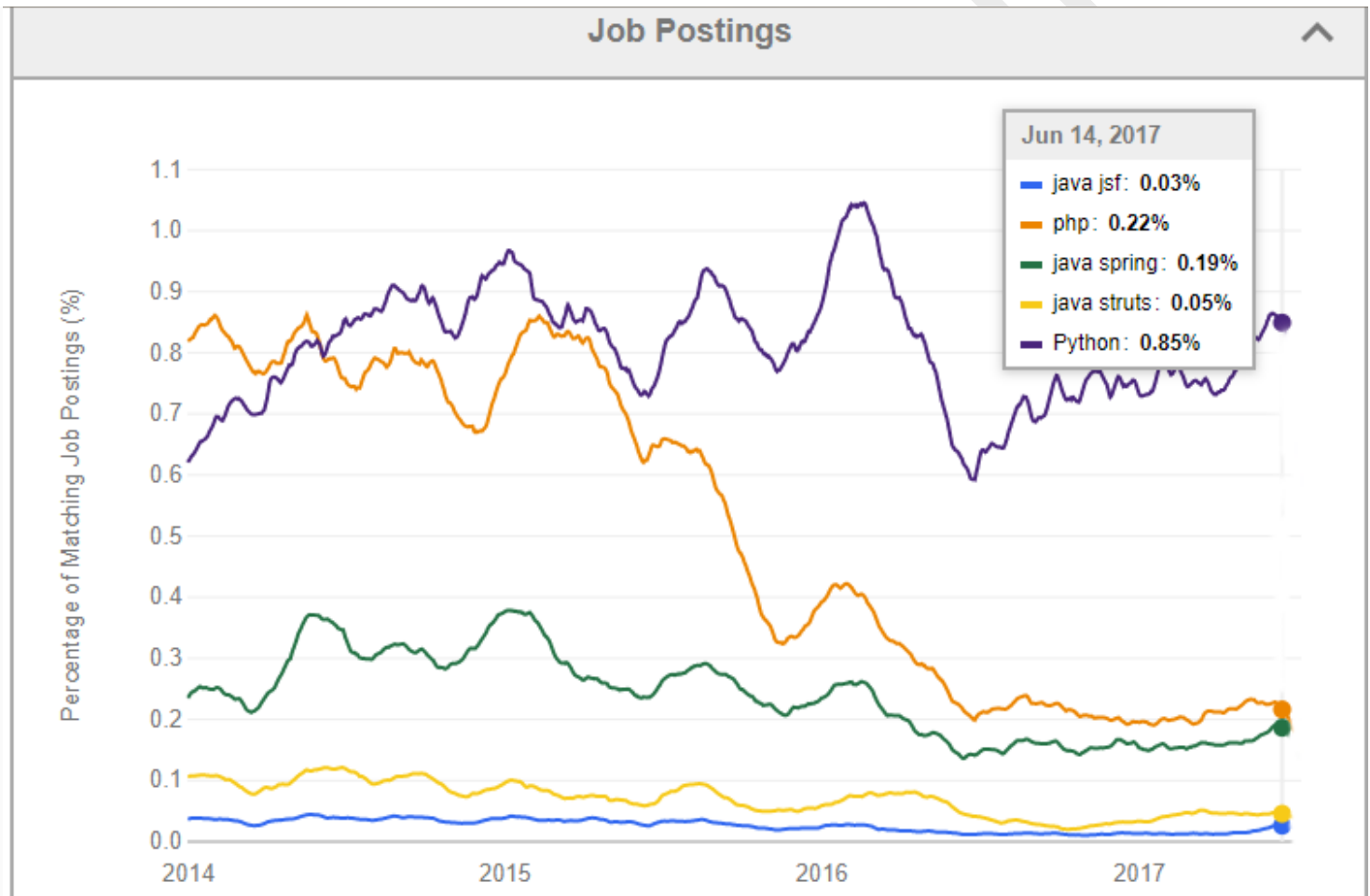


**Job portals like Monster and Naukri offer around 1300 to 3000 listing for Python jobs.**

With Python on your resume, you may end up with one of the following positions in a reputed company:

- Software Engineer
- Research Analyst
- Data Analyst
- Data Scientist
- Software Developer

# WHY PYTHON

**Areas where Python is used.**

Python is being used worldwide in a wide range of environment. Be it as a part of skills, for corporate use or educational settings, it is being taught at both entry as well as advanced level. It doesn't end here; Python is used by Google machine and big projects like Red Hat, Disney etc.
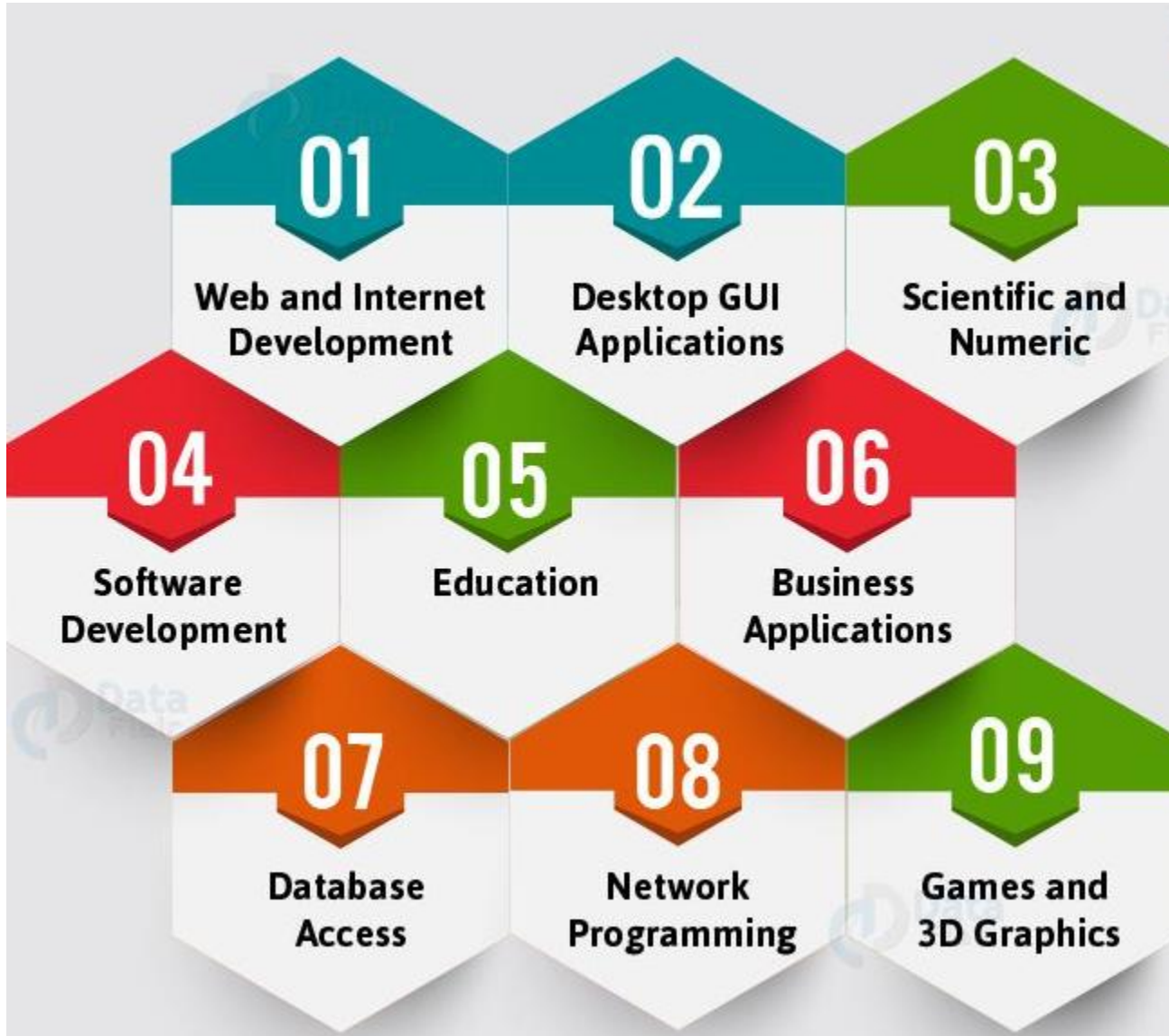
# WHY PYTHON

**Companies using Python and Examples.**

According to realpython.com



Industrial Light & M...  Google  Facebook  Quora  Netflix  Dropbox

**8 World-Class Software Companies That Use Python**

- Industrial Light and Magic.
- Google.
- Facebook.
- Instagram.
- Spotify.
- Quora.
- Netflix.
- Dropbox.

- Red Hat (https://www.redhat.com) is the worlds largest pure-play Open Source Software Company and has many products heavily leveraging Python or are completely written in Python. Examples being Ansible (https://www.ansible.com/) and OpenStack.

- Blogger was originally written in Python.
- YouTube (Alexa rank #2 (1.5B MAU)) is written in Python.
- There's now Grumpy, which transpiles Python to Go.
- Yelp is written in Python.
- Pandora was originally a Django / ActionScript app.
- Reddit is a Pylons app.
- Pinterest (175M MAU) migrated from Django to Flask.
- Instagram is a Django app.
- Disqus is a Django app.
- Zapier is a Django App.
- Patreon is a Flask App.
- Stackshare lists a number of Stacks that containPython
- https://www.python.org/about/success/

# WHY PYTHON

## History of Python

Python was conceived in the late 1980s,[32] and its implementation began in December 1989[33] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL)[34] capable of exception handling and interfacing with the Amoeba operating system

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed.[37]

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward-compatible with previous versions.[38] However, many of its major features have been backported to the Python 2.6.x[39] and 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.[40]

Python 2.7's end-of-life date was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3

## Compiled and Interpreted Languages.

| Compiler Language | Interpreter |
|---|---|
| Takes entire program as single Input and converts it into object code which is stored in file. | Takes Single instruction as single input and executes instructions. |
| Intermediate Object code is Generated | Intermediate Object code is NOT Generated |
| e.g.: C, C++ | e.g.: Perl, Python, Matlab |
| compiled programs run Faster because compilation is done before execution. | Interpreted programs run slower because Compilation and execution take place simultaneously. |
| Memory requirement is more due to the creation of object code. | Memory requirement is Less |
| Errors are displayed after the entire program is compiled | Errors are displayed for each single instruction |
| Source Code ---Compiler ---Machine Code ---Ouput | Source Code ---Interpreter ---Output |

## Why Python as hybrid language.

Hybrid language is a programming language that supports multiple approaches of writing programs. A few examples of hybrid languages are C++, Kotlin, Python, PHP.  Hybrid languages provide the programmer freedom to use whichever approach is suitable in the given context. For example, if you want to write a simple program that sorts numbers alphabetically, you may use imperative programming approach where you specify each step of the process. But, if you want to build a large application to manage all the activities of an organization then you may use object-oriented programming approach because as the program gets bigger, it is easier to mantain with this approach.