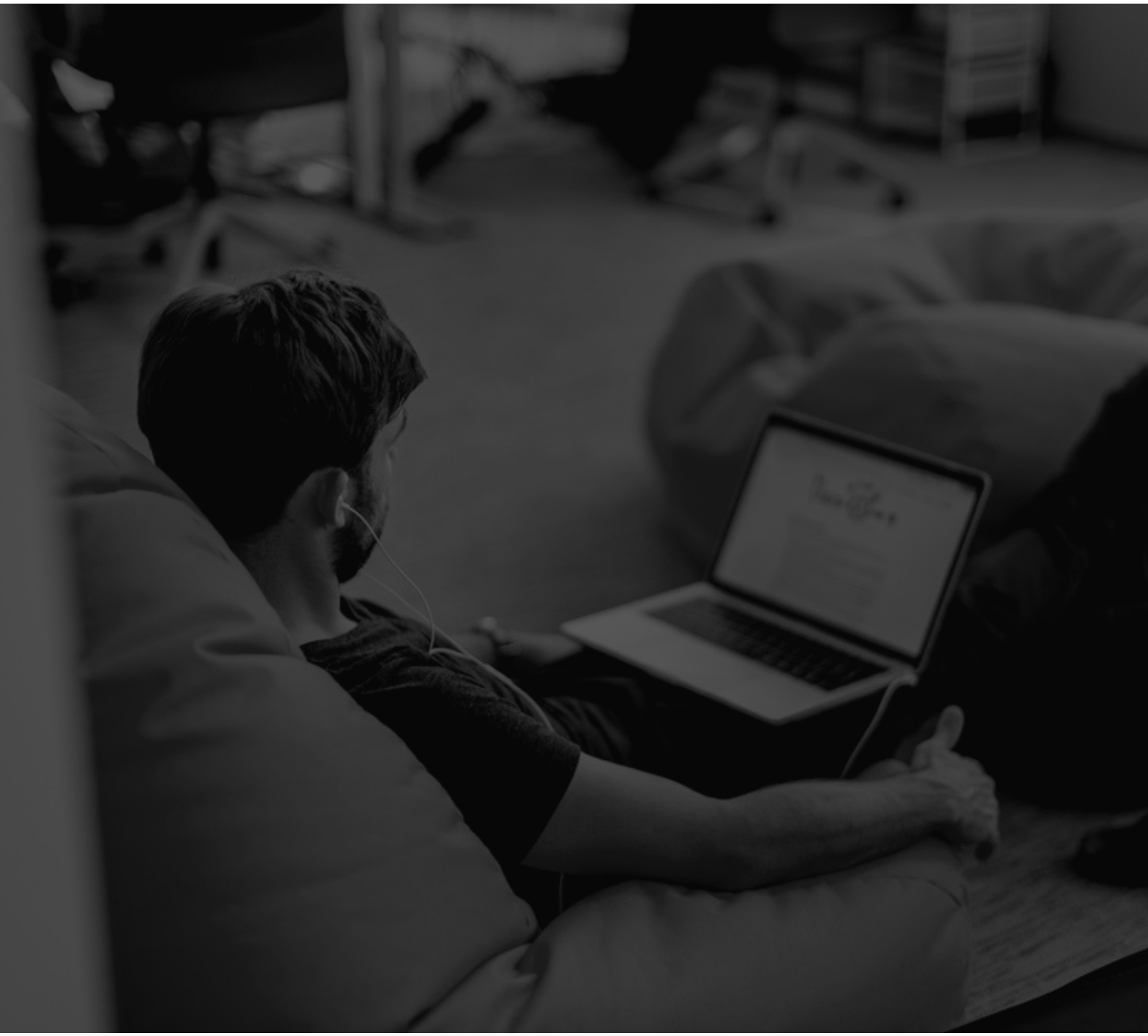
merixstudio



FUEL YOUR PROJECT WITH PYTHON

A SHORT GUIDE FOR CEOS

www.merixstudio.com

CONTENTS

A WORD OF INTRODUCTION

Do you have a brilliant idea for a web app on your mind? So you have to face the serious challenge of choosing the programming language for make it come true. It's a big decision, as technology strongly influences the possibilities of further development of your idea.

As there are many programming languages to choose from, picking the right one may seem to be confusing. Why should you consider Python?

Python has been considered one of the top and fastest-growing programming languages. Ever since it was released almost 30 years ago, Python has cherished a rapidly growing community and become the go-to language for business owners.

When it comes to simplicity and flexibility, Python is definitely one of the favorites. Easy to learn and quick to implement, Python development is readable and might be applied in many diverse cases.

In this guide we are sharing our knowledge about Python so that you could decide whether it is a great fit for your software project.

FOR WHAT KIND OF PROJECT PYTHON IS A GOOD FIT?

what can you do with Python?
where to use Python?
developing web projects with Python
best-known Python web apps

#For what kind of project Python is a good fit?

WHERE TO USE PYTHON?

Basic 3 ways to use Python in the real world are web and app development, Data Science, and scripting.

WEB & APP DEVELOPMENT

When it comes to web development, there's a verity of popular frameworks. The most commonly used are:

- Django (general use),
- **Flask** (smaller projects).

Everything that needs to be done in the back-end development can be done with Python.

SCRIPTING

Another use of Python and possible implementation in a project is scripting.

Basically, **making small tasks automatic**: from email search and classification to desktop apps.

DATA SCIENCE

Next application of Python is Data Science: **machine learning**, **data visualization,** and **data analysis** included.

You can make an algorithm that automatically can recognize, let's say, a dog in the picture. Without a doubt, not everyone needs an algorithm to detect dogs in the pictures but think on a bigger scale. If in your project you need some kind of a **recommendation system, face or voice recognition** - **machine learning powered by Python** is the way to go. For this exact reason, there are already created frameworks and libraries - enough to have a **scikit-learn** or **TensorFlow**.

Last but not least, **data analysis** and **visualization**. Not much to explain here but it's worth to mention the libraries dedicated to them:

- Matplotlib
- seaborn

Instead of comparing many varied data and trying to figure out by yourself why this worked like that - i.e. customer behavior, let the Python do the job. **Compare and visualize the output.**

#For what kind of project Python is a good fit?

WHAT CAN YOU DO WITH PYTHON?

To understand why it is relatively easy to be creative with Python, we should take a deeper look at the language itself.

Basically, **Python programming is very similar to English** and the code reminds of it a bit. There are words like 'in' or 'not' included in the script, which makes it more familiar even for someone that had nothing to do with programming before.

Moreover, thanks to the PEP 8 set of rules, the formatting is not only intuitive but also clear. It literally shows the user where to put the new line and so on.

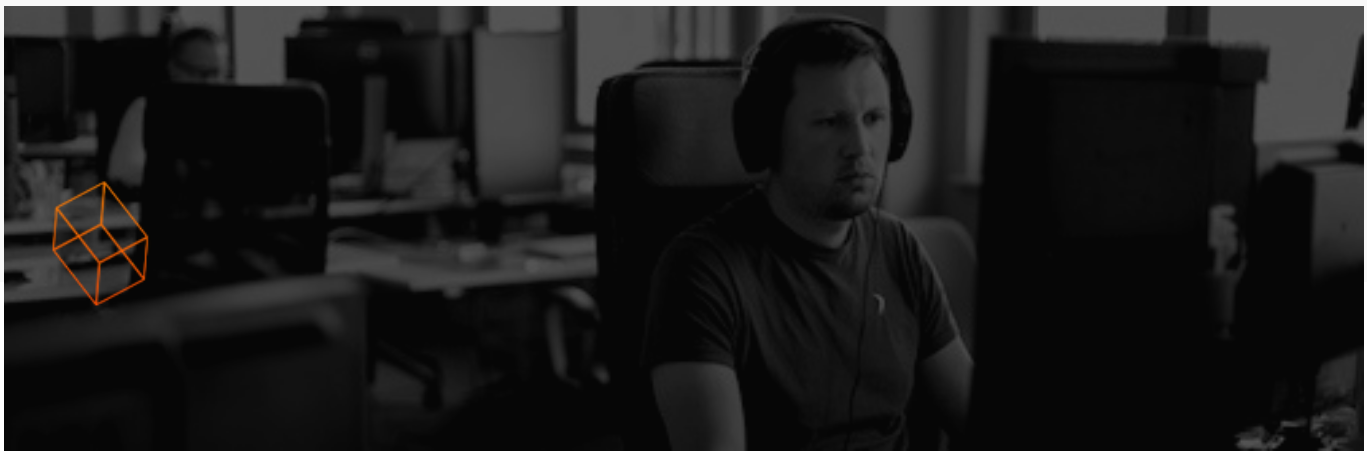
Another important aspect is that the language itself is an **open code one with a strong community** supporting the main idea and working on the language and its libraries. We all know how much easier the life of a developer is with a set of libraries.

Python development gives a lot of chances to make your app/website/project come true quickly and efficiently.

You can:

- **create an app,**
- use it in **data analysis** (anaconda),
- **machine learning** (Kensho),
- **computer vision** (SimpleCV) or just to
- **create a web service** (TravelPerk)

Not only **scripting and automation** can be done with Python but also a **configuration in tools, automate interactions with web browsers or app's GUIs**. Python is also a glue language and enables other codes to work together. Quite helpful when it comes to **machine learning** or **executing data**.



Jun 2019	Jun 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.004%	-0.36%
2	2		C	13.300%	-1.64%
3	4	^	Python	8.530%	+2.77%
4	3	v	C++	7.384%	-0.95%
5	6	^	Visual Basic .NET	4.624%	+0.86%
6	5	v	C#	4.483%	+0.17%
7	8	^	JavaScript	2.716%	+0.22%
8	7	v	PHP	2.567%	-0.31%
9	9		SQL	2.224%	-0.12%
10	16	^^	Assembly language	1.479%	+0.56%

tiobe index

STATISTICS

In June 2019 Python overtook C++ and joined the top 3 most popular programming languages in the world. Its popularity continues to soar month by month.

#For what kind of project Python is a good fit?

DEVELOPING WEB PROJECT WITH PYTHON

Python has won the title "programming language of the year 2018".

Python has received this title because it has gained most ranking points in 2018 if compared to all other languages.

It is the most frequently taught first language at universities nowadays, it is number one in:

- the **statistical** domain,
- in **AI** programming,
- in **scripting** and
- in writing **system tests**.

Besides this, **Python is also leading in web programming and scientific computing** (just to name some other domains).

In summary,
Python is everywhere.

Moreover, as a dynamic type system language with automatic memory management, is often preferable by programmers.

No matter how big in scale the project is, with Python you can **create simple apps**, **simple web pages** or **state-of-art applications** which are going to expand in the future and you want to keep the room for improvements.

#For what kind of project Python is a good fit?

BEST-KNOWN PYTHON WEB APPLICATIONS

App development using Python leaves the door wide open for your creativity. Here are some examples when Python would be the best choice for the project.

If you come up with the idea to put a question about it on **Google**... basically, you are served by Python.

and then one of the first websites popping up - **Quora** - is also using... Python.

If you dig more, you get to **Reddit**, and guess what - Python again!

Check your **Instagram** account in the meantime and play a song on **Spotify** or **YouTube.** Of course, all powered by Python.

Looking at those examples you might get a false impression that Python is only for the 'big' ones in the industry. Nothing could be more wrong!

Python, as a language, is beginner friendly, but still extremely powerful.

You may see the majority of Python applications in this list.

PYTHON & STARTUPS: WHY IT IS A PERFECT MATCH

why should you consider Python?
building an MVP with Python
scaling business with Python
managing Python project
skilled Python Developers
why should you work with an experienced Python team?

#Python & startups: why it is a perfect match

STARTUPS: WHY CONSIDER PYTHON?

Technology strongly influences the possibilities of further development of your startup.

As there're many programming languages to choose from, picking the right one may seem to be confusing.

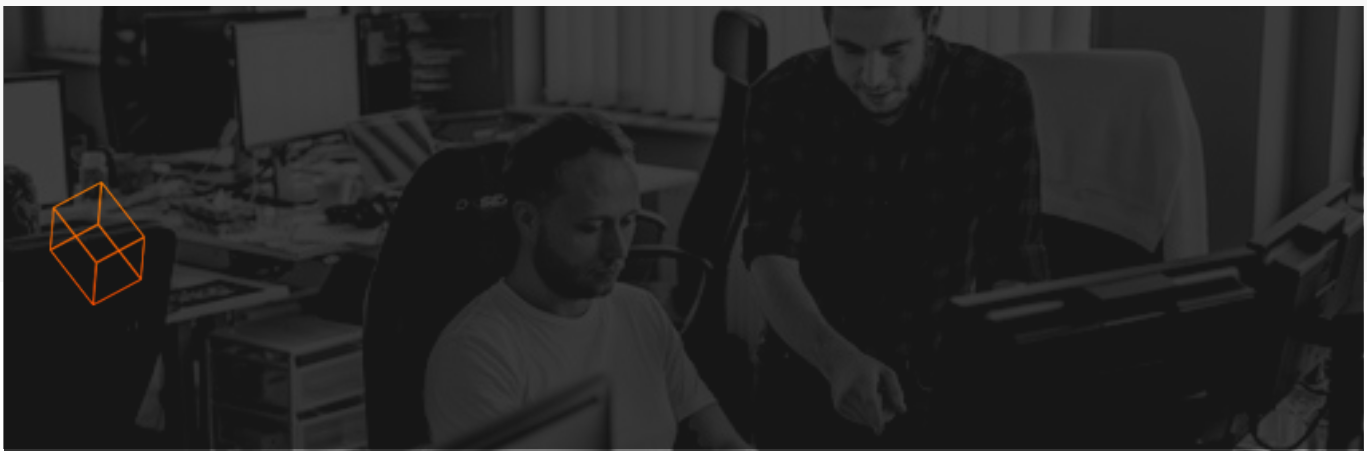
It's worth to do a cold calculation and think deeply about what challenges your startup will face both in the near future and in the long term. Although each startup is different, we can identify certain joint needs for all of them:

- quickly built MVP,
- fast iterations,
- implementation of new features when necessary,
- easy integration with other software,
- scalable business,
- a varying number of team members.

What's extremely important is that startups have limited time and budgets. It should be taken into account while choosing the programming language.

Python proves to be quite economical concerning both time and money.

Below you will find a list of business reasons why should you consider Python as a programming language in your startup.



#Python & startups: why it is a perfect match

BUILDING AN MVP WITH PYTHON

Python is great for building an MVP.

'Time is money'.

It's a true statement for all startup founders.

Making a quick yet functional prototype is crucial to convince investors that there's a huge potential behind the project.

So, if you need to test your business model or keep your partners calm with quick iterations (sure you need to), then speed is of the essence.

Some programming languages are designed to promote rapid application development, and Python definitely is one of them.

Thanks to its modules and packages, which are at the core of reusable code, the majority of projects based on Python is completed faster than with the use of e.g. Java or C++.

So, the truth is that **Python helps startups iterate quickly after every feedback cycle and build an MVP extremely fast**.

#Python & startups: why it is a perfect match

SCALING BUSINESS WITH PYTHON

Python is great for building scalable web applications.

Startup founders not only have to focus on short-term results but also build the foundations for long-term success in a meanwhile.

To realize their business goals and handle rapid growth, they need to choose a programming language that will allow them to achieve scale.

That's where Python comes into play.

The key to success is its **simplicity and consistency**, e.g. **adding new features is super easy**, developers write less code for the same functionality than, for example, using Java, and it can be run in a virtual machine like PyPy, JVM, or CLR for additional performance increases.

Still not convinced that Python is scalable? Look at YouTube - almost 2 billion unique visitors per month, billion hours watched daily and 400 hours of video content uploaded every minute, **all with Python as a core technology.**

#Python & startups: why it is a perfect match

MANAGING PYTHON PROJECT

Python let the agile teams create good-quality products.

In general, **projects using Python don't require as many developers as other projects**. An agile team of experienced developers led by a brisk project manager is enough to make a functional application.

As developers often highlight, **Python is very clear and intuitive.** It's **major strengths are readability and speed of app development.**

What's more, Python has a lot of features and characteristics that developers appreciate as it's a "**strongly typed dynamic language**." It also has a ton of optimizations built-in to help speed it up.

Python's other great strength is an extensive set of libraries that allow it to perform a wide array of tasks.

Last but not least, the Python community is already large, strong and supportive and is getting more powerful each month.

In effect, a small team of developers can write code quickly and almost effortlessly.

As we mentioned above, all the startuppers lead a sort of battle against time, and that's why it is worth considering Python as a programming language in your startup.

#Python & startups: why it is a perfect match

SKILLED PYTHON DEVELOPERS

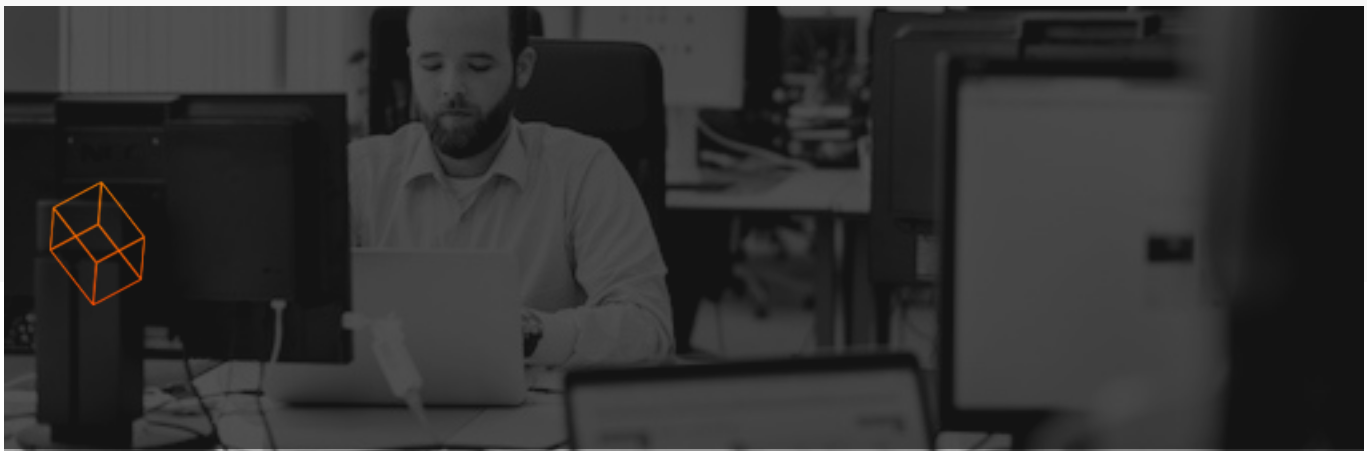
Python is popular among developers.

Why should the popularity of programming language have been taken into account when choosing one for a startup?

There's a saying that **all companies are tech companies now**. As startups strive to put state-of-the-art, cutting-edge products and services on the market, the technology plays the crucial role in fulfilling their dreams. It means that one of the most important aims of a startup is to hire the right people - tech experts, who will be able to support the project.

Python community is large and stable - it's rather unlikely that this programming language will soon fall into oblivion and your web application will have to be rewritten. An active community guarantees that Python will be supported, maintained and developed on and on.

Of course, what counts is quality, not quantity. What your startup really needs to gather momentum, is a team of skilled engineers. How to reach them? There are many ways to hire experienced python developers, e.g. you can work with top web software houses, especially when you have to succeed fast.



#Python & startups: why it is a perfect match

WHY YOU SHOULD WORK WITH AN EXPERIENCED PYTHON TEAM?

As a startup you need to grow fast, that's obvious. **Choosing Python as a programming language definitely may help you prepare rapid prototypes**. However, does the decision itself guarantee that your project will be realized fast and will succeed?

One thing is for sure:

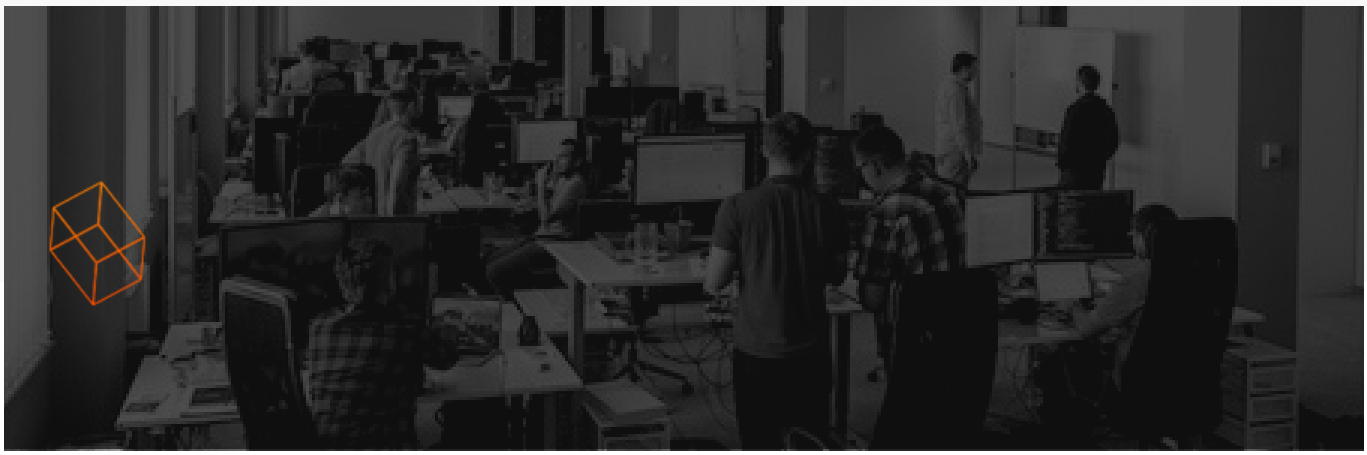
experienced developers will work faster than younger developers, regardless of the language they work with.

Competencies of developers are crucial for your startup to succeed.

However, creating one's own team of software developers is a massive investment. Not all startups are ready for it at the moment when it becomes necessary to make a rapid prototype. That's why it may be a good idea to **seek the support of a remote team, especially when you're a non-technical founder**.

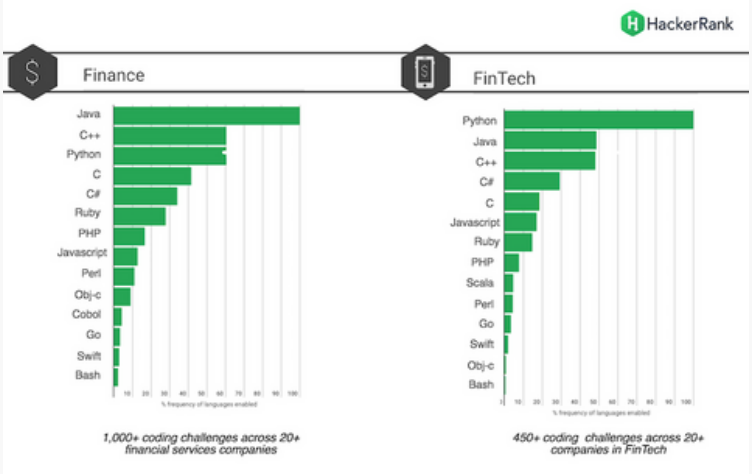
It's logical that if you decide to have Python in your tech stack than it's worth to cooperate with the best Python developers worldwide.

You will find more practical tips on how to choose the right digital team for your startup in this short article.



PYTHON & INDUSTRIES

why successful FinTech apps are built with Python?
why is Python so popular in Machine Learning?
is there a room for Python in the hospitality industry?



hackerrank

STATISTICS

HackerRank has analysed 3,000 coding challenges across six industries. To be included, each industry had to represent at least 20 companies.

#Python & Industries #FinTech

PYTHON & FINTECH = PERFECT MATCH

Simplicity of Python equals fewer errors and less debugging. Creating a financial service is demanding and it is already quite complicated, thus the more transparent the programming language the better - not only for the startup owner but also for the dev team.

Due to Python's flexibility, it is simply a great choice for FinTech.

What follows the simplicity factor is the time asset. **Writing a few lines of the same code in Python takes less time** than if it was written by the vast majority of languages. But what makes it so suitable for finances sector? It's the amount of **open-source financial libraries**.

Take a look at **Python libraries for FinTech:**

- SciPy, NumPy - scientific and technical computing,
- pandas - data analysis,
- pyrisk - financial risk and performance,
- pyalgotrade, zipline - algorithmic trading library,
- ffn - financial function library,
- scikit-learn - machine learning algorithms,
- finmarketpy - backtesting trading strategy and financial markets analysis.

There are more ready-made libraries to use within Python in regard to finances, stocks or even quantitative economics (quantecon.py). Python might be used in general banking services, data analysis, cryptocurrency markets, algorithmic trading, pricing, trade, and risk management.

#Python & Industries #FinTech

FINTECH STARTUPS BASED ON PYTHON

ZOPA

It's a personal finance peer-to-peer lending platform on the market since 2005. Regarding technical features, Zopa (website and mobile app) is created with Python, C#, and Java. Python is used to implement sophisticated algorithmic models. Also, the dev team has built a machine-learning stack which runs on Python.

VENMO

Social media and a payment service in one. You can chat with your friends, use it for payment in stores or transfer money to another account. In 2009, the reason was simple - split the bill between friends for shared services like dinner or paying for a taxi. Currently, it is converted into a full-fledged payment system.

QONTO

In simple words Qonto is this for small business what N26 for individuals - an application based on a subscription (9 euro per month) with a Mastercard and opened account, of course as well with IBAN, so users can receive payments. The data team is focused on a result-oriented approach and uses Python to make it all work better.

FIGO

A German-based banking as a service platform that enables connection between banks (more than 3000) and other financial institutions via API. Banks which use figo can offer their customers to access various accounts and deposits in one online banking system, and from the user side - all data is in one place.

ROBINHOOD

This app allows the user to trade stock without any trading fees. Currently has also a feature in a new field: cryptocurrency. You can buy and sell Bitcoin either Etherneum via Robinhood and don't get additionally charged for the transaction. Python and Django were used in the app.

VYZE

"Enterprise retail solutions for every way your customers shop". Vyze platform is a full package product which combines lending supply, customer support, and technology. Accessible from POS terminal as well as through the Cloud for eCommerce services with adjusted API integration.

#Python & Industries #Machine Learning

MACHINE LEARNING: BASICS

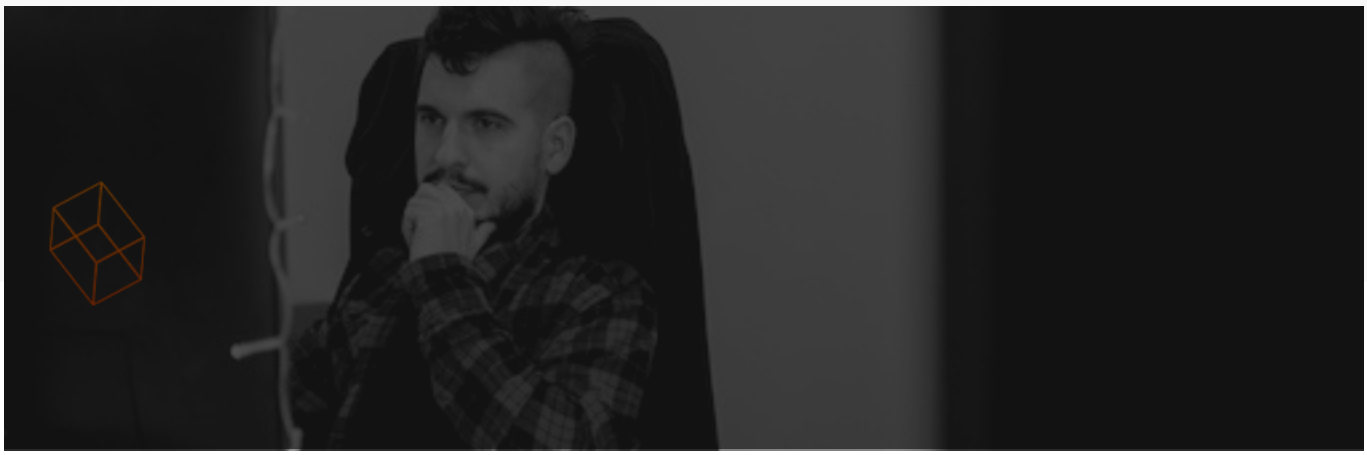
Why is it often a good idea to implement machine learning in your project and why is Python the best choice for that?

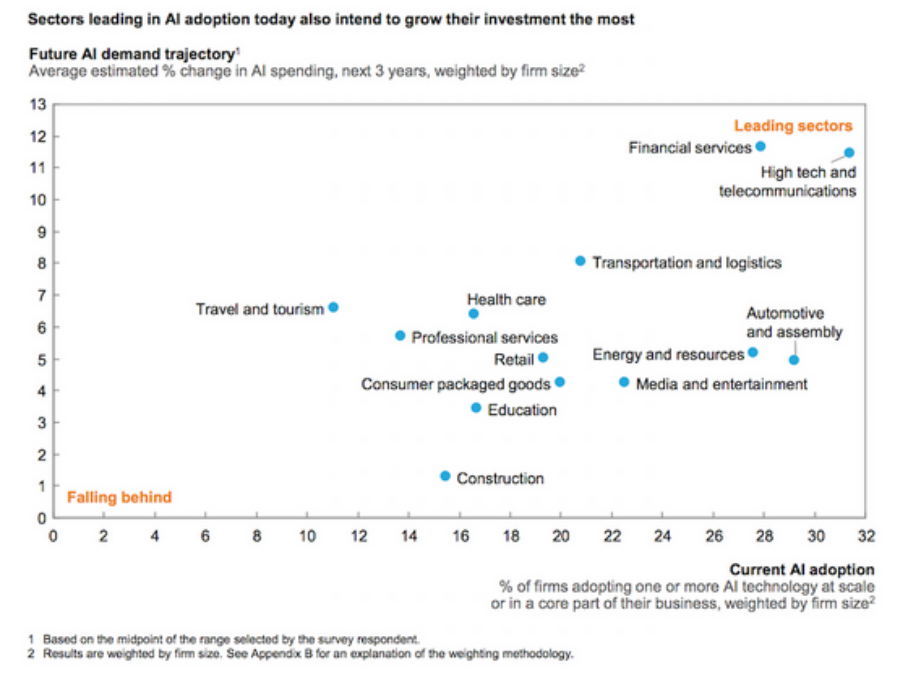
Imagine a computer which can make decision and predictions based on the data you have given it before. The machine gets a sample of data (the quality of it is also an important factor) and through the algorithm which is analyzing it, the machine 'learns' (recognizes patterns in the data) - after that is able to make decisions or predictions. A step further is **deep learning which can come up with conclusions similar to human-like decision-making pattern.**

A simple example of machine learning is a spam detection algorithm or categorizing and labeling received emails, creating a product recommendation list and so on.

To make it clear, let's name only a few **real-life examples of machine learning**:

- **Siri** - the voice recognition app learns to 'understand' the nuances and semantics of our language.
- **Google Maps** - its algorithm is analyzing the anonymous location data from smartphones to know about speed of traffic
- **Gmail** - since a while, you have three automatic responses to the email you have received. Coincidence? No, Machine Learning.
- **Netflix and Spotify** - based on videos and music you have watched or listened, the engine is matching you with similar positions.
- **Uber** - the time of arrival and pick-up location is determined heavily on ML.





STATISTICS

Machine learning or in general AI work more than fine in many industries. Take a look at the Future AI demand trajectory prepared by McKinsey&Company to get the point.

#Python & Industries #Machine Learning

MACHINE LEARNING IN INDUSTRIES

The challenge for a machine learning engineer is to first take the data, process it, clear it out and arrange, to finally develop intelligent algorithms which, based on the collected data, are able to make decisions on their own. Sounds time consuming and a bit tricky, right? This is when Python comes to the rescue!

#Python & Industries #Machine Learning

MACHINE LEARNING WITH PYTHON

Why is Python a great choice for machine learning implementation?

There are two main reasons for choosing Python: its simplicity, as well as the full range of available libraries and frameworks.

When it comes to big data or just very complex decision-making path, the simplicity is the key. Python is well-known for its readability, developer-friendly syntax, and semantics. The less the developer has to worry about the code itself, the more focus and emphasis can be put on finding solutions.

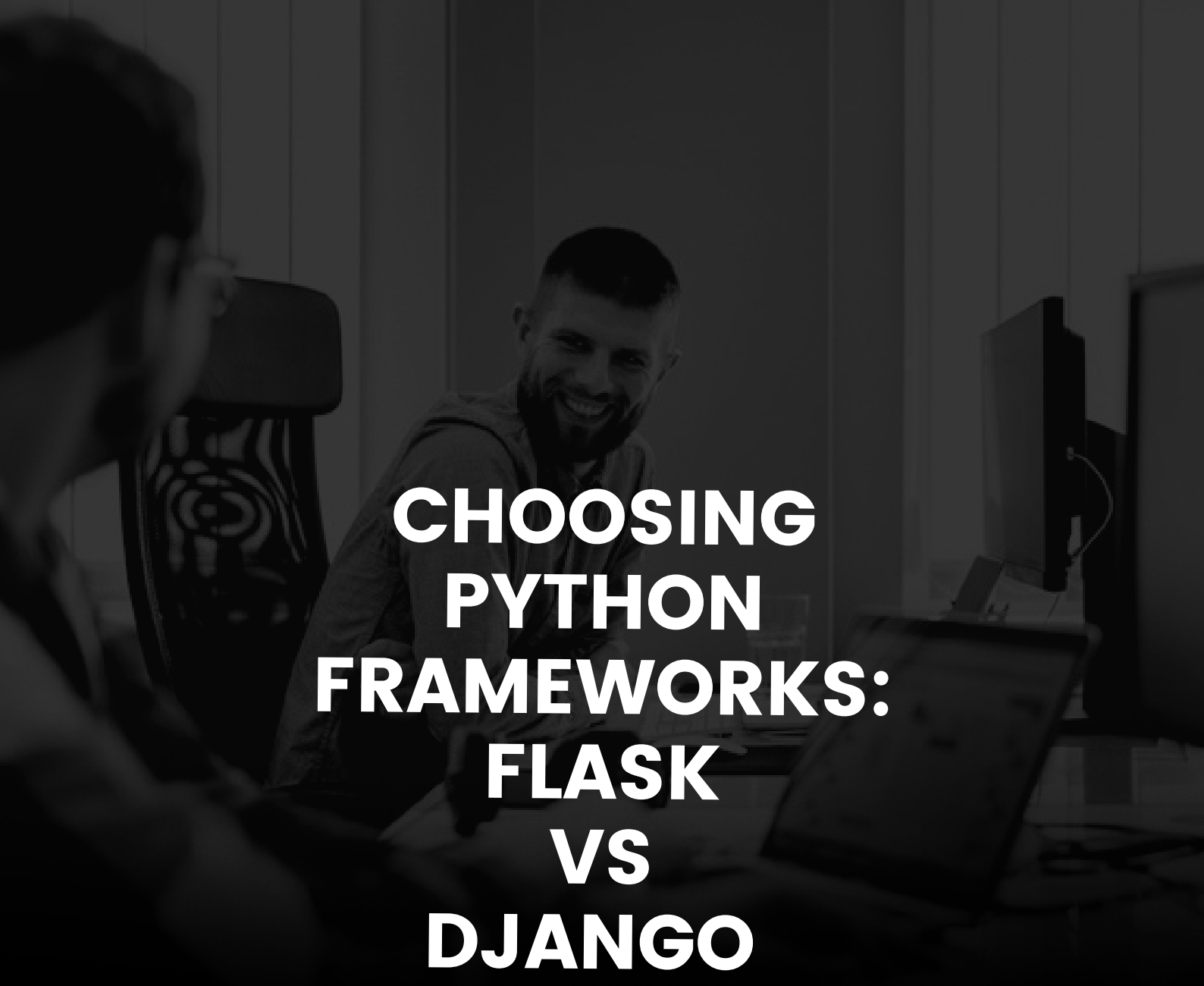
Not only the ability to take away the cognitive overhead from a dev team and its simplicity is guaranteeing Python the first place as a cutting-edge language for machine learning. What makes Python so suitable is the amount of ready, open source libraries and frameworks.

LIBRARIES USED IN ML WITH PYTHON:

- Pandas - we have already established that machine learning requires good data and to handle it well Pandas come to the rescue;
- NumPy - very useful when you need to manage data like matrices, often paired with Tensorflow or SciPy (or many other libraries which allow you to analyze data better). NumPy helps to deal with complex statistics, especially for neural networks;
- scikit-learn - complex data algorithms with data visualization techniques in one library;
- Tensorflow - a great use example is Google photos or Google voice search;
- Theano - fast numerical computation for multidimensional arrays;

There are many more Python libraries for machine learning and data science in general but those few mentioned above should give an overview on how many ready-made options there are and how much faster the process of finishing the project could be.

> *Depending on the project and its goal, there are many languages, tools and frameworks which could be handy. However, due to Python's simplicity, versatility and readability, it makes dev team's work lighter. Additionally, when the task is complex, the machine learning-focused libraries available for Python can save time and stress.*



CHOOSING PYTHON FRAMEWORKS: FLASK VS DJANGO

frameworks: basics
Django & Flask: characteristics
Django & Flask: comparaison

medium

STATISTICS

Top Python Framweorks for web developers in 2018

#Python frameworks

FRAMEWORKS: BASICS

To develop a web application in Python it is better to use a so-called web framework.

By using it, you can save time and avoid getting bored before actually starting your project. Top Python developers don't bother with building from the scratch backend logic, user interface or the link between your app and the browser so it is easy for the user to navigate in it.

Web framework gives the **ability to implement some basic functions which are the same for the vast majority of web applications**. In this way, you save time and energy. Exactly this, what is already in the framework and this what needs still to be done by the Python developer, depends on the framework you choose.

Of course, there are many more frameworks than Django and Flask, i.e.:

- Pyramid,
- TurboGears,
- Bottle,
- CherryPy,

which might be a better choice for a particular project and we shouldn't forget about them.

However, let's cut straight to the chase and dig into differences between the two most popular web frameworks used by top Python-devs.

#Python frameworks

FRAMEWORK FOR A MID-SIZE PROJECT

Flask or Django: which framework suits better for a mid-size project?

Which one would be more suitable for your project? Obviously, it will also resonate with your choice upon the outsource Python development company providing you with the service.
The basic variable is the **scale, full-stack web framework (Django) vs micro and lightweight Flask**.

In Django development, there is quite a lot already implemented, while Flask includes a necessary minimum and gives more space for the developer.

DJANGO:

- you get a full package: an admin panel, ORM, database interface, directory structure for apps;
- Django-admin makes it easier to handle the administrative part of common projects due to ready-to-use and fully functional admin framework;
- longer on the market (2005) and a bigger community;
- widely used;
- time-saver due to a built-in template engine;
- allows Python developers to build web apps without an external input and also to create new applications in the project, thanks to a built-in bootstrapping tool;
- ORM system enables developers to work with databases (MySQL, SQLite, Oracle and so on) with ease;
- no need to use any third-party tools or libraries;
- security features;
- clean code;
- on the cons side, Django is a large framework and might be overwhelming in particular for beginners;

FLASK:

- light, simple and flexible;
- gives space to the developer;
- based on the Jinja2 template engine which is an easy environment to accelerate the development of dynamic web apps;
- extensible framework with a customizable structure;
- perfect for smaller scale projects which need to be done quickly;
- isn't that powerful as Django due to its microstructure;
- beginner-friendly;

FLASK

Flask is usually used in smaller scale applications, thus it's more difficult to find real-life examples. However, let's not forget that Flask enables a developer to accelerate and enhance the simple web app - but add-ons are not built-in like in Django case. Based on job postings, it might be taken for granted that companies like Linkedin, Uber or even Pinterest mentioned before (and developed mainly with Django), are using Flask in their backend. For microservices, Flask is usually a faster solution.

DJANGO

Quite a few high-traffic websites are developed with Python. Starting with Google, which uses Python (and Django) to support its search engine and handle the traffic to YouTube, originally based on PHP, but together with rapid growth, moved to Python. Another platform using Python is Instagram - based on Django to make it all work with high efficiency in a real-time, while also using it in the back-end. Python and Django appear in many, many services and apps, regardless of the project size.

PYTHON VS PHP

Python vs PHP: basics
Python vs PHP: core differences
Python vs PHP: which one to choose?

#Python vs PHP

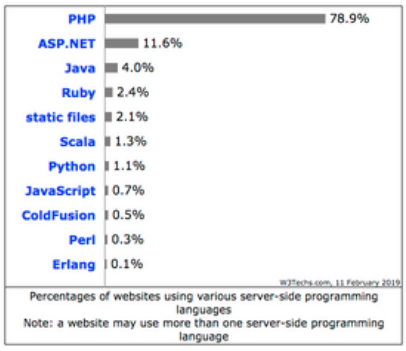
PYTHON & PHP: BASICS

A flexible language for every project - does it exist?

Web design and app development is a long distance run. You need to bear in mind coding at the client side and server side, then decide upon Database technology.

For each step, the choice of the particular programming language is crucial.

There are a few ways to compare programming languages, how++ever, what **most likely concerns you is the... budget**, and after that - the final outcome.



*When it comes to **PHP**, it is the most popular language for the server-side programming.*

Jan 2019	Jan 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.904%	+2.69%
2	2		C	13.337%	+2.30%
3	4	^	Python	8.294%	+3.62%
4	3	v	C++	8.158%	+2.55%
5	7	^	Visual Basic .NET	6.459%	+3.20%
6	6		JavaScript	3.302%	-0.16%
7	5	v	C#	3.284%	-0.47%
8	9	^	PHP	2.680%	+0.15%
9	-	^^	SQL	2.277%	+2.28%
10	16	^^	Objective-C	1.781%	-0.08%

*On the other hand, we have **Python** which is praised for its simplicity and dynamics.*

#Python vs PHP

PYTHON & PHP: CORE DIFFERENCES

A flexible language for every project - does it exist?

First of all, the mindset. **Python development is aspect-oriented while PHP is object-oriented**. Well, this sounds fancy but what does it mean for your project?

With PHP, developers would make up an integral code object which executes particular tasks based on the user's input. In Python development, the program is split into modules depending on the functionality. Thus, in Python, the dev team would first create modules and after making the connection depending on the "if-then" action. Triggered by the user's action, the algorithm executes a particular program block.

Another important difference is syntax, a.k.a fundamental grammatical rules of the programming language. However, let's leave this to the dev team and get back to major differences that matter in the area of time, efficiency and budget.

PHP is considered less flexible than Python and has more strict rules, as well, PHP requires libraries to be loaded manually, while Python uses packages to load them in a breeze.

Moreover, PHP is considered not that safe as Python applications because in PHP there is a need for additional tools to support security. However, there must be a reason why over 80% of the web is using PHP, right?

Let's not forget that PHP is made for web developers to create dynamic and interactive HTML pages. Most popular CMS platforms like Wordpress, WooCommerce and Magneto are made in PHP. Moreover, websites like Wikipedia, Yahoo, Tumblr and to some extent Facebook, are written in PHP too.

PHP code can be simply inserted into the HTML and voilà! When it comes to frameworks that make the life of a developer easier, PHP ones as Symfony or Laravel have many enthusiasts. Another important fact worth to mention is that **PHP is simple to update** - the update is always from the server-side, thus additionally it is basically the safest way to apply improvements.

For Python, the most popular frameworks used to create web applications are Django and Flask. Regardless of its frameworks, **Python grows in popularity since Google decided to base some projects on this language**. Other giants as **YouTube, Spotify, Instagram or Reddit** are also following the trend. Not surprising if you care about **efficiency and how easy you can acquire it into a project**.

#Python vs PHP

PYTHON & PHP: WHICH ONE TO CHOOSE?

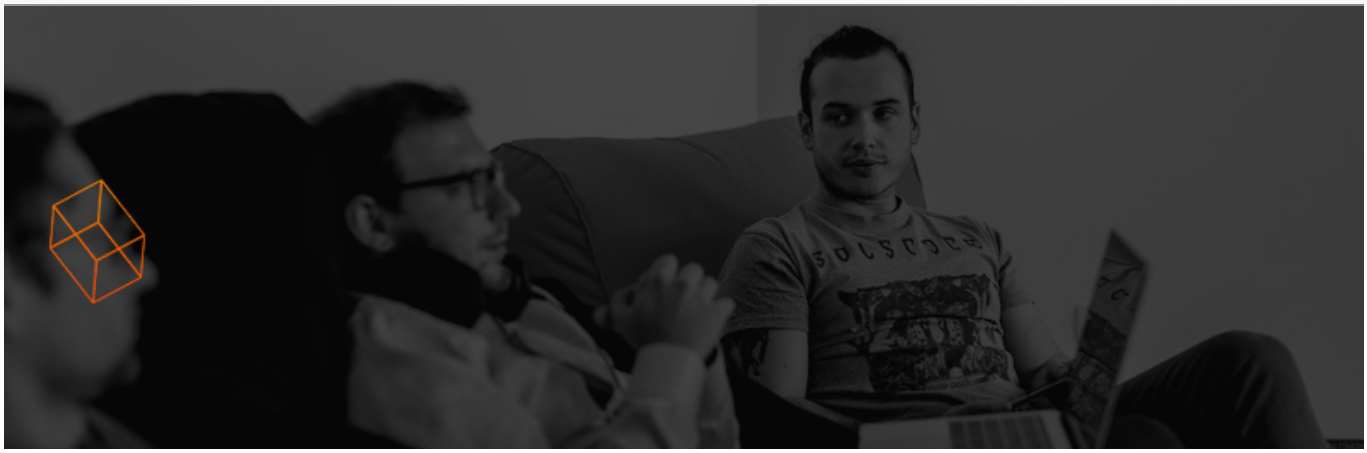
Python or PHP: which one is better for a small business or a startup?

PHP and Python have a lot in common and both are quite powerful when it comes to web development, however, in the end, you have to make a choice.

Let's make it clear, in best software development firms you will find programmers for both languages. PHP is the core of the vast majority of web apps while Python is gaining more and more audience due to its readability and flexibility.

Depending on the project, if you strive for a state-of-art app (not only for web development purposes!), which might be fully accustomed in the future and additionally, you want to make it visible quickly, **Python seems a better choice**.

Definitely there is a room for PHP as well, however, when the time is money and you need to sell your idea before anyone else comes up with it as well, a quicker solution which supports also the quality of the product is simply wiser.



FINDING & HIRING TOP PYTHON DEVELOPERS

what should a Python dev know?
a software house or a freelancer?
success of a web project

#Finding & hiring Python devs

WHAT SHOULD A PYTHON DEV KNOW?

The minimum level of skills required for a great Python developer.

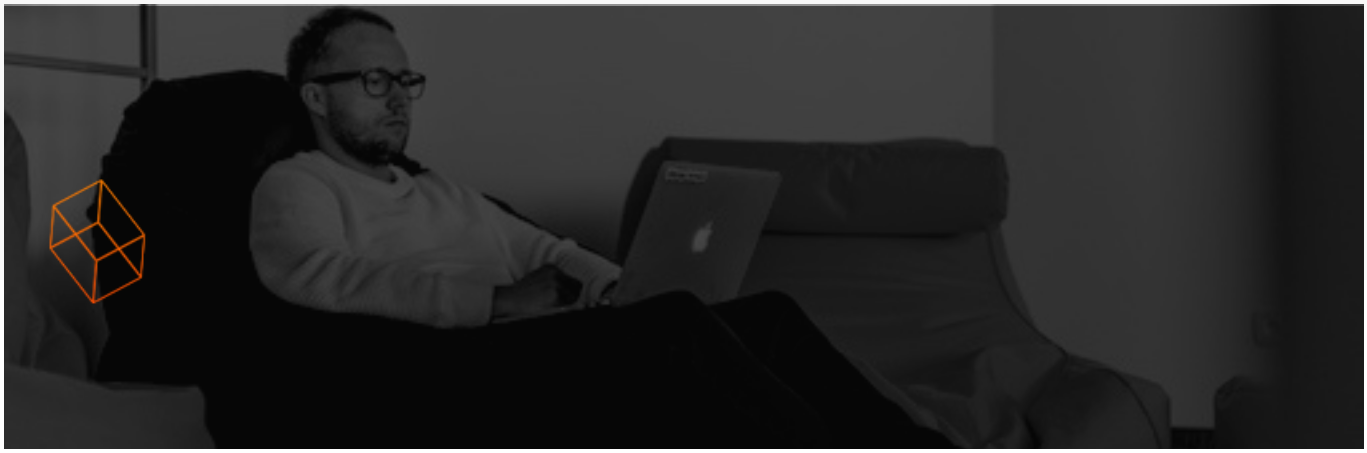
Regardless of knowledge of Python itself, its syntax and semantics, obviously, the dev (team) should understand in-depth **Python frameworks** (whichever you use in your project, but most likely it would be Django or in case of a smaller-scale project, then Flask - but those are not the only ones in common use).

Another skill required is at least basic **knowledge of Object Relational Mapper (ORM) and familiarity with front-end (as HTML5, CSS3, JavaScript)** in case of web applications. A

Python dev should also be able to use **RESTful API, databases** and understand how **System Administration** works.

Moreover, Python developer should have the ability to **write tests and know how to optimize the code**. It goes together with critical thinking and problem-solving approach.

If you are still not sure what the developer should know, check the official Python website to see which frameworks and libraries would fit into your project.



#Finding & hiring Python devs

A SOFTWARE HOUSE OR A FREELANCER

Which one would be a better solution for your project?

Best software development firms might cost more if you compare it with a regular freelancer but offer much more. Of course, you can also find a freelancer that will charge you more than a software house but this depends on the agreement you make. How to find balance?

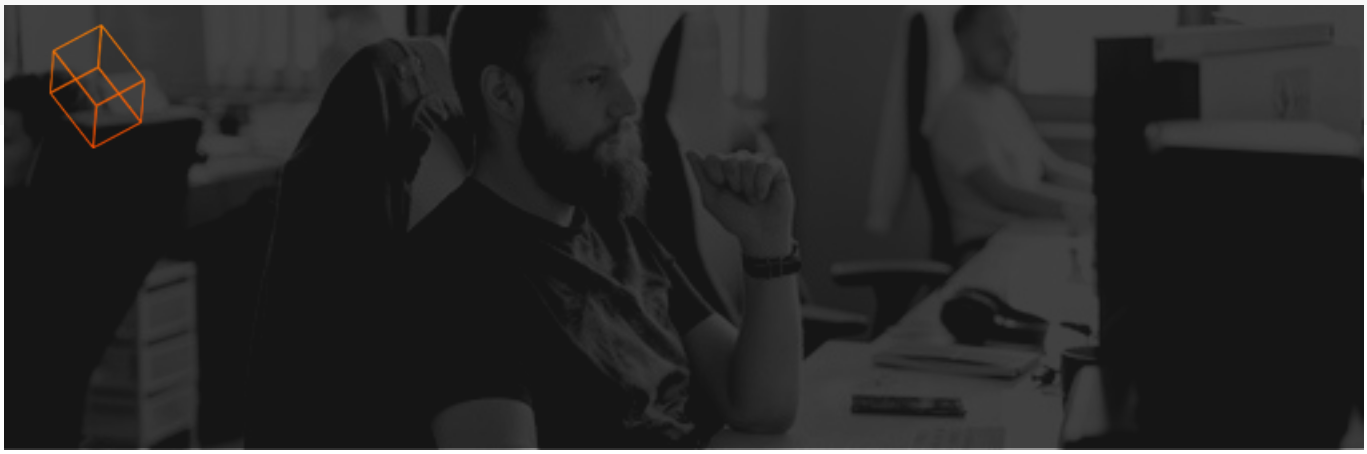
First of all, estimate the scale and lifespan of your project, then think about the time aspect and your budget

There are plenty of factors that you need to take into account before making up your mind.

On one hand, a software house will probably charge you more than a freelancer and it might take more time to close the deal but on the other, a freelancer might not be enough or simply could drop the task (or have a sick leave).

Additionally, depending on your agreement, you need to have in mind what to do when a bug appears or you need to customize some part of your project. This seems a lot to take care of, right? If your project is on a larger scale and your business depends on the outcome, consider hiring an outsourced software company.

If you don't know where to search for recommendations to find the Python dev dream team, try popular services like Clutch.co, AppFutura. When it comes to best software companies, it is enough to dig into recommendations, check the offer, talk with the sales team and start the project.



#Finding & hiring Python devs

A SUCCESS OF A WEB PROJECT

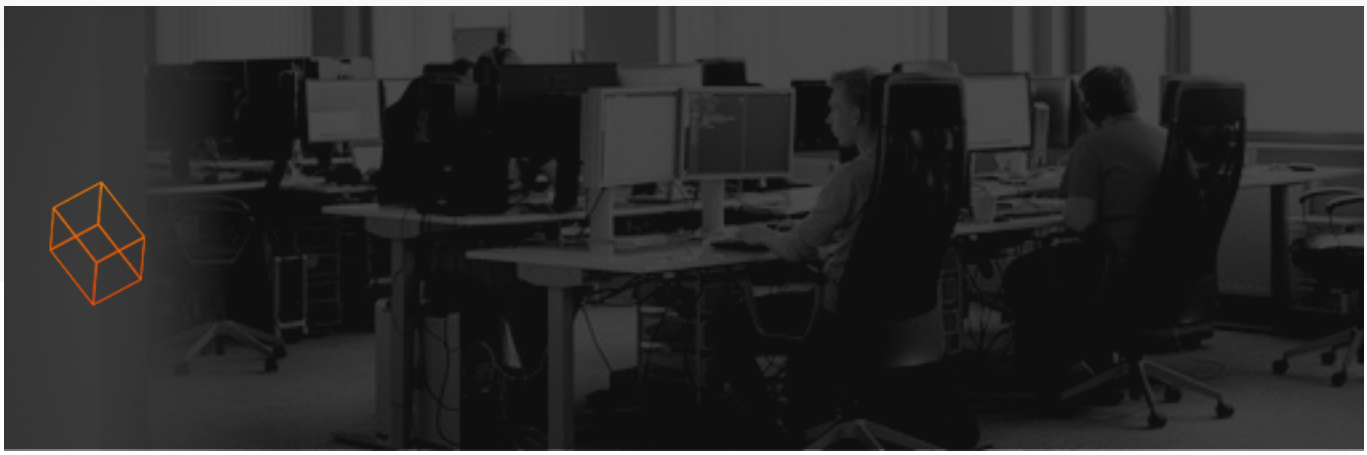
The final choice.

Web app development is a process and picking up the right people to work with is an extremely important part.

For small-scale projects or debugging something that you already have, a freelancer or in-house Python developer might be a better solution.

However, for larger ideas, where not only a developer is needed but **self-managed full-stack SCRUM-team** including a project manager, SCRUM master, UX designer, also a quality assurance team and so on - consider a software house. You can be sure that they will deal with software development process comprehensively and without delay.

There are pros and cons for each solution so think about it critically while the success of your project rests often in hands of the developer.



FINAL THOUGHTS

#Why Python

PYTHON: THE GO-TO LANGUAGE FOR ALL BUSINESS OWNERS

Python programming language is an interpreted, dynamically-typed, and high-level language that uses primary concepts of object-oriented programming. It has a unique syntax that differentiates it from other programming languages such as Java, C and C++.

Python also offers a wide array of frameworks such as Django, Flask, Pyramid, Falcon and many others. That's what makes it enrapture by its **simplicity, versatility, and rapidity of development**.

The reason why **Python is a good choice for both fast-growing start-ups and established corporate enterprises** is the fact that its rapid and robust development capabilities can be applied in projects of basically any size.

The world's largest information technology companies such as Facebook, Google, Spotify, Netflix, Instagram, IBM have successfully placed a bet on Python.

Python plays a considerable role in

- **prototyping,**
- **quick MVP**
- **POC development,**
- **application scalability and security**

Recently, it has been considered an informed choice for:

- **Machine Learning,**
- **Big Data,**
- **IoT**
- **AI solutions**