A Complete Guide to Web Development in Python

Apr 15, 2020 - 9 min read

Aaron Xie

In the past few years, **Python** has exploded in popularity. The programming language has surpassed Java in popularity, but, for many, this is no surprise. With the rise in machine learning, data analysis, and web application development, many developers utilize Python for its powerful and abundant libraries, easy-to-learn syntax, and portability.

Without a doubt, it's a great time to be a Python developer. Today, you will learn how Python is utilized in web development.

We will cover the following:

- What is web development?
- What makes Python suited for web development?
- Python web frameworks
- Python libraries for web development
- A roadmap for web development with Python
- How to create your first web application in Python
- Become a Python web developer

Learn how to develop web apps with Python.

Flask is simple and lightweight.If you’ve got some experience in Python, picking up Flask will be a breeze.

Flask: Develop Web Applications in Python (https://www.educative.io/courses/flask-develop-web-applications-in-python)

What is web development?

For those new to programming, you might be wondering what exactly web development is. Though it’s impossible to pinpoint an exact term, **web development** is loosely defined as building, creating, and maintaining websites. Typically, web development involves a frontend, everything that interacts with the client, and a backend, which contains business logic and interacts with a database.

Want a good overview of web development? Check out our Beginner's Guide to Web Development (https://www.educative.io/blog/beginner-guide-to-web-dev) article.

What makes Python suited for web development?

Advantages of developing web applications in Python

- **Easy to learn:** Python is the most popular language for first-time learners for a reason. The language relies on common expressions and whitespace, which allows you to write significantly less code compared to some other languages like Java or C++. Not only that, but it has a lower barrier of entry because it's comparatively more similar to your everyday language so you can easily understand the code.

- **Rich ecosystem and libraries:** Python offers a vast range of library tools and packages, which allows you to access much pre-written code, streamlining your application development time. For example, you have access to Numpy and Pandas for mathematical analysis, Pygal for charting, and SLQALchemy for composable queries. Python also offers amazing web frameworks like Django and Flask, which we'll dive into later in the article.

- **Fast prototyping:** Because Python takes significantly less time to build your projects compared to other programming languages, your ideas come to life a lot faster, allowing you to gain feedback and iterate quickly. This quick development time makes Python especially great for startups who can hit the market sooner to gain a competitive edge.

- **Wide-spread popularity:** Python is one of the most popular languages in the world, with communities from all over the world. Because of how popular the language is, Python is continuously updated with new features and libraries, while also providing excellent documentation and community support. Especially for new developers, Python provides extensive support and framework for one to begin their developer journey.

Python web frameworks

What are web frameworks, and why are they important?

Think of a toolbox. A **web framework** is a collection of packages and modules made up of pre-written, standardized code that supports the development of web applications, making development faster and easier, and your programs more reliable and scalable. In other words, frameworks already have built-in components that “set up” your project, so you have to do less grunt work.

Python web frameworks are only utilized in the backend for server-side technology, aiding in URL routing, HTTP requests and responses, accessing databases, and web security. While it is not required to use a web framework, it is extremely recommended because it helps you develop complex applications in significantly less time.

> ***Enjoying the article?*** *Scroll down to sign up (https://www.educative.io/blog/blog-newsletter-annoucement) for our free, bi-monthly newsletter.*

What are some popular Python web frameworks?

Django and Flask are, by a wide margin, the most popular Python web development frameworks

Django is a Python web framework that offers an open-source, high-level framework that “encourages rapid development and clean, pragmatic design.” It’s fast, secure, and scalable. Django offers strong community support and detailed documentation.

Django is incredibly flexible in which you can work with MVPs to larger companies. For some perspective, some of the largest companies that use Django are Instagram, Dropbox, Pinterest, and Spotify.

> *Read out tutorial on Django for more details: What is Django Python? Build your first program from scratch (https://www.educative.io/blog/what-is-django-python)*

Flask is considered a microframework, which is a minimalistic web framework. It’s less “batteries-included,” meaning that it lacks a lot of features and functionality that full-stack frameworks like Django offer, such as a web template engine, account authorization, and authentication.

Flask is minimalistic and lightweight, meaning that you add extensions and libraries that you need as you code without automatically being provided with it by the framework.

The philosophy behind Flask is that it gives only the components you need to build an app so that you have the flexibility and control.

Flask is also a prevalent and powerful web framework as it’s used by large companies like Netflix, Linkedin, and Uber.

> **Other notable frameworks:**
>
> - Pyramid
> - Web2Py
> - Turbogears

Which should you use?

So, you might ask yourself: what framework should I choose? The answer is that it depends. Consider your skill level as a web developer. If you are quite experienced, consider developing your program with something more “barebones.”

Whereas if you are a junior dev, it might be better to use a framework that provides more support like Django.

Also, ask yourself, would you prefer a “foundation” codebase to build off or have the flexibility to build the backbone of your codebase? If you prefer the first option, go with Django; if you prefer the second option, go with Flask.

At the end of the day, they can both achieve the same functionality, and it’s more important to start coding rather than fretting over which framework is better.

Python libraries for web development

Some useful Python libraries for web development to keep note of:

- If you ever need a web crawler to extract data for your application, **Scrapy** is great for that. It’s a widely used library for scraping, data mining, automated testing, and more.
- **Zappa** is a powerful library for developing a serverless application on AWS Lambda.
- **Requests** is a library that allows you to send HTTP requests easily, which is used to communicate with an application, allowing you to get HTML pages or data, for example.
- Another useful library is **Dash**, which helps those developing web applications that have to do with data visualization. Built on top of Flask, it offers features like charts, graphs, dashboards, and more.

A Roadmap for Web Development with Python

Step 1: HTML + CSS

When you're starting with web development, it's important that you first learn HTML (https://www.educative.io/blog/revamp-front-end-skills) and CSS (https://www.educative.io/blog/revamp-front-end-skills), which are the fundamentals of learning how to build websites. It would be best if you learned how to structure responsive static pages to start your web development journey. It might also be helpful to learn concepts like the internet, HTTP, browsers, DNS, hosting, and more.

You can also learn a CSS framework like Materialize or Bootstrap, which significantly speeds up your development, but it's not needed.

Step 2: Javascript

An excellent next step is to learn vanilla Javascript. You should learn basic concepts like data types, variables, general conventions, string manipulation, arithmetic and operators, control statements, loops, etc. Learning the basics of Javascript will make it easier for you when you're applying Javascript to the client-side code.

Develop Web Applications in Python with Flask

Educative's Python web dev course will help you pick up Flask with ease. This project-based web dev course will propel your career. Get started on Educative's personalized learning platform with no extra downloads.

Flask: Develop Web Applications in Python (https://www.educative.io/courses/flask-develop-web-applications-in-python)

Step 3: DOM & jQuery

After you learn javascript fundamentals, you should learn how to manipulate the DOM and jQuery, which is a javascript library that makes DOM manipulation easier. Now you know how to create dynamic pages.

Frontend Framework (optional)

While it's not required to learn a frontend framework like React to build a functional full-stack web application, it's recommended. Not only does it help you create beautiful SPAs, but it's also often a requirement to be hired as a frontend or full-stack developer.

Step 4: Python

Now, onto the backend. You should cover the basics of Python (https://www.educative.io/blog/how-to-learn-python-in-5-easy-steps) just like you did for Javascript before learning DOM manipulation. Learning the fundamentals will prepare you for Django, so you don't have as much head-ache jumping right in. Learning introductory Python shouldn't be too tricky, though, as many of the concepts will be similar to Javascript.

Step 5: Django + Database

With Django, you'll be able to set up your backend environment and develop the business logic. You will also have to learn about databases such as SQLite, how to make queries, and the CRUD function. With this, you can make a full-stack application!

How to create your first web application in Python

You will create a Flask ‘hello world’ app as an introduction to Python web frameworks

Installing Flask

The first thing you need to do is install Flask. Depending on what version of Python you are using, you will have to use `pip` or `pip3` .

```
pip install flask
```

Flask app

Next, create a filed called `hello.py` with the following code.

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
    def hello():
        return "Hello World!"

if __name__ == "__main__":
    app.run()
```

- **Line 1** imports Flask
- **Line 2** instantiates Flask with `app` variable, using the `__name__` attribute
- **Line 4** sets up a route `/` for your index, or default, page. When a user goes to `locahost:5000/` , you can set up particular code to be triggered
- **Line 5** creates a function called `hello`
- **Line 6** returns the string `Hello World!` to the user
- **Line 8** tests to make sure the right script is being run
- **Line 9** runs the application from the `app` variable we initialized so that the user can visit the web application by going to `localhost`

Become a Python web developer

Congrats! You’re well on your way to becoming a proficient web developer with Python. You should have a good overview of how Python operates within web development and its uses. With this understanding, you have many directions you can pursue when learning Python.

If you’re ready to start learning and get your hands dirty, we recommend Educative’s course **Flask: Develop Web Applications in Python. (https://www.educative.io/courses/flask-develop-web-applications-in-python)** This project-based course is a detailed guide to web application development using the increasingly popular Flask framework.

Continue reading about Python and web dev

Here’s a list of resources for you to learn more about web development with Python.

- Dynamic Programming Tutorial: making efficient programs in Python (https://www.educative.io/blog/python-dynamic-programming-tutorial)
- 50 Python Interview Questions and Answers (https://www.educative.io/blog/python-interview-questions)
- 3 Tips for Beginner Python Devs: advice from an industry expert (https://www.educative.io/blog/tips-beginner-python-developers)

WRITTEN BY

Aaron Xie

educative
(/)

Learn in-demand tech skills in half the time

LEARN

Courses
(/explore)

Early Access Courses
(/explore/early-access)

Edpresso
(/edpresso)

Blog
(/blog)

Pricing
(/unlimited)

Free Trial New
(/trial)

For Business
(/business)

CodingInterview.com
(//codinginterview.com/)

SCHOLARSHIPS

For Students
(/github-students)

For Educators
(/github-educators)

CONTRIBUTE

Become an Author
(/authors)

Become an Affiliate
(/affiliate)

LEGAL

Privacy Policy

(/privacy)

Terms of Service

(/terms)

Business Terms of Service

(/enterprise-terms)

MORE

Our Team

(/team)

Careers

(//angel.co/educativeinc/jobs)

For Bootcamps

(//try.educative.io/bootcamps)

Blog for Business

(/blog/enterprise)

Quality Commitment

(/quality)

FAQ

(/courses/educative-faq)

Press

(/press)

Contact Us

(/contactUs)

educativeinc)

(//linkedin.com/company/educative-inc/)

(//twitter.com/educativeinc)

(//www.youtube.com/channel/UCT_8FqzTIr2Q1BOtvX_DPPw/?sub_confirmation=1)

(//educativesess