Next Steps in Web Development

Learning Objectives

After this lesson, you will be able to:

- Identify common web concepts to dive into next.
- Identify common Python libraries used for front-end web development.
- Identify common Python-based tech stacks for back-end web development.

Web Development Is Pretty Cool. What's Next?

Web development is a huge field. Where should you go next?

There are a few options to explore:

- Front-end web development.
    - Code in Python to just host the site.
- Back-end with Python.
    - Code in Python

With one of those, you can also explore…

- Extending Flask.
- Other popular Python web framework libraries.

Path 1: Front-End Development?

Configuring the part of the website people see.

Why might you go this route?

- You’re design-inclined.
- You want to make cool sites!
- You’re using Python to accomplish the goal of propping up your designs.

Path 1: Front-End Development — What Next?

GA has some classes! (A shameless self-plug, but part-time Front-End Web Development is a good place to start.)

For your own study:

- Get better at HTML and CSS (*important*).
  - The core of front-end web development.
- Check out Bootstrap and Flexbox.
  - HTML/CSS frameworks.
- Check out JavaScript and JavaScript frameworks.
  - Add interactivity to your webpages.
  - React, Angular, jQuery, and Ember are all popular in different locations.

Path 1: Front-End Development — Then What?

Start building up a portfolio of projects.

- Make yourself a portfolio website to showcase your work!

Once you’re comfortable, try modifying premade Bootstrap themes.

Path 2: Back-End Development?

Handle the server side:

- Make websites work.
- Bring in data.

Why might you go this route?

- You love programming!
- You don’t care how the website looks.
- You just want to write Python.
- The idea of using someone else’s designs sounds great.

Path 2: Back-End Development — What Next?

GA has some classes! (Another shameless self plug. The full-time Web Development Immersive will give you a strong foundation in back-end essentials.)

For your own study:

- Learn Git version control (*important*).
  - Learn GitHub or another version control system.
- Check out connecting to databases.
- Learn SQL.
- Learn about SQLAlchemy and PyMongo.
- Learn about servers like WSGI and NGINX.

Path 2: Back-End Development — Then What?

Start a portfolio (preferably on GitHub or another version control system).

Once you’re comfortable:

- Try starting a MongoDB yourself, then having Flask retrieve data from there.

Knowledge Check: I want to...

… make a beautiful website.

All this Python coding is a bit much for me, but I like websites.

Which should I look into?

- Front-end web development
- HTML/CSS/JavaScript
- Flask extensions
- Back-end web development

Which do you think?

Solution: I want to...

… make a beautiful website.

All this Python coding is a bit much for me, but I like websites.

Which should I look into?

- Front-end web development
- HTML/CSS/JavaScript

Path(ish) 3: Expanding on Flask

This is not really a path! But you can do it in conjunction with either of the above.

Look into integrating these with Flask:

- Bootstrap
- SQLAlchemy/PyMongo

Bootstrap

Front-end or Flask? This slide's for you.

Like awesome-looking websites?

Don't like typing out HTML and CSS?

Bootstrap is for you!

- It's a free and open-source front-end framework.
- Handles all of the "pretty" stuff.

Bootstrap is great for:

- Prospective front-end developers.
  - Extremely popular in many companies.
- Integrating with Flask!
  - Bootstrap makes a site pretty, and Flask makes it work.

How to Get Started With Bootstrap and Flask

Here's how to get it:

- `pip install Flask-Bootstrap`.
- `git clone https://github.com/mbr/flask-bootstrap.git`.
- `cd` into the `flask-bootstrap` directory.
- Run `pip install -r sample_app/requirements.txt` to install the required packages.

And here's how to run it:

- Deploy Flask with Bootstrap by running `flask --app=sample_app dev`.
- Go through directories.
  - Comment out code to be certain you know exactly what every line is doing.

Here's where you can read the docs: https://getbootstrap.com/docs/3.3/getting-started/.

SQLAlchemy and PyMongo

Back-end or Flask? This slide's for you.

We can't always connect to a list!

- Most information is stored in databases (like a spreadsheet).

SQL is *the* language for working with databases.

Knowing a database is also crucial! Popular databases include:

- MySQL
- MongoDB
- PostgresQL

Some Python libraries to look into include:

- SQLAlchemy
- PyMongo

How to Get Started With PyMongo

This slide's long! It's for your later reference.

How do I use Flask and databases?

- Install Flask SQLAlchemy with `pip install Flask-SqlAlchemy`.
- Install Flask PyMongo with `pip install Flask-PyMongo`.

What about back-end database with no Flask?

- Install just PyMongo with `python -m pip install pymongo`.

Either way, you need a database. Install MongoDB:

- `brew update`
- `brew install mongodb`
- `brew install mongodb --devel`

How to Get Started With PyMongo

Start your MongoDB instance:

- `mkdir -p /data/db`
    - Note: If that results in an error, you need different privileges.
    - Run: `sudo mkdir -p /data/db`.
- Start your database with the Mongo daemon with the command `mongod`.

How to Get Started With PyMongo

Then, add some data to your database. Save the below to a Python file and then try making `GET` and `POST` requests!

```
from flask import Flask, jsonify, request
from flask_pymongo import PyMongo


app = Flask(__name__)

app.config['MONGO_DBNAME'] = 'burritodb'
app.config['MONGO_URI'] = 'mongodb://localhost:27017/burritodb'


mongo = PyMongo(app)


@app.route('/burrito', methods=['GET'])
```

Knowledge Check: I want to...

…make a beautiful database.

All this Python coding is exciting!

Which of these should I look into?

- Bootstrap
- Back-end web development
- SQLAlchemy
- MySQL
- PostgreSQL
- MongoDB
- HTML and CSS

Which do you think?

Solution: I want to...

…make a beautiful database.

All this Python coding is exciting!

Which of these should I look into?

- Back-end web development
- SQLAlchemy
- MySQL
- PostgreSQL
- MongoDB

Path(ish) 4: Web Frameworks That Aren't Flask

Flask isn't the only Python web framework.

There are also:

- Pyramid
- Zope
- Bottle
- CherryPy
- **Django**
- **Pelican**

What's Django?

- A complete, "batteries-included" approach philosophy.
  - Conversely, Flask is lightweight; you add pieces on as you need them.
- A more secure framework.
- Highly scalable — think Disqus and Instagram.
- Get started at djangoproject.com.

What's Pelican?

- A Python library for static sites.
  - Blogs don't need to change beyond the text!
- Generates the HTML for you.
- You just need Python code and text!

This is a really quick way to get a site up and running.

```
pip install pelican
```

Web Development in General

There's so much more!

Mozilla (the company that makes Firefox) is an amazing resource.

- developer.mozilla.org/

Knowledge Check: I want to...

… make a blog.

I want to build simple, static websites in less than 10 minutes and then work on content.

Which of these should I look into?

- Pelican
- Databases
- Bootstrap
- Django

Which do you think?

Solution: I want to...

…make a blog.

I want to build simple, static websites in less than 10 minutes and then work on content.

Which of these should I look into?

- Pelican!

Knowledge Check: I want to...

…make an awesome Flask-based website.

I really liked all the above and am right at home between front-end and back-end with Flask.

Which of these should I look into?

- More Flask extensions
- Bootstrap
- PostgresSQL
- HTML and CSS
- Pelican
- Django

Which do you think?

Solution: I want to...

…make an awesome Flask-based website.

I really liked all of the above and am right at home between front- and back-end with Flask.

Which of these should I look into?

- More Flask extensions
- Bootstrap
- PostgreSQL

Knowledge Check: I want to...

Make a website, but without all these libraries and extensions

I really the idea of this whole framework thing, but I wish there were only one, obvious way to do things.

Which of these should I look into?

- More Flask extensions
- Bootstrap
- PostgreSQL
- HTML / CSS
- Pelican
- Django

Solution: I want to...

… make a website, but without all these libraries and extension.s

I really the idea of this whole framework thing, but I wish there were only one obvious way to do things.

Which of these should I look into?

- Django

Summary

There are a lot of ways to go!

- Extending Flask.
- Python front-end libraries.
- Python back-end libraries for databases.
- A few other Python-based web frameworks.
- General front-end or back-end web development.

Additional Reading

- Explore Flask
- PyMongo Documentation
- SQLAlchemy's Fantastic Object Relational Mapper Tutorial
- Django Documentation
- Pelican Documentation
- The Flask Docs
- Beginner Git Tutorial