

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Build Awesome Websites With Flask Python

Christopher W. Pitts

July 12, 2017

Disclaimer

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

The views in this presentation are entirely my own, and *in no way* represent any sort of official or unofficial endorsement by Sandia National Laboratories or NTESS.

About Me

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

- From Albuquerque, New Mexico
- Raging *Star Wars* fan
 - Still bitter about Disney and the Expanded Universe
- Studied computer science at Brigham Young University
- Software Systems Engineer at Sandia National Laboratories
- Ik spreek Hollands (ook Vlaams)!
- Happily married

Core Assumptions

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

In order to maximize your takeaway from this presentation, there are a few things I've assumed about you:

Core Assumptions

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

In order to maximize your takeaway from this presentation, there are a few things I've assumed about you:

- Familiarity with Python

Core Assumptions

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

In order to maximize your takeaway from this presentation, there are a few things I've assumed about you:

- Familiarity with Python
- Understanding of basic HTTP concepts
 - “Verbs”: GET, POST, etc.
 - Endpoints
 - Return codes: 200, 404, 418, etc.

Overview

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

- 1 Intro
- 2 Overview
- 3 Why Python?
- 4 Why Flask?
- 5 A Basic App
- 6 Flask Goodies
- 7 From Dev To Prod

Why Choose Python For Web Development?

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Not the obvious first choice

- Apache/NGINX/etc. written in C/C++
- Interpreted, not compiled
- Semantic whitespace

The Pros Of Python

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

- Rapid development
 - It's easier to get going in Python
 - Fewer initial stumbling blocks (although arguably more later)
- Easy deployment
 - Containerizing is easy with virtual environments
- Lots of community support
 - Easily one of the most well-supported communities

Why Choose Flask?

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Some nice things about Flask:

- Easy endpoint definition
`@app.route("/your/path/here")`
- Page templates
Jinja2 templating engine
- Pythonic, programmatic definition of an API
- Parameterization of endpoints
`@app.route("/path/with/<param>")`
`def route_handler(param):`
 `# Do stuff`

Why Choose Flask?

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Some things that use/have used Flask:

- Pinterest
- Former U.S. president Barack Obama's reelection campaign
- HTTPBin (httpbin.org)
- Twilio (REST API)
- Lyft (backend)

GET Off The Ground

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Let's take a look at a basic Flask web app!

GET Off The Ground

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

```
"""  
Initializes a Flask object and stores  
it in the 'app' variable  
"""  
app = Flask(__name__)  
  
# Defines a 'route' or endpoint  
@app.route("/")  
def index():  
    return render_template("index.html")  
  
# Run the app  
if __name__ == "__main__":  
    app.run(debug=True,  
            port=5050,  
            host="0.0.0.0")
```

Going POSTal

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

What if we wanted to allow the user to submit data to the website?

Going POSTal

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

```
app = Flask(__name__)
```

```
"""
```

*Notice the additional argument in the
app.route decorator*

```
"""
```

```
@app.route("/post", methods=["POST"])
```

```
def handle_post():
```

```
    data = dict(request.get_json())
```

```
    # Multivalue return sets return code
```

```
    return jsonify(data), 201
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True,
```

```
            port=5050,
```

```
            host="0.0.0.0")
```

Some Supporting Tools

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

A powerful stack

- Linux
- NGINX
- MariaDB
- Python

Other good tools

- Supervisor

pip install

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Useful Python Libraries

- flask_moment
- MySQL-python
- sqlalchemy
- uwsgi

PUTting Things Together

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

How about updating a resource?

PUTting Things Together

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

```
app = Flask(__name__)

@app.route("/put", methods=["PUT"])
def handle_put():
    data = dict(request.get_json())
    """
    Update the data or something awesome
    like that
    """
    return jsonify(data), 204

if __name__ == "__main__":
    app.run(debug=True,
            port=5050,
            host="0.0.0.0")
```

DELETE Puns Are Hard

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Deleting a resource?

DELETE Puns Are Hard

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

```
app = Flask(__name__)
```

```
@app.route("/delete", methods=["DELETE"])
```

```
def handle_delete():
```

```
    data = dict(request.get_json())
```

```
    # Delete the resource
```

```
    return jsonify(data)
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True,
```

```
            port=5050,
```

```
            host="0.0.0.0")
```

Multitasking Endpoints

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Thankfully, you don't have to define one method for each HTTP verb.

Multitasking Endpoints

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

```
# All the methods!
```

```
@app.route("/resource",  
           methods=["GET",  
                   "POST",  
                   "PUT",  
                   "DELETE"])
```

```
def handle_request():  
    if request.method == "GET":  
        pass # Return a resource  
    elif request.method == "POST":  
        pass # Create a resource  
    elif request.method == "PUT":  
        pass # Update a resource  
    elif request.method == "DELETE":  
        pass # Delete a resource
```

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

What if we wanted to create dynamic content server-side?

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Let's look at a Jinja2 template!

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

```
<!--Variables are enclosed in {{}}-->  
<h1>{{var}}</h1>
```

```
<!--Flow control and loops are in {%}%-->  
{% if var2 %}  
<h2>{{var2}}</h2>  
{% endif %}  
<ul>  
  {% for item in listVar %}  
  <!--Support for Python objects-->  
  <li><p>{{item.name}}</p></li>  
  {% endfor %}  
</ul>
```

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

A few notes

- Written directly into the HTML file
- Explicit ending of loops and flow control
- Awesome support for Python objects
- Jinja2 is not an HTML-specific format
 - Lots of discussion on the internet about automating report generation with Jinja2, Python, and \LaTeX

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Passing variables to a template:

```
@app.route("/template")
def templates_are_awesome():
    var="I replaced {{var1}}!"
    return render_template("variables.html",
                           var1=var,
                           listVar=["Item 1",
                                    "Item 2",
                                    "Item 3"])
```

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

The template:

```
<p>{{var1}}</p>
<ul>
  {% for item in var2 %}
  <li>{{item}}</li>
  {% endfor %}
</ul>
```

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

The finished product:

```
<p>I replaced {{var1}}!</p>
```

```
<ul>
```

```
  <li>Item 1</li>
```

```
  <li>Item 2</li>
```

```
  <li>Item 3</li>
```

```
</ul>
```

Page Templates

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Jinja2 also has support for:

- Dictionaries
- Custom objects
- Autoescaping strings
- Unicode (is built in)

Error Handlers

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Have you ever wanted a custom error page?

Error Handlers

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Flask has support for that!

```
@app.errorhandler(404)
def page_not_found(err):
    return render_template("404.html")
```

From Dev To Prod

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Playing in a dev sandbox is fun, but at some point in the future we'd like to push this app of ours to production.

Flask's Dev Server IS NOT STABLE

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

This is by design, you should use a production web server.
I recommend NGINX. Minimal configuration and good
performance.

And readable config files!

./run_forever?

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

No. Gross.

`nohup server.py && exit?`

No. Still gross.

Use Supervisor to keep the application running after logout.

Gluing Flask To NGINX

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

UWSGI

- Defined in PEP 333/3333
- Middleware between application and server

Questions?

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

References

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Flask

<https://flask.pocoo.org>

<https://www.fullstackpython.com/flask.html>

Python

<https://docs.python.org/3>

MariaDB

<https://mariadb.org>

NGINX

<https://nginx.org/en>

Supervisor

<http://supervisord.org>

References

Build
Awesome
Websites With
Flask Python

Christopher
W. Pitts

Intro

Overview

Why Python?

Why Flask?

A Basic App

Flask Goodies

From Dev To
Prod

Linux

<https://linux.org>

<http://tldp.org>

L^AT_EX

<https://latex.org>

<https://latex-project.org>