

An Introduction to Developing Web Applications with Flask

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources



[HackRice?] Workshop
Prudhvi Boyapalli

January 13, 2017

Outline

Introduction

Flask
Setup

Hello World

Basic Example
Closer Look
Variable Routing

Conclusion

Resources

① Introduction

Flask

Setup

② Hello World

Basic Example

Closer Look

Variable Routing

③ Conclusion

Resources

How does a web app work? What does Flask do?

Introduction

Flask

Setup

Hello World

Basic Example

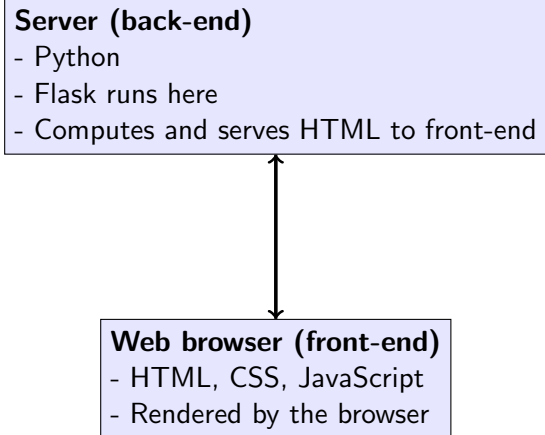
Closer Look

Variable Routing

Conclusion

Resources

Flask is a Python microframework for building web applications.



Install Python

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

- Basic level of Python knowledge is needed for this tutorial
 - If you've taken COMP140 you're in good shape
- Install Python if you haven't already: python.org
- I'm using 2.7, Flask also supports 3.3+

Install Flask

- Install Flask using PIP
 - **Pip** Installs **P**ackages from the **P**ython **P**ackage **I**ndex a.k.a. PyPI

Flask Installation

```
> sudo pip install Flask
```

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

Editing

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

An IDE is overkill for this workshop. Use a programmer's editor that you're comfortable with:

- Vim (All platforms)
- Atom (All platforms)
- Visual Studio Code (All platforms)
- Sublime Text (All platforms)
- Notepad++ (Windows)
- TextMate (OSX)
- Whatever floats your boat

Debugging

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

You'll be testing and debugging your code from the browser, so use one with good developer tools:

- Chrome
- Firefox

Other browsers can be used, but Chrome or Firefox will minimize the chance of running into browser specific issues.

Super basic hello world example

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

This is the bare minimum code needed on the back-end to serve a web page

```
./code/hello_world_example/hello_world.py
```

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

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

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


Running the web app

Introduction

Flask
Setup

Hello World

Basic Example
Closer Look
Variable Routing

Conclusion

Resources

Once we've written the flask code, we can run the server locally and view the results in our web browser

Execute these commands to run the code:

```
> cd /directory/with/project  
> export FLASK_APP=hello_world.py  
> flask run
```

Then you'll see something like:

```
* Serving Flask app "hello_world"  
* Running on http://127.0.0.1:5000/  
(Press CTRL+C to quit)
```

Running the web app

Introduction

Flask
Setup

Hello World

Basic Example
Closer Look
Variable Routing

Conclusion

Resources

Once we've written the flask code, we can run the server locally and view the results in our web browser

Execute these commands to run the code:

```
> cd /directory/with/project  
> export FLASK_APP=hello_world.py  
> flask run
```

Then you'll see something like:

```
* Serving Flask app "hello_world"  
* Running on http://127.0.0.1:5000/  
(Press CTRL+C to quit)
```

How did that work?

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

- ❶ We started up a local flask server which is listening at IP address 127.0.0.1, port 5000
- ❷ We visited this page and the browser sent a request to our flask server
- ❸ Because we visited 127.0.0.1:5000/ (emphasis on the "/"), the method bound to "/" was invoked
- ❹ The `hello()` method in `hello_world.py` was executed and the return value is sent to the browser as HTML
- ❺ The browser received the HTML and rendered the webpage

How did that work?

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

- ① We started up a local flask server which is listening at IP address 127.0.0.1, port 5000
- ② We visited this page and the browser sent a request to our flask server
- ③ Because we visited 127.0.0.1:5000/ (emphasis on the "/"), the method bound to "/" was invoked
- ④ The `hello()` method in `hello_world.py` was executed and the return value is sent to the browser as HTML
- ⑤ The browser received the HTML and rendered the webpage

How did that work?

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

- ❶ We started up a local flask server which is listening at IP address 127.0.0.1, port 5000
- ❷ We visited this page and the browser sent a request to our flask server
- ❸ Because we visited 127.0.0.1:5000/ (emphasis on the "/"), the method bound to "/" was invoked
- ❹ The `hello()` method in `hello_world.py` was executed and the return value is sent to the browser as HTML
- ❺ The browser received the HTML and rendered the webpage

How did that work?

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

- ① We started up a local flask server which is listening at IP address 127.0.0.1, port 5000
- ② We visited this page and the browser sent a request to our flask server
- ③ Because we visited 127.0.0.1:5000/ (emphasis on the "/"), the method bound to "/" was invoked
- ④ The `hello()` method in `hello_world.py` was executed and the return value is sent to the browser as HTML
- ⑤ The browser received the HTML and rendered the webpage

How did that work?

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

- ① We started up a local flask server which is listening at IP address 127.0.0.1, port 5000
- ② We visited this page and the browser sent a request to our flask server
- ③ Because we visited 127.0.0.1:5000/ (emphasis on the "/"), the method bound to "/" was invoked
- ④ The `hello()` method in `hello_world.py` was executed and the return value is sent to the browser as HTML
- ⑤ The browser received the HTML and rendered the webpage

Interactive hello world with variable routing

Introduction

Flask
Setup

Hello World

Basic Example
Closer Look
Variable Routing

Conclusion

Resources

Now that we have a basic idea of how flask works, we can do some cool stuff with routing:

```
./code/hello_world_example/hello_user.py
```

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

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

@app.route("/user/<name>")
def hello_user(name):
    return "Hello %s, how are you today?" % name

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


Running the new and improved web app

Introduction

Flask

Setup

Hello World

Basic Example

Closer Look

Variable Routing

Conclusion

Resources

As before:

```
> export FLASK_APP=hello_user.py
> flask run
* Serving Flask app "hello_world"
* Running on http://127.0.0.1:5000/
(Press CTRL+C to quit)
```

Hello World

```
http://127.0.0.1:5000/world
```

Hello <user>

```
http://127.0.0.1:5000/user/Foo
```

```
http://127.0.0.1:5000/user/Bar
```

Resources

Introduction

Flask
Setup

Hello World

Basic Example
Closer Look
Variable Routing

Conclusion

Resources

- **This presentation & and all of the code**
`github.com/hack-rice/flask-tutorial`
- **Flask website & documentation**
`flask.pocoo.org`