

Programming with Python

Lesson 6: Flask

December 6th, 2016

Last week's goals

Last week's goals

- We learnt about tuples and their uses in returning multiple values

Last week's goals

- We learnt about tuples and their uses in returning multiple values
- We learnt about dictionaries

Last week's goals

- We learnt about tuples and their uses in returning multiple values
- We learnt about dictionaries
- We discussed objects

Last week's goals

- We learnt about tuples and their uses in returning multiple values
- We learnt about dictionaries
- We discussed objects
- We finished writing our own text based game!

What is a web server?

What is a web server?

A web server is

What is a web server?

A web server is something which **serves** you things.

What is a web server?

A web server is something which **serves** you things.
When you go on a website, you, the client, ask a server for something.

What is a web server?

A web server is something which **serves** you things.
When you go on a website, you, the client, ask a server for something. This file could be an html document, an image or something completely different.

HTTP requests

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

GET

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

GET gets you data from a location/service

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

GET gets you data from a location/service

POST

HTTP requests

Specifically, you send the server an HTTP request. What is HTTP?
HTTP - HyperTextTransferProtocol - The protocol which the world wide web relies on

There are many types of HTTP request. The most common two used are:

- GET gets you data from a location/service
- POST sends your data to a location/service

A little more on HTTP

A little more on HTTP

Status code responses

A little more on HTTP

Status code responses

- 200

A little more on HTTP

Status code responses

- 200
- 304

A little more on HTTP

Status code responses

- 200
- 304
- 400

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401
- 403

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401
- 403
- 404

A little more on HTTP

Status code responses

- 200
- 304
- 400
- 401
- 403
- 404
- 500

APIs

Any website you go on, you will have requested, using GET, the web page from a server somewhere. Any website is an example of a web server.

APIs

Any website you go on, you will have requested, using GET, the web page from a server somewhere. Any website is an example of a web server.

APIs are also examples of web servers.

APIs

Any website you go on, you will have requested, using GET, the web page from a server somewhere. Any website is an example of a web server.

APIs are also examples of web servers.

They serve over the HTTP protocol, but instead of serving you a website, they serve information (or take in information and do something neat with it).

An example

`https://pokeapi.co/`

Python in action: Flask

Flask is a web framework, written in python allowing us to write web apps in python.

Python in action: Flask

Flask is a web framework, written in python allowing us to write web apps in python.

We can use it to write interactive websites or neat web services.

Getting started with flask - Setting up our environment

We first need pip. To install pip, i'd suggest googling it. It can be tricky to install (though if you're ubuntu based, apt-get install python3-pip is all you need to do ;)

Getting started with flask - Setting up our environment

We first need pip. To install pip, i'd suggest googling it. It can be tricky to install (though if you're ubuntu based, apt-get install python3-pip is all you need to do ;)

To install flask - *pip install flask*

Getting started with flask - Setting up our environment

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/ root directory of code

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/ root directory of code

my-awesome-flask-app/templates/

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/ root directory of code
my-awesome-flask-app/templates/ directory of html to be rendered

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/ root directory of code
my-awesome-flask-app/templates/ directory of html to be rendered
my-awesome-flask-app/public/

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/	root directory of code
my-awesome-flask-app/templates/	directory of html to be rendered
my-awesome-flask-app/public/	directory of static files e.g. css & js

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/ root directory of code

my-awesome-flask-app/templates/ directory of html to be rendered

my-awesome-flask-app/public/ directory of static files e.g. css & js

my-awesome-flask-app/app.py

Getting started with flask - Setting up our environment

Folder structure:

my-awesome-flask-app/	root directory of code
my-awesome-flask-app/templates/	directory of html to be rendered
my-awesome-flask-app/public/	directory of static files e.g. css & js
my-awesome-flask-app/app.py	where our code will start

Hello World!

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

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

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

Hello World!

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

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

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

Hello World!

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

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

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


Hello World!

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

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

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

Hello World!

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

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

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

Endpoints

An endpoint is a *destination*. When you go to 127.0.0.1:5000 you are going to the root endpoint `/`.

Endpoints

An endpoint is a *destination*. When you go to 127.0.0.1:5000 you are going to the root endpoint `/`.

If instead you go to 127.0.0.1:5000/hello/world, you are going to the endpoint `/hello/world`.

Templates

Templates are pretty much HTML++.

Templates

Templates are pretty much HTML++.

They are pre-rendered on the server side and can be passed in values.

Jinja2

Flask, by default, uses the Jinja2 templating engine. It's really powerful and has some neat features.

Putting everything together

That's all for tonight

To summarise:

That's all for tonight

To summarise:

- We learnt about flask and it's ability to serve us

That's all for tonight

To summarise:

- We learnt about flask and it's ability to serve us
- We learnt about pip and how it can help us manage packages

That's all for tonight

To summarise:

- We learnt about flask and it's ability to serve us
- We learnt about pip and how it can help us manage packages
- We learnt about templates and how they can render stuff for us

That's all for tonight

To summarise:

- We learnt about flask and it's ability to serve us
- We learnt about pip and how it can help us manage packages
- We learnt about templates and how they can render stuff for us
- We made a neat album search tool!

THAT IS ALL

I hope you all enjoyed our experience going through python.
Next semester, Amy is going to be teaching much, much more on
web apps through a powerful web language, ruby on rails.

Hope you all have a good holiday!