



Intro to Flask

Learning Objectives:

After this lesson, you will be able to:

- Write a basic Flask application.

Discussion: Commonalities

What do you think these websites have in common?

- [Pinterest](#)
- [Instagram](#)
- [LinkedIn](#)

They're each:

- High on user interactivity.
- Handling a large server load.

What else?

They All Use Flask



Some quick notes about Flask:

- It's a Python micro web framework.
- It can create and write the entire back-end in Python!
- It can do small tasks (e.g., create a microblog or stand up a simple API).
- It can do complex tasks (e.g., Pinterest's API or create a Twitter clone).

Flask Syntax

How?

We just make a normal Python app.

It looks like:

```
# Import Flask class from flask library. (Note the upper/lowercase convention)
from flask import Flask

# Initialize an instance of the Flask class.
# This starts the website!
app = Flask(__name__)

# The default URL ends in / ("my-website.com/").
# Could be instead "my-website.com/about" or anything - more on this later.
@app.route('/')

```

We Do: Let's Try!

We'll run the Flask app like any other app.

- We need to install Flask!
 - `pip install flask`

Create a file called `my_website.py`.

Start with:

```
# Import Flask class from flask library.  
from flask import Flask
```

We Do: The Main Flask App

Let's add:

```
# Initialize an instance of the Flask class.
# This starts the website!
app = Flask(__name__)

# The default URL ends in / ("my-website.com/").
@app.route('/')

# Function that returns the page: Display "Hello, World!"
def index():
    return 'Hello, World!'
```

We Do: Flask App — Try it!

Run the app like normal:

```
python my_website.py
```

Go to:

```
http://localhost:5000/
```

You made a web app!

Let's change the string:

```
def index():  
    # The "return" determines what's displayed.  
    return 'Hello, World!'
```


I Do: Displaying the App

It's just Python — we can write any code.

- But `return` essentially just takes strings.

```
def index():  
    my_list = ["Hey", "check", "this", "out"]  
    return my_list[0] # Works!
```

Conversely:

```
def index():  
    my_list = ["Hey", "check", "this", "out"]  
    return my_list # WON'T WORK
```

We Do: Flask Variations

`app` and `index` are just naming conventions.

- `def index():` could be `def monkey():`.
- `app` could be `guitar`.
 - Be sure to change it in all places!

But, naming variables sensibly is important!

```
from flask import Flask

guitar = Flask(__name__)
@guitar.route('/')

def monkey():
    return 'Hello, World!'

if __name__ == '__main__':
    guitar.run(debug=True)
```

Flask History

Let's back up. Where did Flask come from?

- Before 2010:
 - No easy method for Python websites.
- 2010:
 - A few developers built Flask to fix this.

Flask is built on two libraries:

- *Werkzeug*:
 - Interfaces with the web.
 - Helps handle request and connections.
- *Jinja*:
 - We'll be using this later!
 - We can write templates for all pages across our web app.

Summary: Flask

- A Python micro web framework
- Developed in 2010

Looks like this:

```
# Import Flask class from flask library.
from flask import Flask

# Initialize an instance of the Flask class.
app = Flask(__name__)

# The default URL ends in / ("my-website.com/").
@app.route('/')

# Function that returns the page: Display "Hello, World!"
def index():
```

Additional Reading

- [Flask Documentation](#)
- [A Flask Tutorial to Follow Along With](#)
- [The Flask Mega-Tutorial](#)
- [A Great Guide to Those Weird “Decorators”](#)