**Using Flask in the Classroom Workspace**

In the next part of the lesson, you'll see a classroom workspace. The classroom workspace already has Flask set up for you. So for now, all you need to do to run the Flask app is to open a Terminal and type.

python worldbank.py

That assumes you are in the default workspace directory within Terminal. That will get the server running.

**Seeing your App in the Workspace**

Once the server is running, open a new terminal window and type

env | grep WORK

This command will return the Linux environmental variables that contain information about your classroom workspace. The env command will list all the environmental variables. The | symbol is a pipe for sending output from one command to another. The grep command searches text, so grep WORK will search for any text containing the word WORK.

The command should return two variables:

WORKSPACEDOMAIN=udacity-student-workspaces.com

WORKSPACEID=viewc7f3319f2

Your WORKSPACEID variable will be different but the WORKSPACEDOMAIN should be the same. Now, open a new web browser window, and type the following in the address bar:

http:*//WORKSPACEID-3001.WORKSPACEDOMAIN*

In this example, that would be: [**https://viewc7f3319f2-3001.udacity-student-workspaces.com/**](https://viewc7f3319f2-3001.udacity-student-workspaces.com/)

DON'T FORGET TO INCLUDE -3001. You should be able to see the web app. The number 3001 represents the port for accessing your web app.

**Creating New Pages**

To create a new web page, you first need to specify the route in the routes.py as well as the name of the html template.

@app.route('/new-route')

**def** **render\_the\_route**():

**return** render\_template('new\_route.html')

The route name, function name, and template name do not have to match; however, it's good practice to make them similar so that the code is easier to follow.

The new\_route.html file must go in the templates folder. Flask automatically looks for html files in the templates folder.

**What is @app.route?**

To use Flask, you don't necessarily need to know what @app.route is doing. You only have to remember that the path you place inside of @app.route() will be the web address. And then the function you write below @app.route is used to render the correct html template file for the web address.

In Python, the @ symbol is used for decorators. Decorators are a shorthand way to input a function into another function. Take a look at this code. Python allows you to use a function as an input to another function:

**def** **decorator**(input\_function):

**return** input\_function

**def** **input\_function**():

print("I am an input function")

decorator\_example = decorator(input\_function)

decorator\_example()

Running this code will print the string:

*I am an input function*

Decorators provide a short-hand way of getting the same behavior:

**def** **decorator**(input\_function):

print("Decorator function")

**return** input\_function

@decorator

**def** **input\_function**():

print("I am an input function")

input\_function()

This code will print out:

*Decorator function*  
*I am an input function*

Instead of using a decorator function, you could get the same behavior with the following code:

input\_function = decorator(input\_function)

input\_function()

Because @app.route() has the . symbol, there's an implication that app is a class (or an instance of a class) and route is a method of that class. Hence a function written underneath @app.route() is going to get passed into the route method. The purpose of @app.route() is to make sure the correct web address gets associated with the correct html template. This code

@app.route('/homepage')

**def** **some\_function**()

**return** render\_template('index.html')

is ensuring that the web address '[**www.website.com/homepage`**](http://www.website.com/homepage%60) is associated with the index.html template.

If you'd like to know more details about decorators and how @app.route() works, check out these tutorials:

* [**how @app.route works**](https://ains.co/blog/things-which-arent-magic-flask-part-1.html)
* [**general decorators tutorial**](https://realpython.com/primer-on-python-decorators/)