9.5.2

Create the Welcome Route

Our first route will be one of the most important. We need to ensure that our investors can easily access all of our analysis, so our welcome route will essentially be the entryway to the rest of our analysis.

Our first task when creating a Flask route is to define what our route will be. We want our welcome route to be the **root,** which in our case is essentially the homepage.

REWIND

To understand routes, remember the Google example we used earlier. If you google "surfer," for example, you'll see search options for images, videos, news, maps, and more. These are all the different "routes" you can take, and the Google homepage is essentially the root.

IMPORTANT

All of your routes should go after the `app = Flask(__name__)` line of code. Otherwise, your code may not run properly.

We can define the welcome route using the code below:

```
@app.route("/")
```

Now our root, or welcome route, is set up. The next step is to add the routing information for each of the other routes. For this we'll create a function, and our return statement will have f-strings as a reference to all of the other routes. This will ensure our investors know where to go to view the results of our data.

First, create a function `welcome()` with a return statement. Add this line to your code:

```
def welcome():
    return
```

Next, add the precipitation, stations, tobs, and temp routes that we'll need for this module into our return statement. We'll use f-strings to display them for our investors:

```
def welcome():
    return(
    '''
    Welcome to the Climate Analysis API!
    Available Routes:
    /api/v1.0/precipitation
    /api/v1.0/stations
    /api/v1.0/tobs
    /api/v1.0/temp/start/end
    ''')
```

NOTE

When creating routes, we follow the naming convention `/api/v1.0/` followed by the name of the route. This convention signifies that this is version 1 of our application. This line can be updated to support future versions of the app as well.

The welcome route is now defined, so let's try to run our code. You can run Flask applications by using the command below, but you'll need a web browser to view the results.

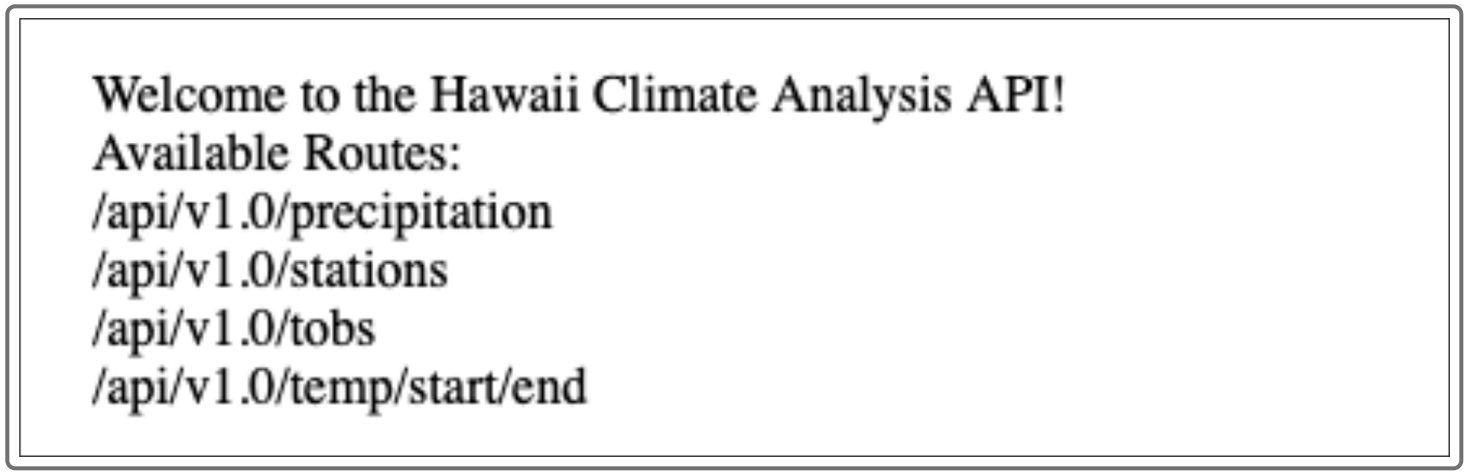
Let's start by using the command line to navigate to your project folder. Then run your code:

```
flask run
```

After starting the flask application, you'll likely see more text output than you have so far. This is exactly what should happen. The output will probably look something like the following, with a web address where you can view your results:

```
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Now, all you need to do is copy and paste that web address into your web browser and you'll be able to see your second Flask route!



Great work on successfully setting up your welcome route. You're learning how to build and run a more complex Flask application. Next, we'll split up the code we wrote for the temperature analysis, precipitation analysis, and station analysis, and apply it to the respective routes. Let's start with the precipitation route.