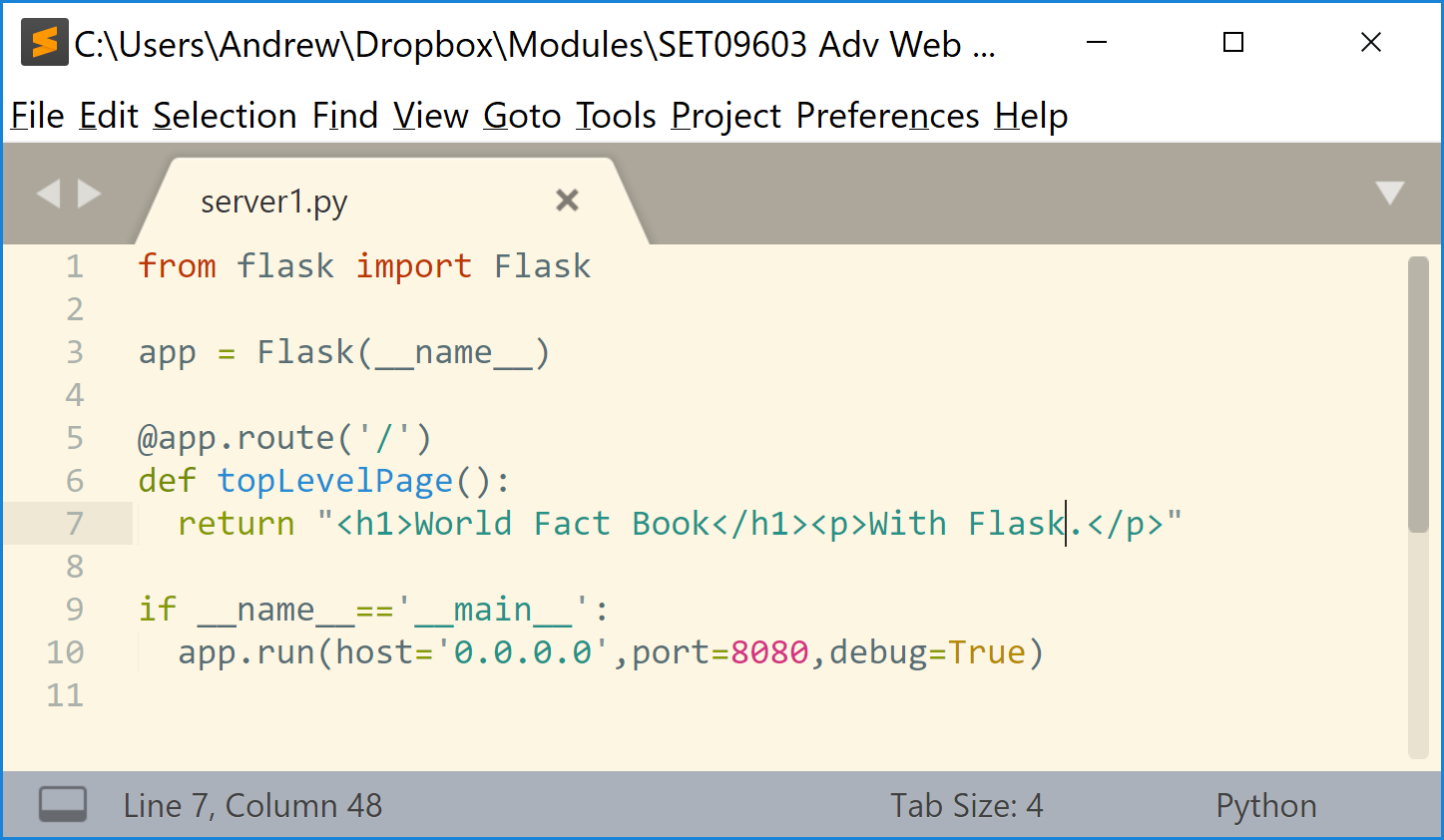
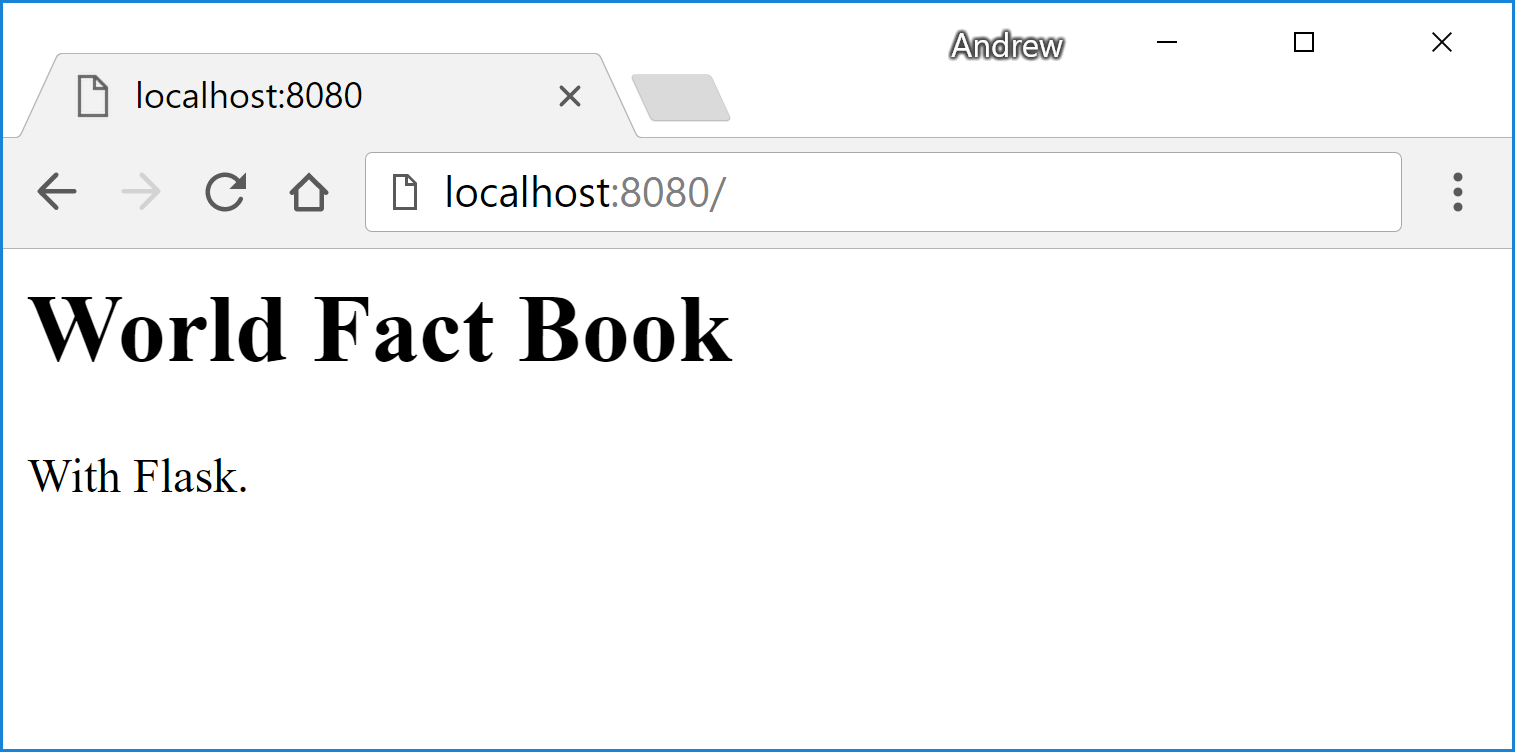
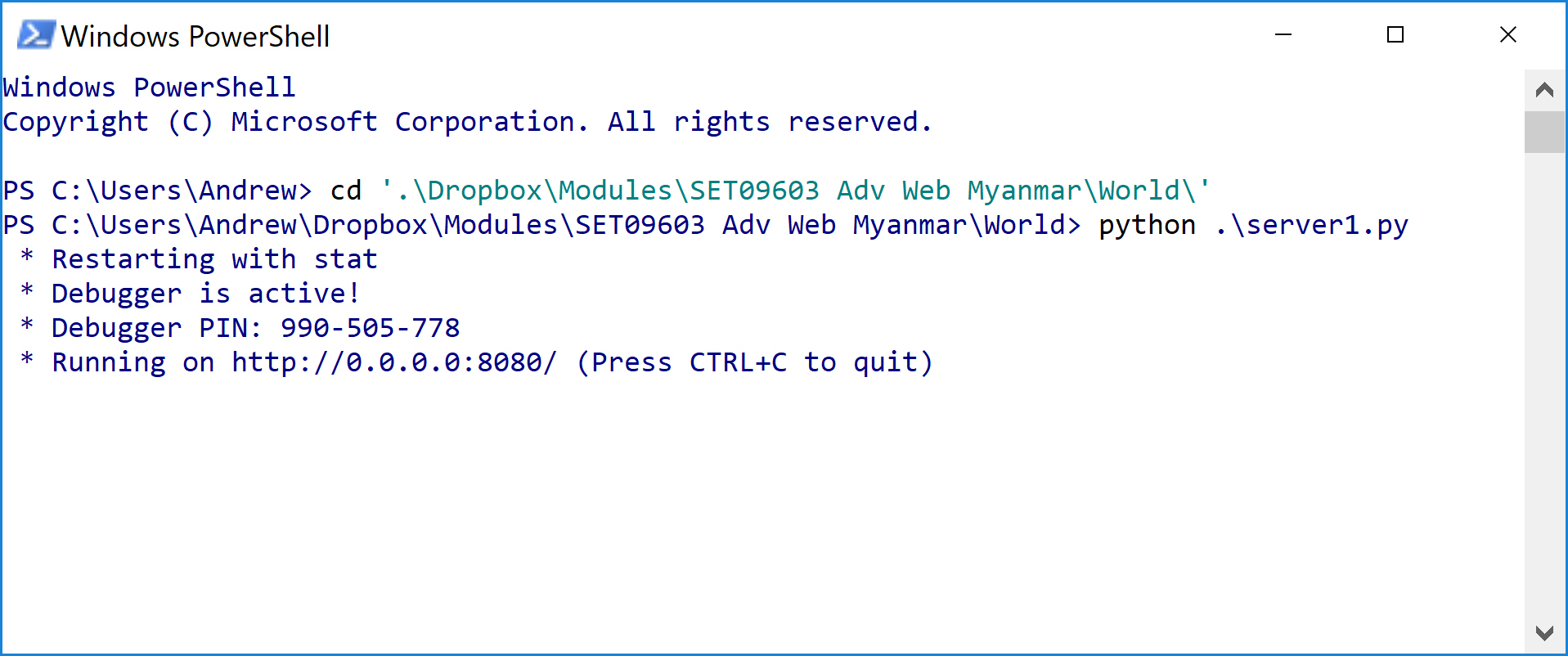
# World Fact Book with Flask

In the tutorial you will learn how to build a simple application using python Flask.

You will need python, a web browser and a text editor.

* Create the python program server1.py as shown
* Open a command prompt and run the program
* Open a browser and visit the address http://localhost:8080/





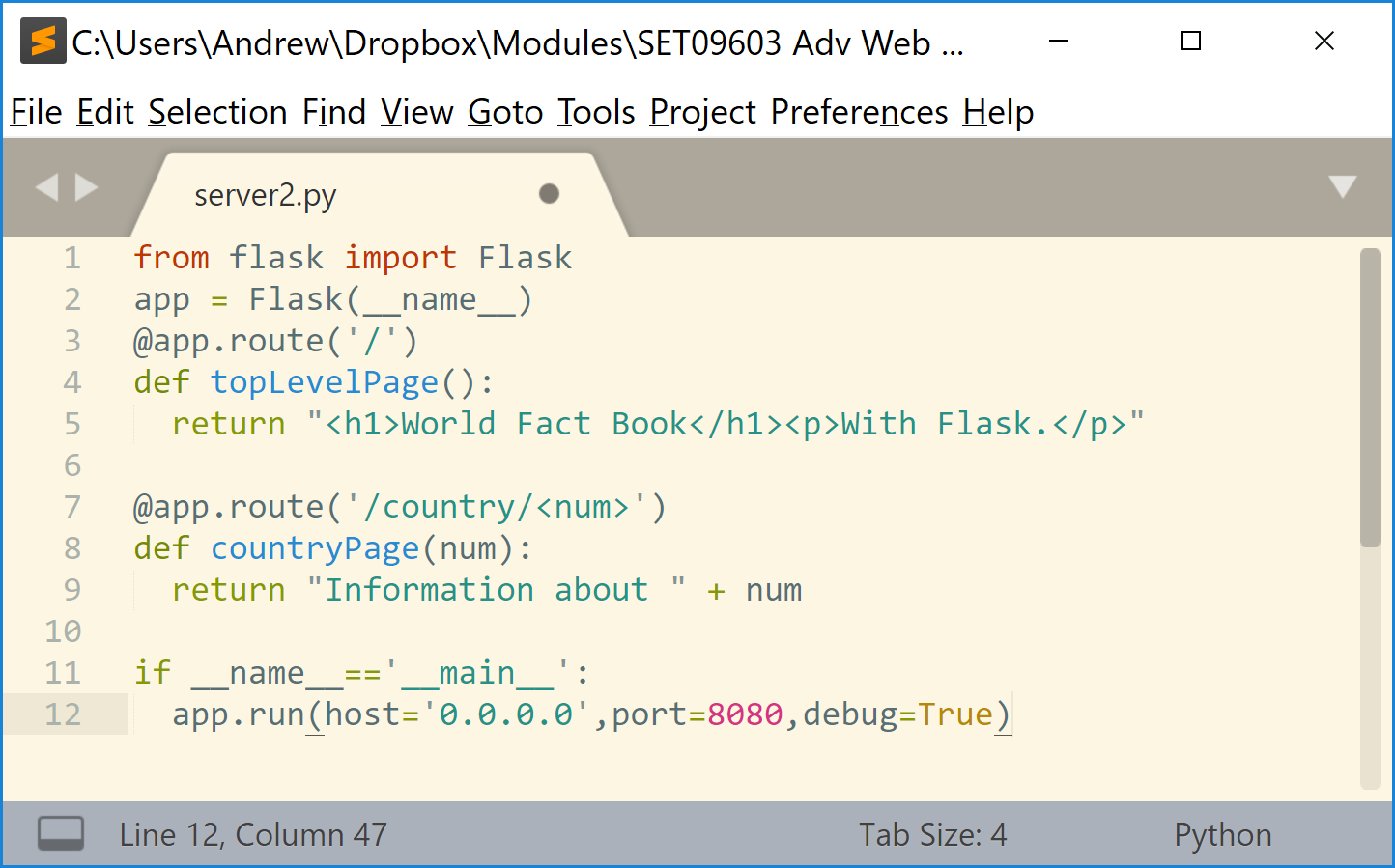
The python program listens on port 8080 – if it gets a request to the address / it runs the method **topLevelPage** and delivers this html to the web browser.

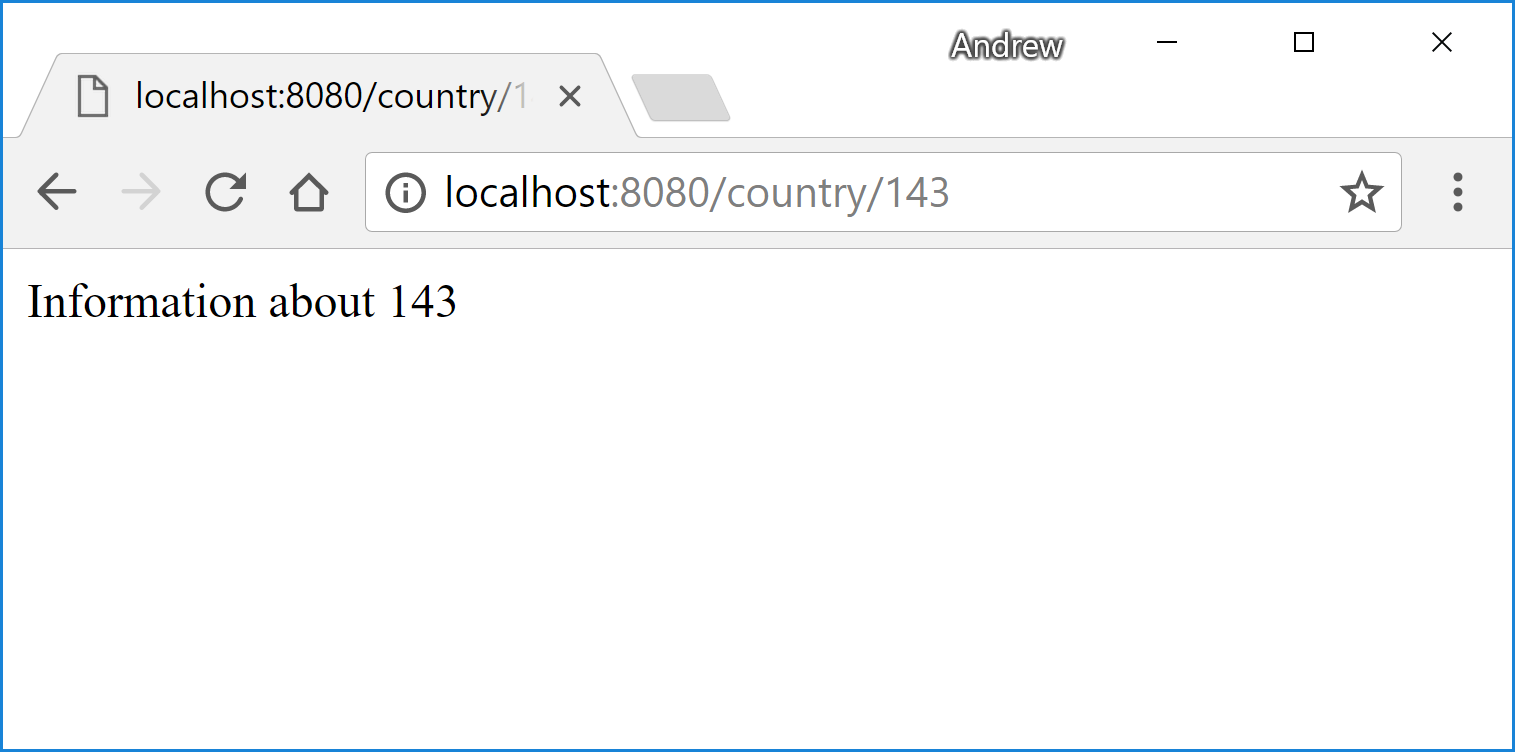
Things to try:

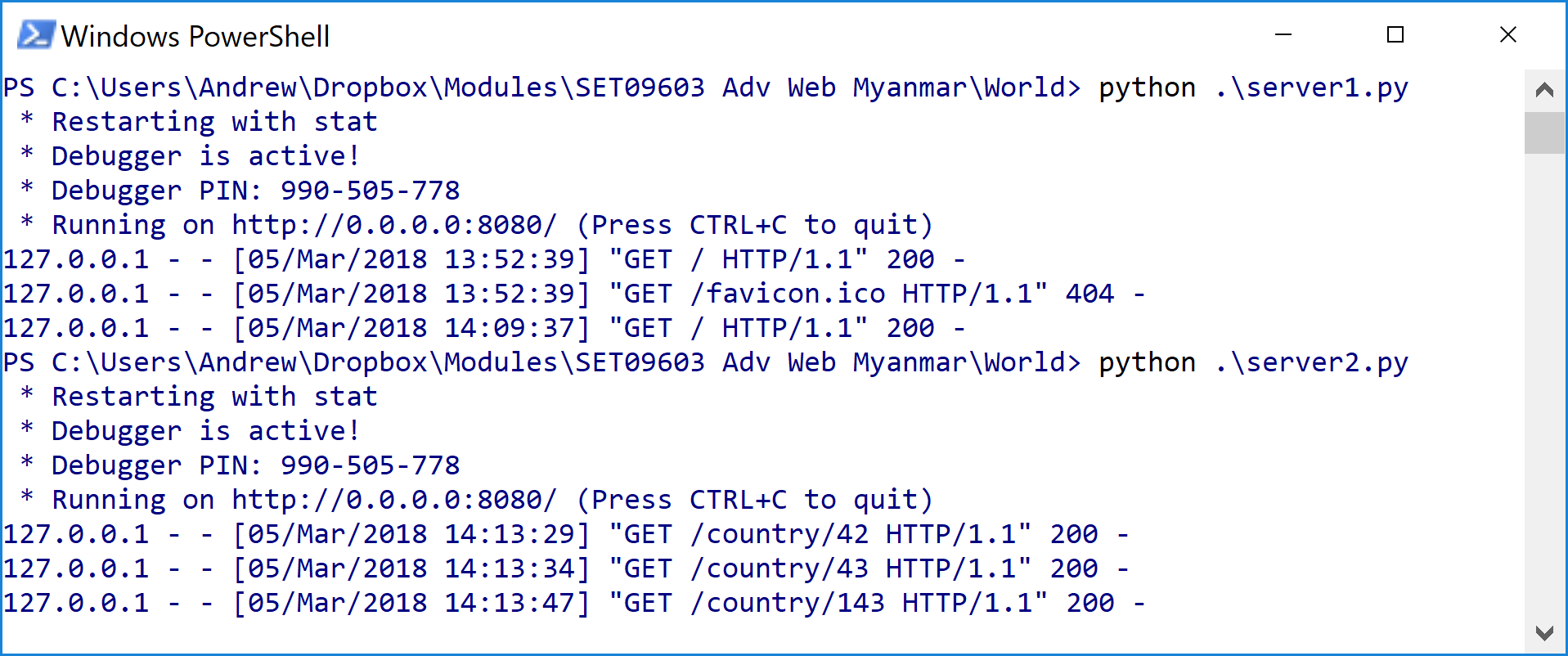
* Visit the page <http://localhost:8080/country/42> (you should get a 404 error)
* Use Control-C to stop the server and visit the top page (you should get a server not found error)

## Implement the country page

We will change the server so that it responds to another address. This is a more complicated route – it allows us to pass a parameter from the browser address bar to the python program.





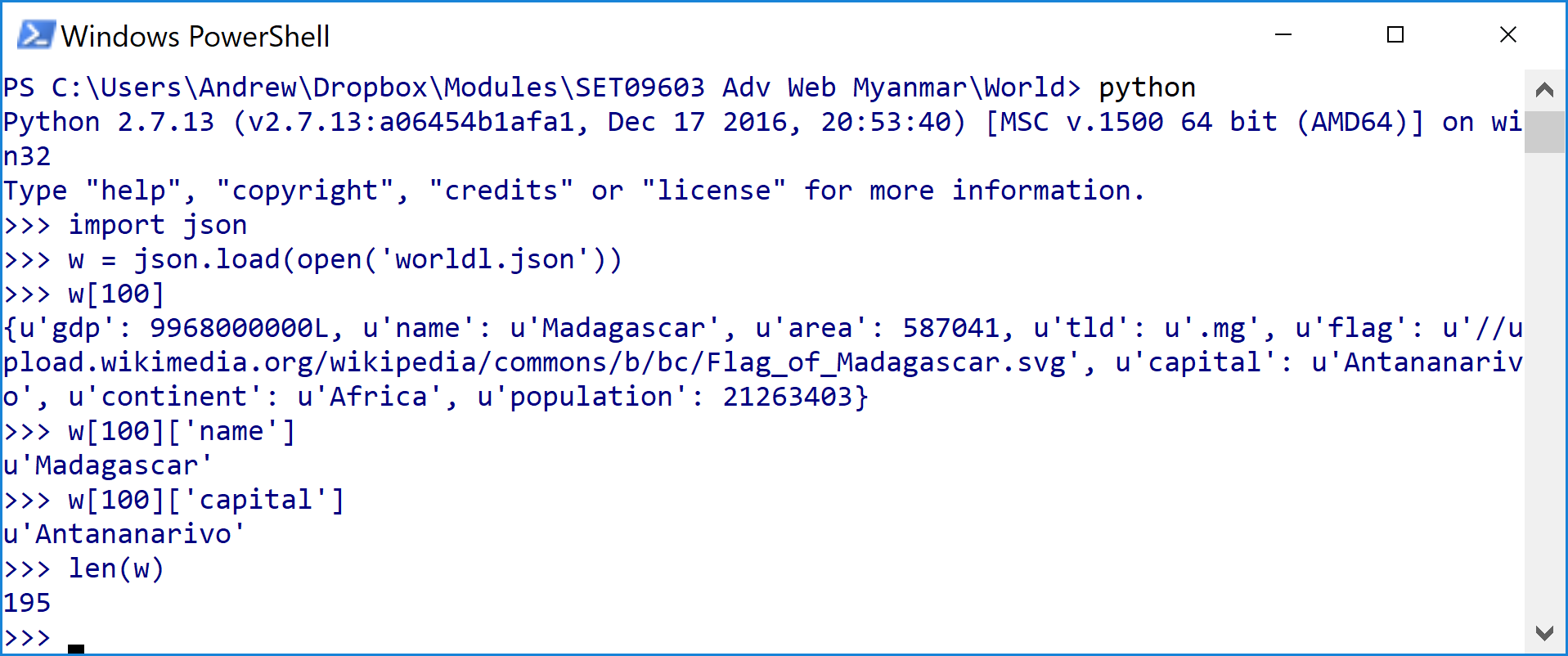


## Working with lists and dictionaries in python

It’s time to work with some data. We will work at the command line to start with.

Download some data – the file at <http://progzoo.net/worldl.json> has information on nearly 200 countries. You should save this file on your computer.

From the python command prompt you can import the json library and then load the worldl.json file into a list called **w**.



The list w has details of 195 countries.

Country number 100 is Madagascar, the capital of Madagascar is Antanarivo.

Using python, answer the following questions:

1. What is the name of country 42
2. Find the capital of country 50
3. Find the flag of country 184

## List Comprehension

List comprehension can be used to filter the countries. You can use it to extract different fields and different elements.

Some examples:

|  |  |
| --- | --- |
| **Expression** | **Value** |
| [c['name'] for c in w] | [u'Afghanistan', u'Albania', u'Algeria', u'Andorra', u'Angola', u'Antigua and Barbuda', u'Argentina', u'Armenia', u'Australia', u'Austria',… |
| [c['name'] for c in w if c['population']>250000000] | [u'China', u'India', u'Indonesia', u'United States'] |
| sum([c['population'] for c in w]) | 7118386310L |
| len([c for c in w if c['continent']=='Africa']) | 53 |
| [c['name'] for c in w if c['name'][0]=='D'] | [u'Denmark', u'Djibouti', u'Dominica', u'Dominican Republic'] |

Use list comprehension to get the following lists:

1. The countries with an area of more than 5000000 sq km.

[u'Australia', u'Brazil', u'Canada', u'China', u'Russia', u'United States']

1. The populations of those countries with a population of more than 250 million

[1365370000, 1246160000, 252164800, 318320000]

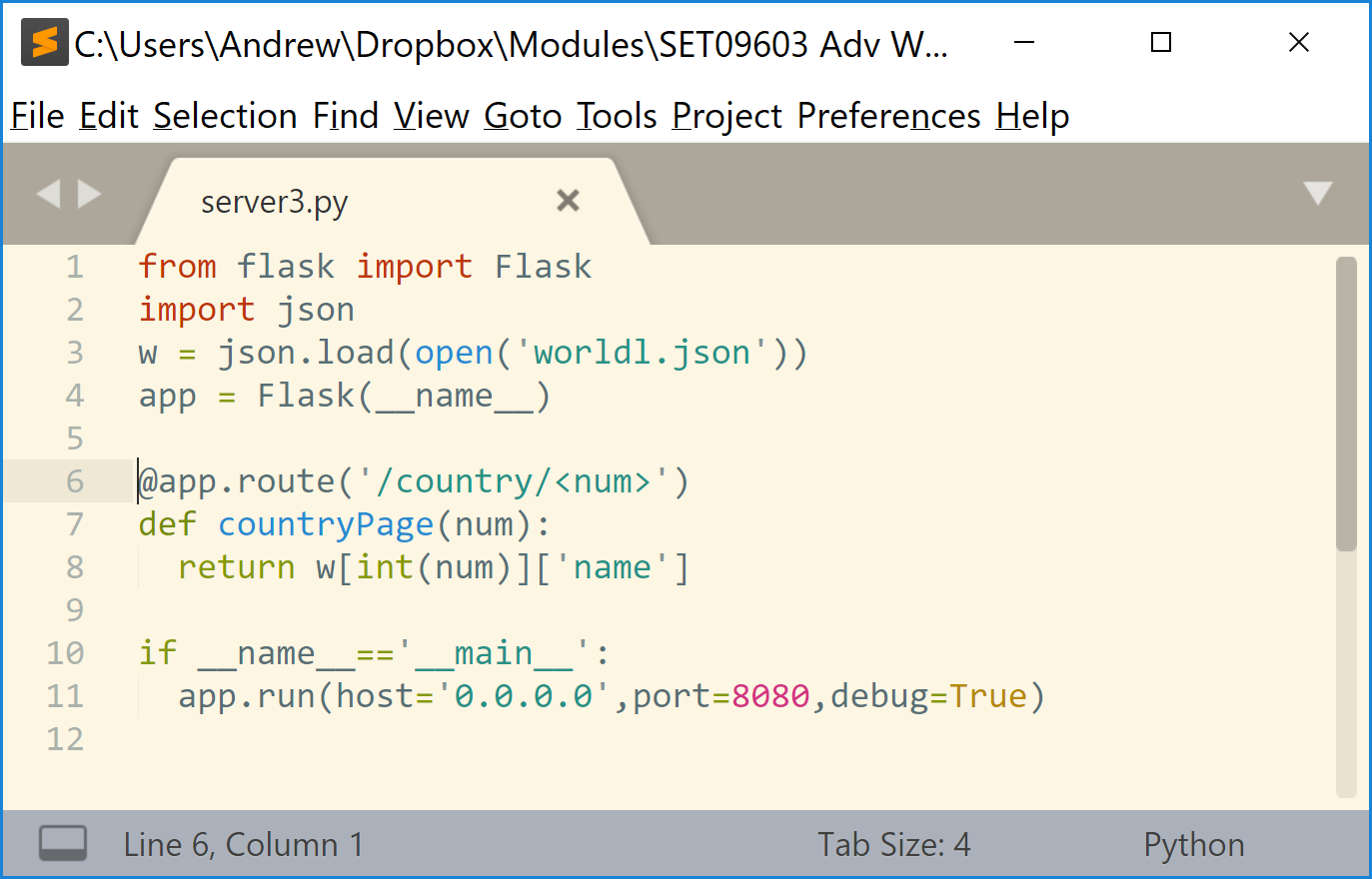
1. The total population of Africa

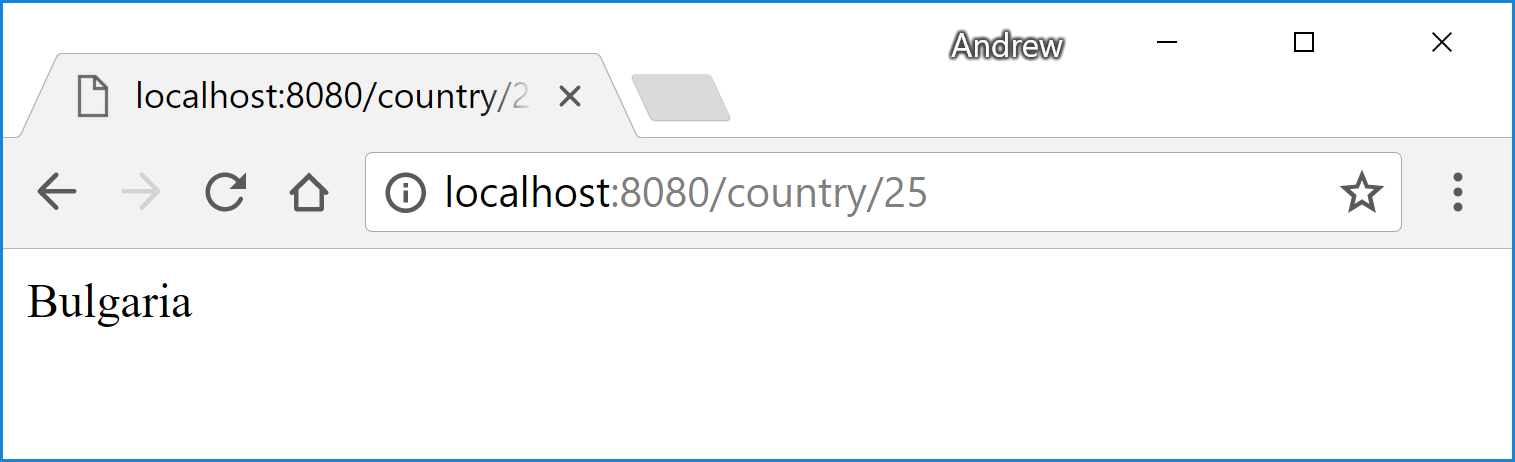
1015844577

You can use the function **next** to find the first country with a particular feature:

|  |  |
| --- | --- |
| next(c for c in w if c['name']=='France') | {u'gdp': 2611221000000L, u'name': u'France', u'area': 640679, u'tld': u'.fr', u'flag': u'//upload.wikimedia.org/wikipedia/commons/c/c3/Flag\_of\_France.svg', u'capital': u'Paris', u'continent': u'Europe', u'population': 65906000} |
| next(c['capital'] for c in w if c['name']=='France') | u'Paris' |

## Delivering data from the web application





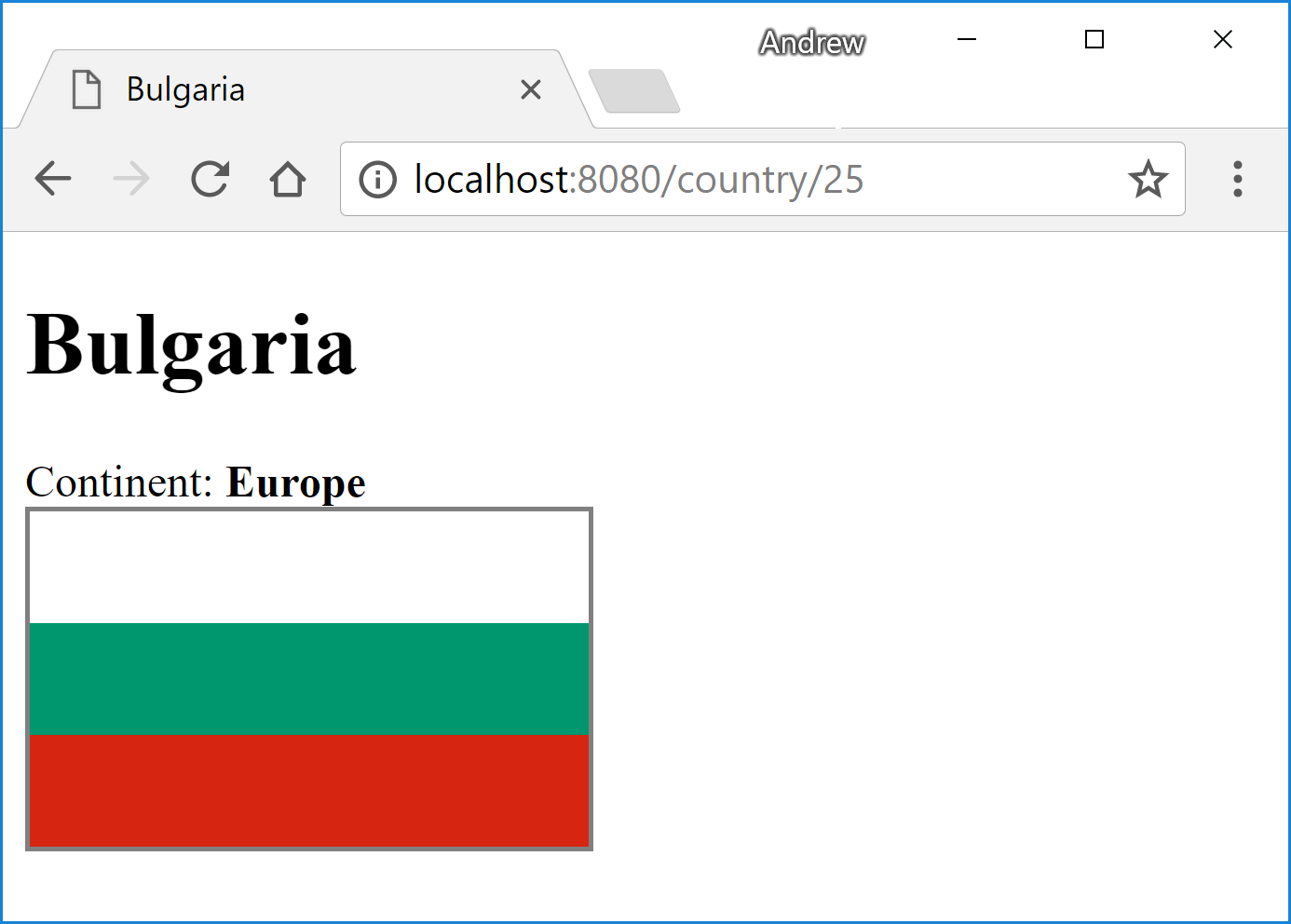
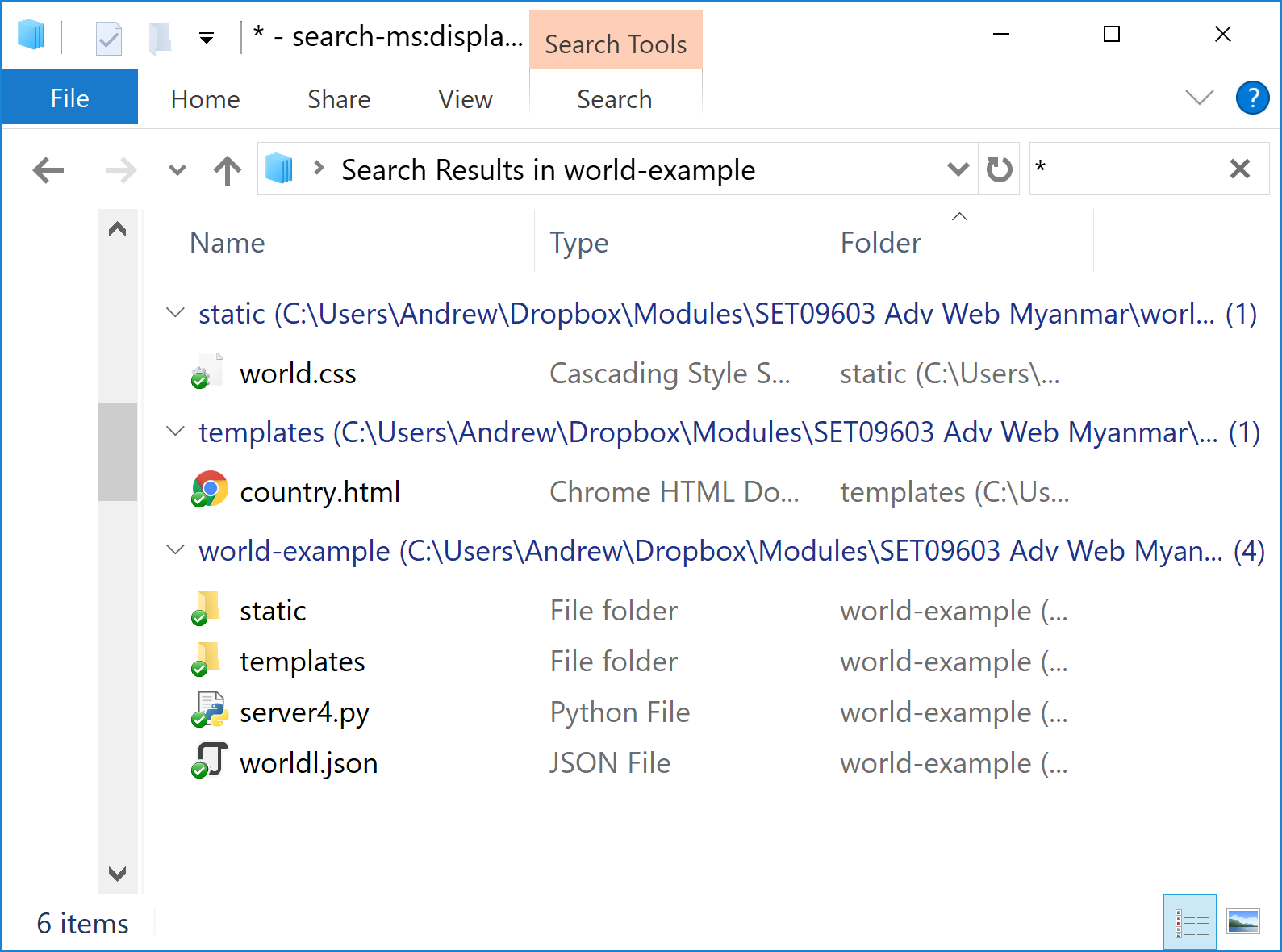
Things to try:

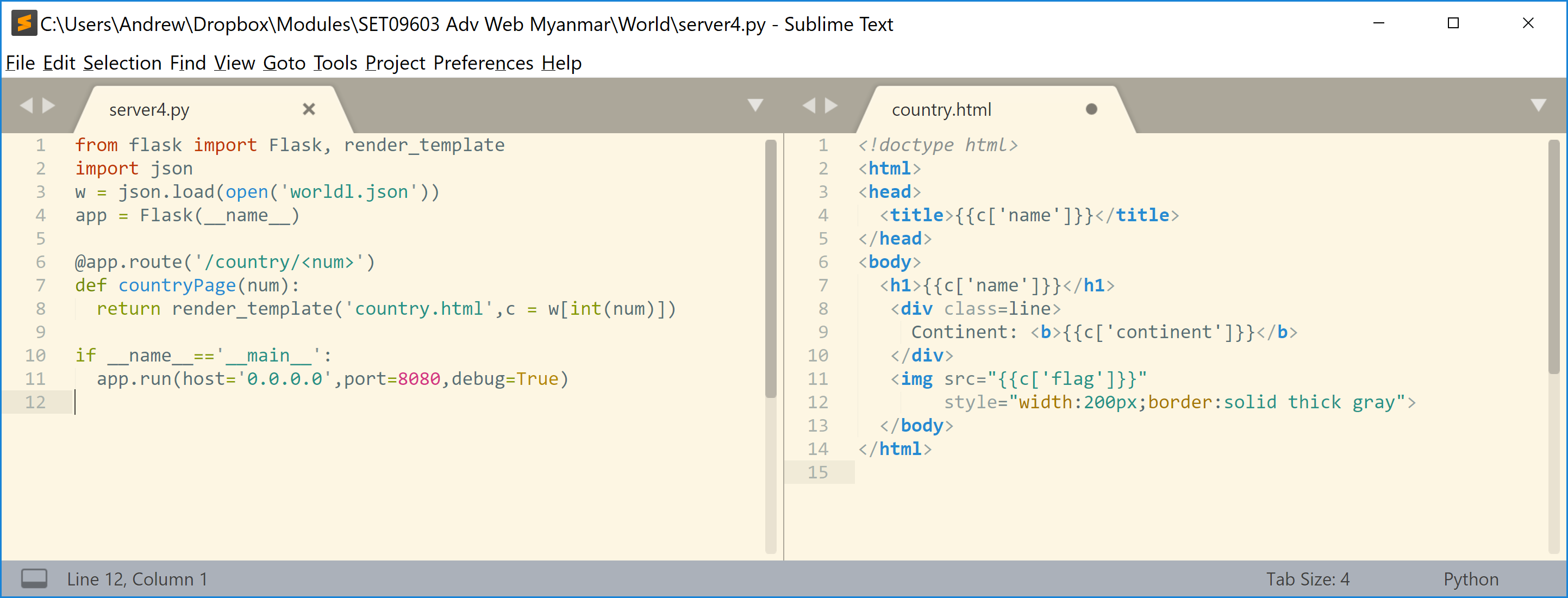
1. You can try other numbers to find the names of different countries
2. You can change line 8 – get it to show the name and the continent of the country

## Using templates

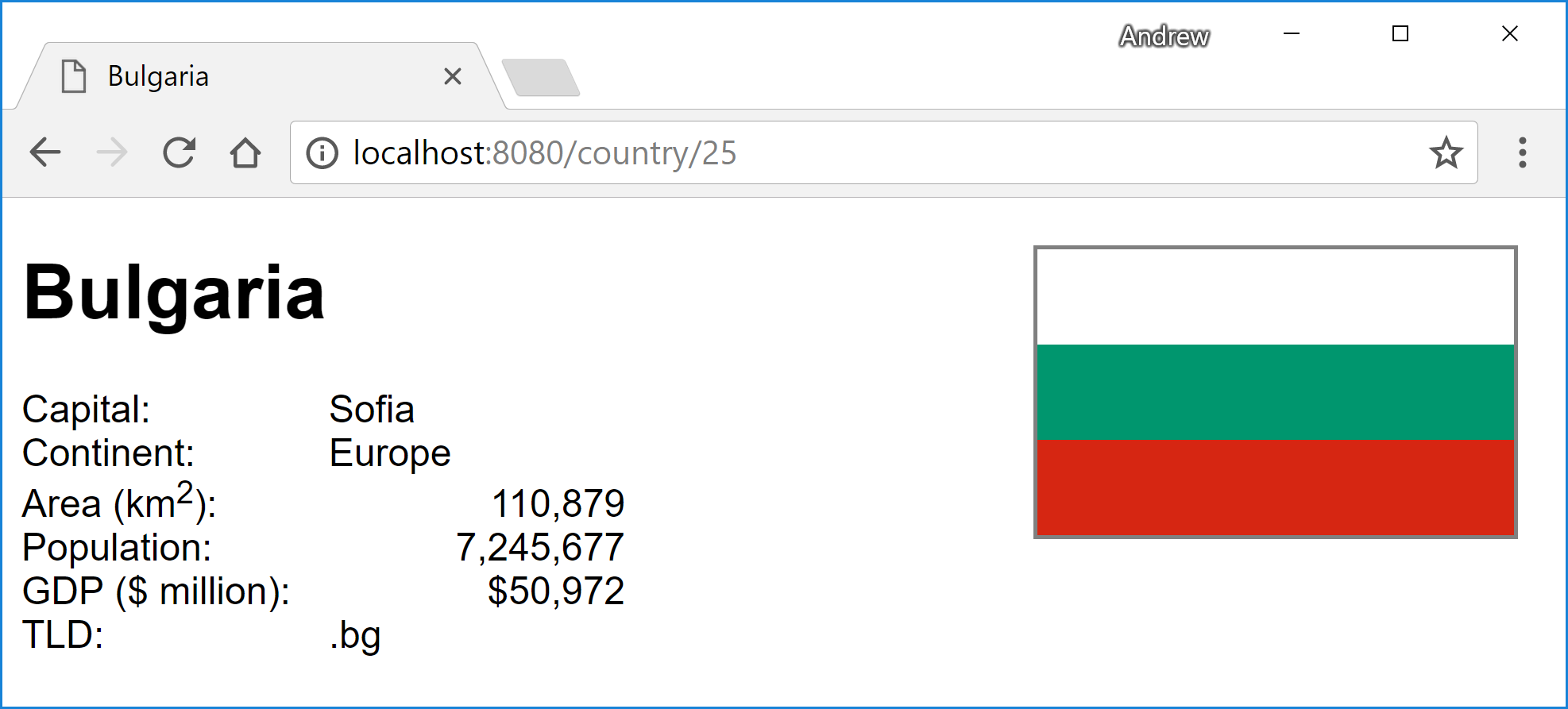
To use a template you must create a folder **templates** and make up a file country.html in that folder.

The template file (country.html) can access variables sent by the python program.

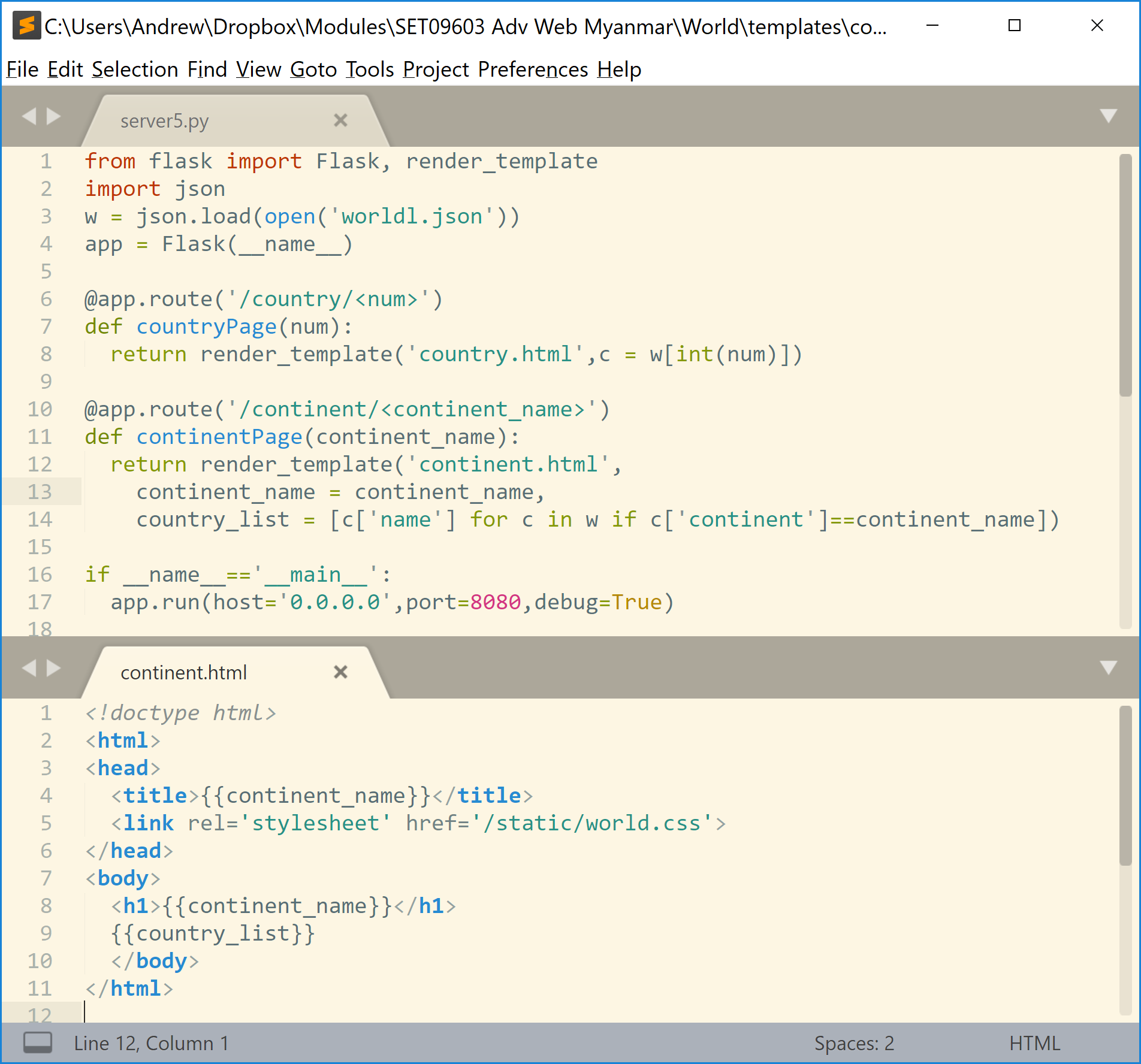


Things to try:

1. Use a stylesheet to improve the look of the template. You should use an external stylesheet in a folder called **static**. The folders **static** and **template** are at the same level.
2. Include all of the details available in the output.

## Showing lists

A second template is required. This will list all of the countries in a specified continent. You should be able to visit the page <http://localhost:8080/continent/Asia> for example:



|  |  |
| --- | --- |
| Good | Better |
|  |  |

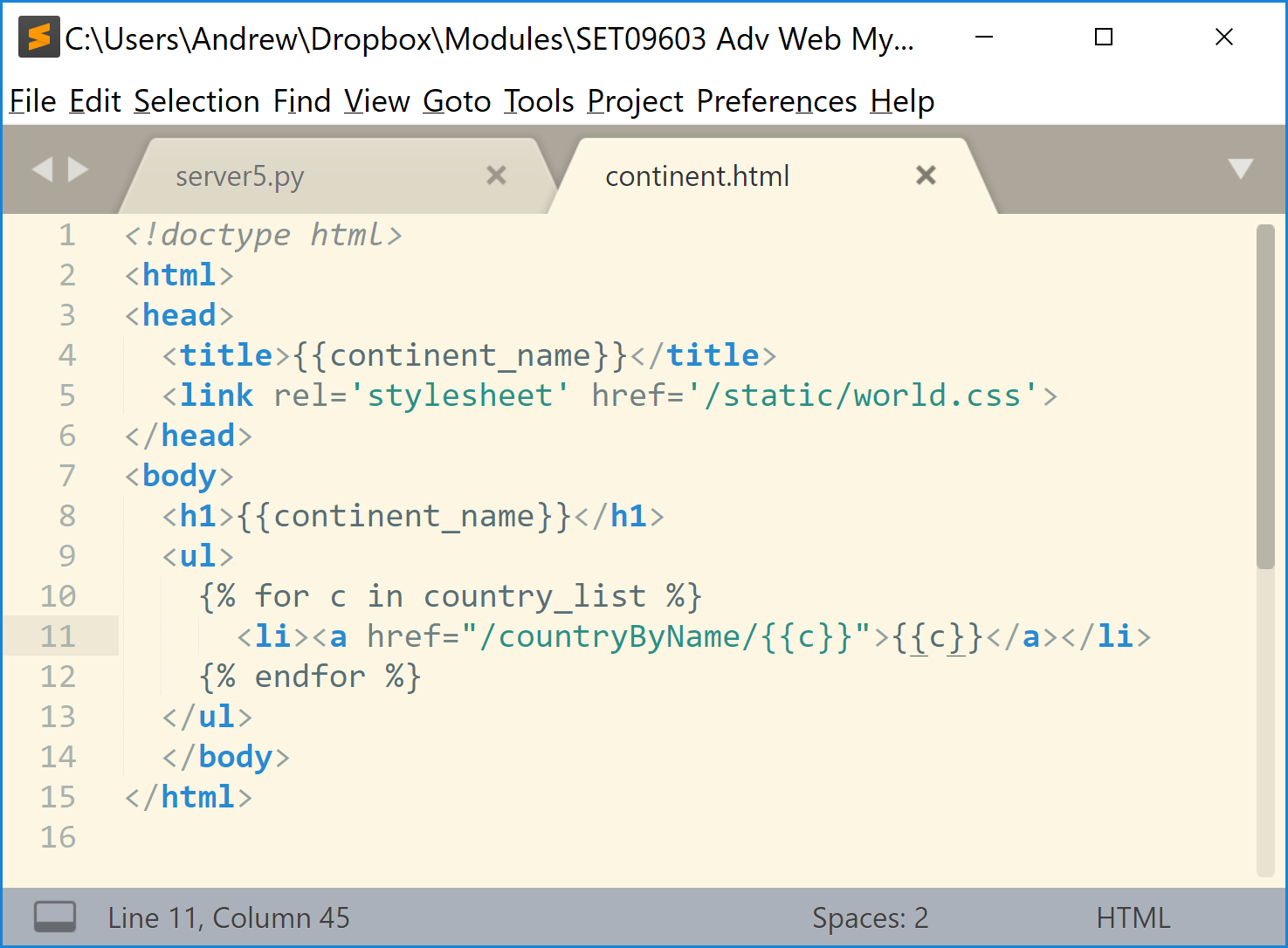
Things to try

1. Format the list of countries as an <li> list of names. Look for the Jinja2 documentation for how to make a for loop
2. Make each country name a link to an address such as
   1. <li><a href='/countryByName/China'>China</a><li>

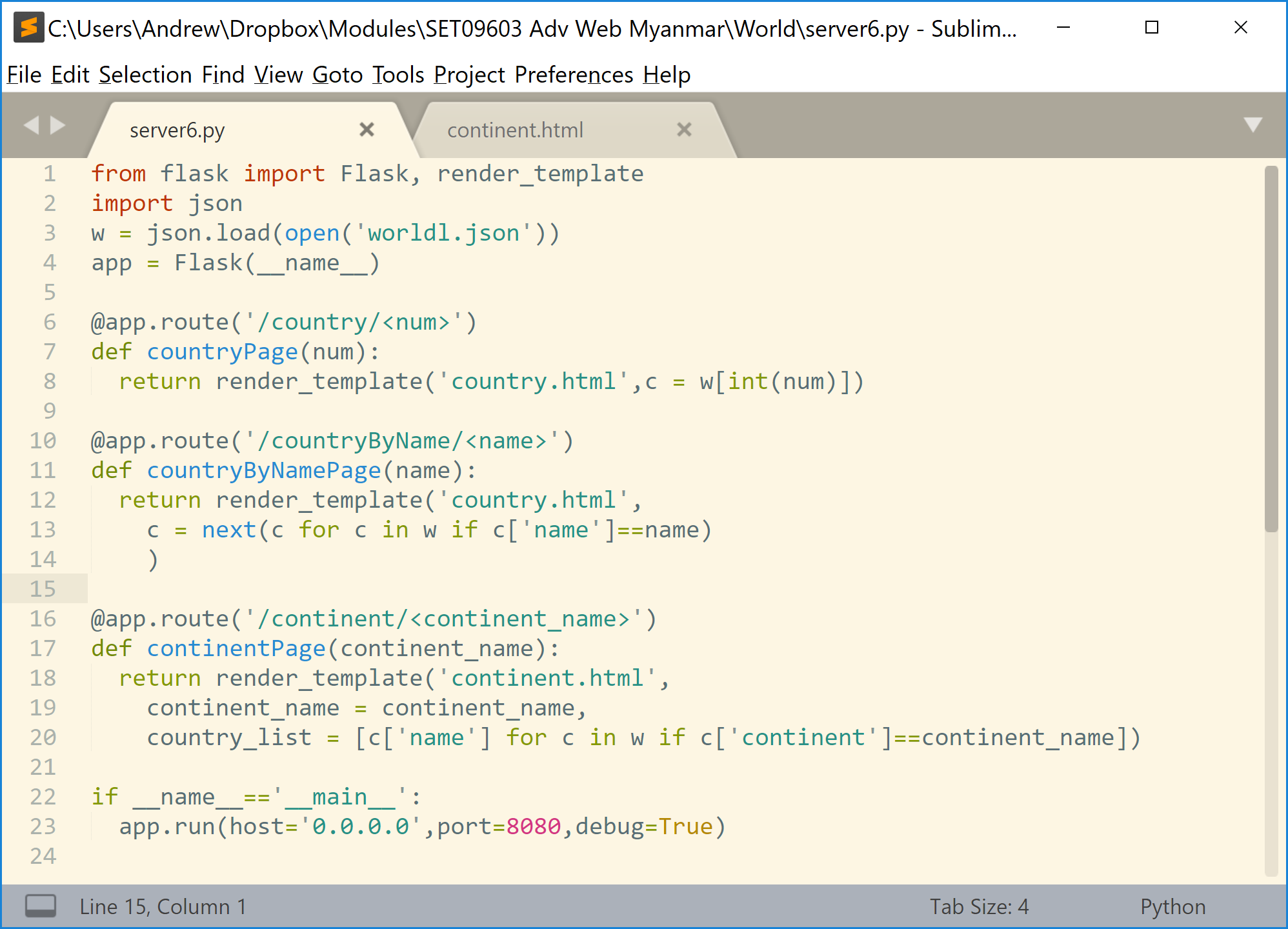
The link will not work yet – we’ll see how to make it work in the next example.

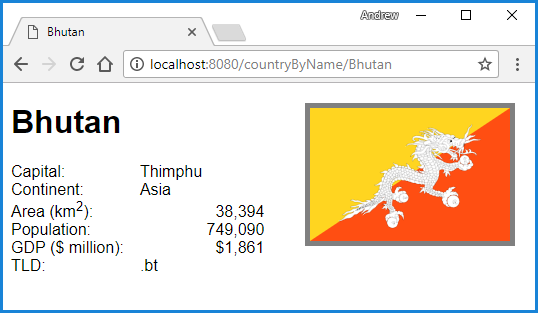
## Making countryByName work

You should be able to create a clickable list of country names with the following template:



To make the links work you need a new route in the server – but you can reuse the country.html template for this route:





## Taking it further

Add the following features to your site:

1. Include an index page at the top level, this should list all continents and allow the user to click on a continent
2. Make the continent value in the country template a link
3. Use template inheritance to include a list of continents in a side label on every page
4. Include an alphabetic list so that the users can easily list all countries beginning with each letter of the alphabet. Include this list on every page.
5. Change the first page so that it shows details of a different random country each time it is visited
6. On the country page include buttons linking to previous and next country (alphabetically).