Web Development Lesson Plan

|  |  |  |
| --- | --- | --- |
| Session: 2 | Topic: Linking pages | Day: 2 |

|  |
| --- |
| Lesson Objective:  This exercise will...   * Instruct students how to link between HTML pages in Django.   The purpose of this lesson is also that   * Students can practice routing HTML files. |
| Materials Needed:   * VS Code * Web browser |
| Agenda:   1. HTML buttons (10 minutes) 2. Django routing (15 minutes) 3. HTML routing (10 minutes) 4. Run your app (5 minutes) |
| Procedure:  Student can do this fully independently by reading the instructions, or someone can read the instructions to make sure the task is understood before doing the activity. The reflection part can be done with guided instructions, or in pairs with other students if available to compare notes. |
| Assessment/Check for learning:   * Students can identify concrete things that they missed in their observation (note that this is not a bad thing, the purpose of this lesson is to realize that intentionally looking for information comes with the ‘penalty’ of reduced attention to other things. |

What if:

* Student has already shown proficiency in that skill?
  + Pick a movie or episode in which a lot of things are happening, so that there are more things for the student to keep track of.
* Student is having a hard day and needs special accommodations?
  + Pick a movie or episode in which only a few things are happening, so that there are less things for the student to keep track of.
  + Instead of writing, the student can also take spoken notes with an audio recorder. This will be a bit trickier in the second part but audio in the movie can be supported with subtitles.

NOTES:

* Duration is approximately 50 minutes; can be adjusted by shortening the duration of the film clip. Less than 10 minutes is not recommended unless really necessary.

**Instructions**

**HTML buttons (10 minutes)**

Step 1:

In the lesson we will link HTML pages together. To navigate to another page we will be using the HTML href tag. Learn about how to use the href tag here:

<https://www.w3schools.com/tags/att_a_href.asp>

Step 2:

Try the examples listed in the tutorial above.

Step 3:

Walk through the following link that explains how href tags can be embedded into other HTML tags:

<https://www.w3schools.com/tags/att_a_href.asp>

Step 4:

Try the examples listed in the tutorial.

**Django routing (15 minutes)**

Step 1:

In this section we will create a new route inside our Lesson10 project.

Step 2:

In VS Code open the Lesson10 project.

Step 3:

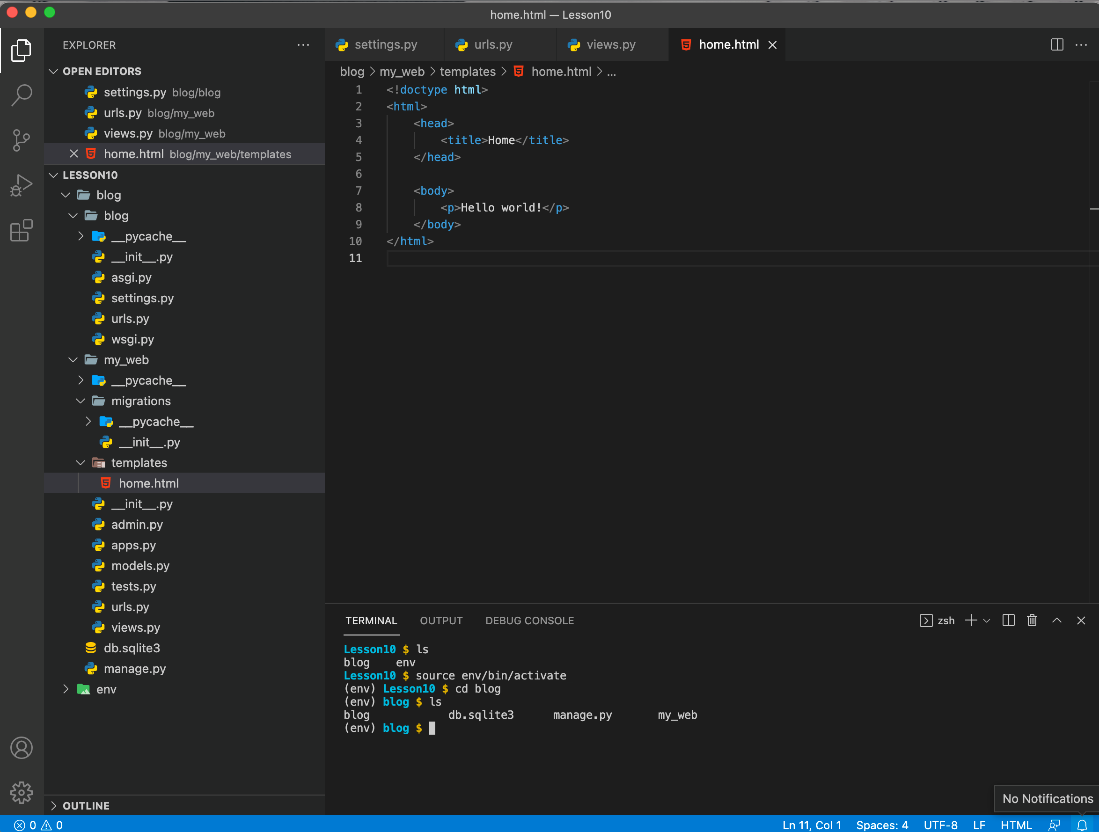
Open a terminal in VS Code.

Step 4:

Activate the virtual environment with the following command:

source env/bin/activate

Step 5:

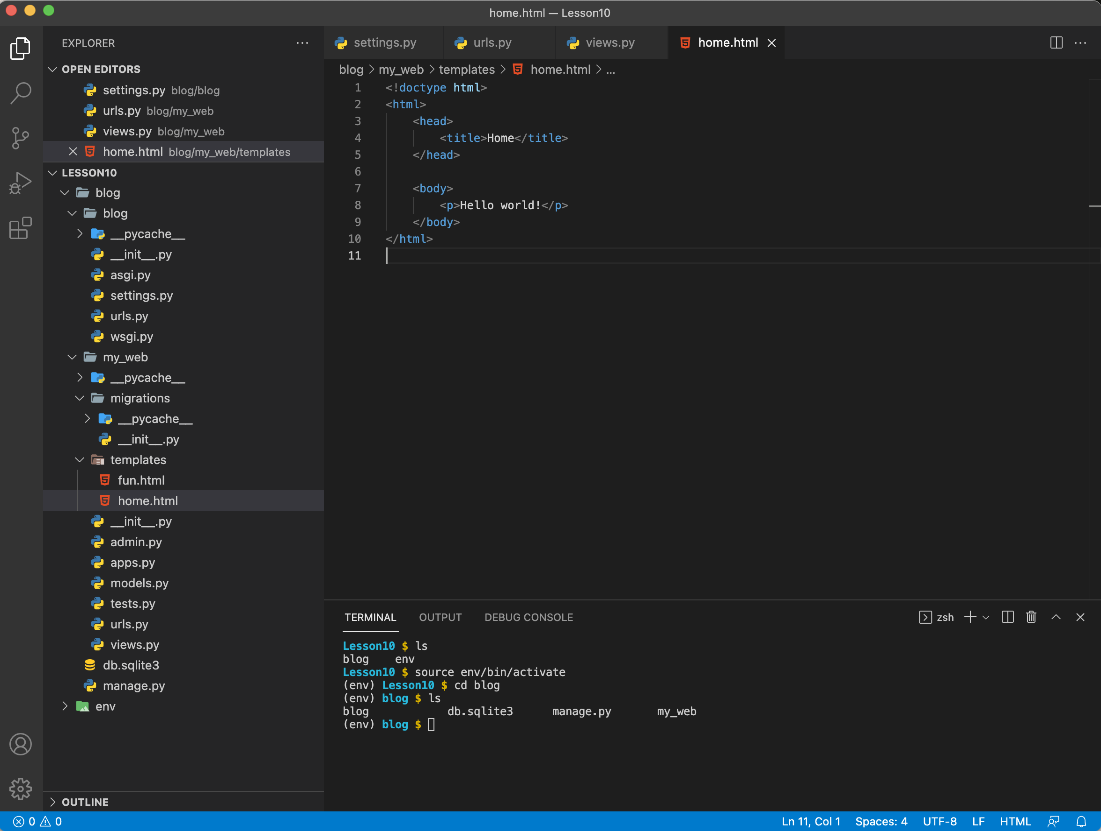
In the terminal cd into the “blog” folder.  


Note:

To make sure you are in the correct directory in your terminal type **ls** and you should see 4 items like above.

Step 6:

First, we will create a new HTML file. Right click on the “templates” folder and create a new file. Name the file: “fun.html”



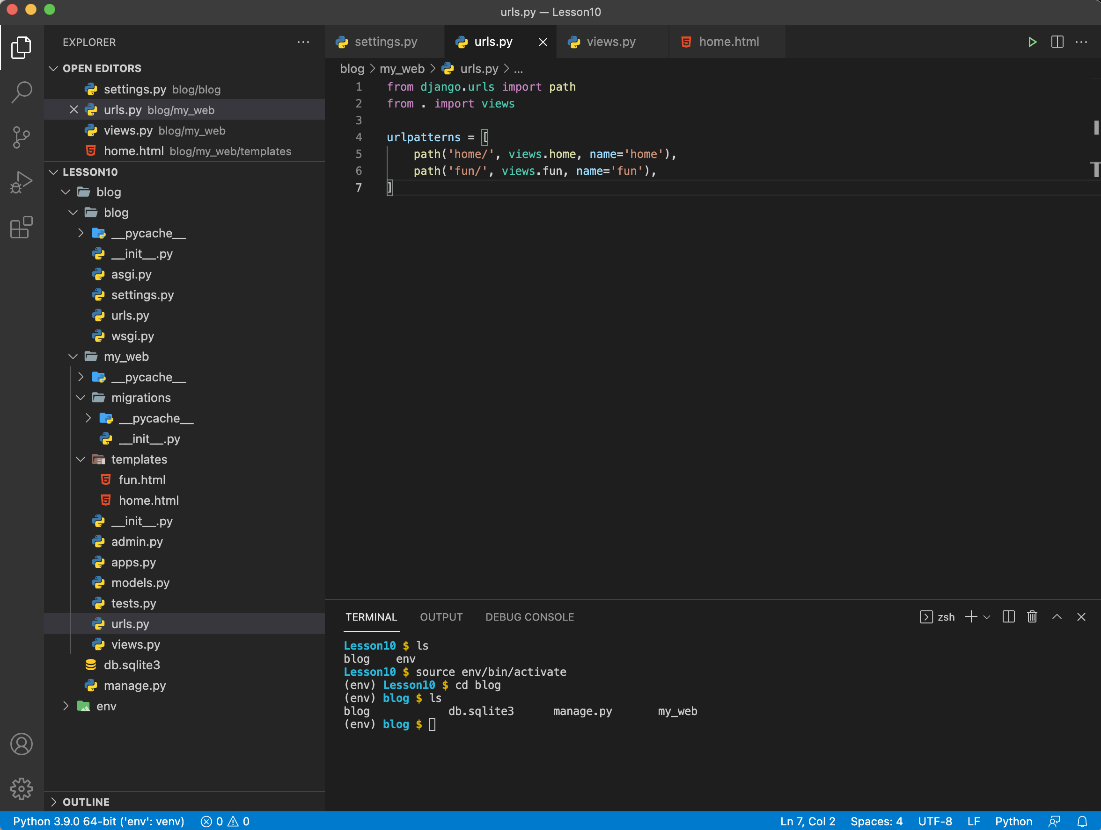
Step 7:

Open the “urls.py” file that exists within the “my\_web” folder. This file is directly above the “views.py” file. You should see the following when you open the file:

Step 8:

Add the following code after line 5:

path('fun/', views.fun, name='fun'),

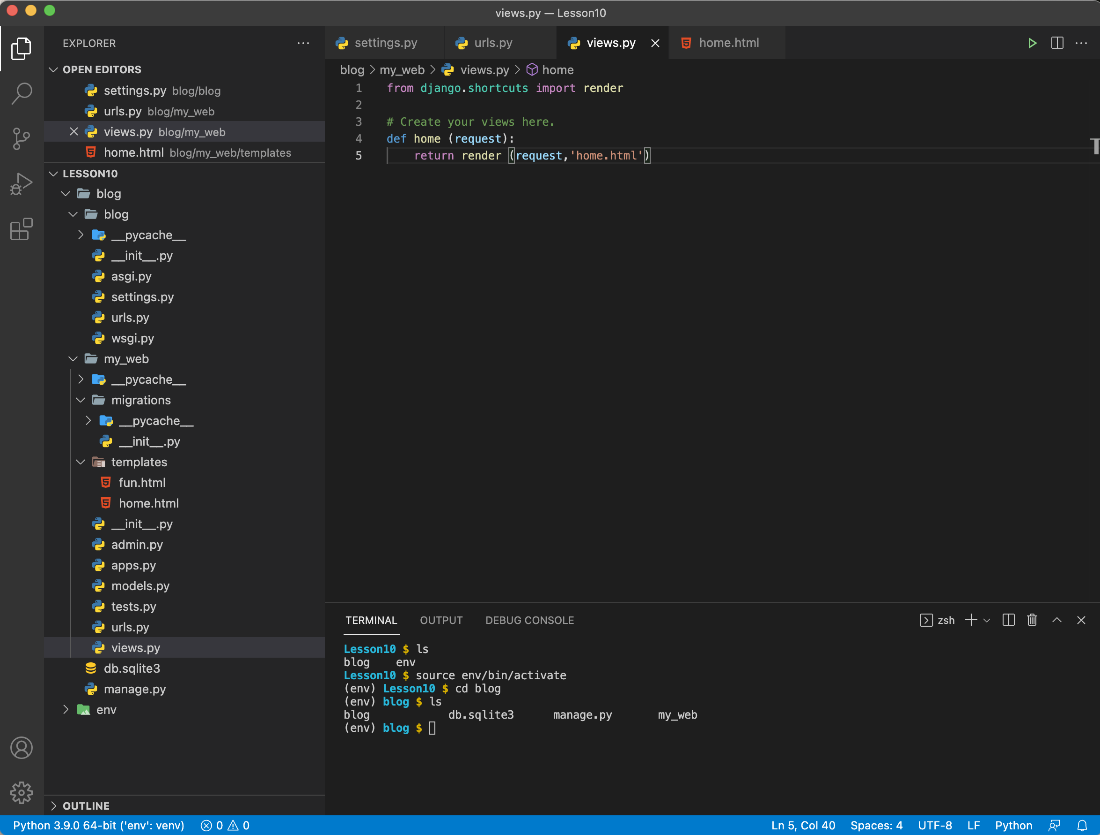


Step 9:

Save the file.

Step 10:

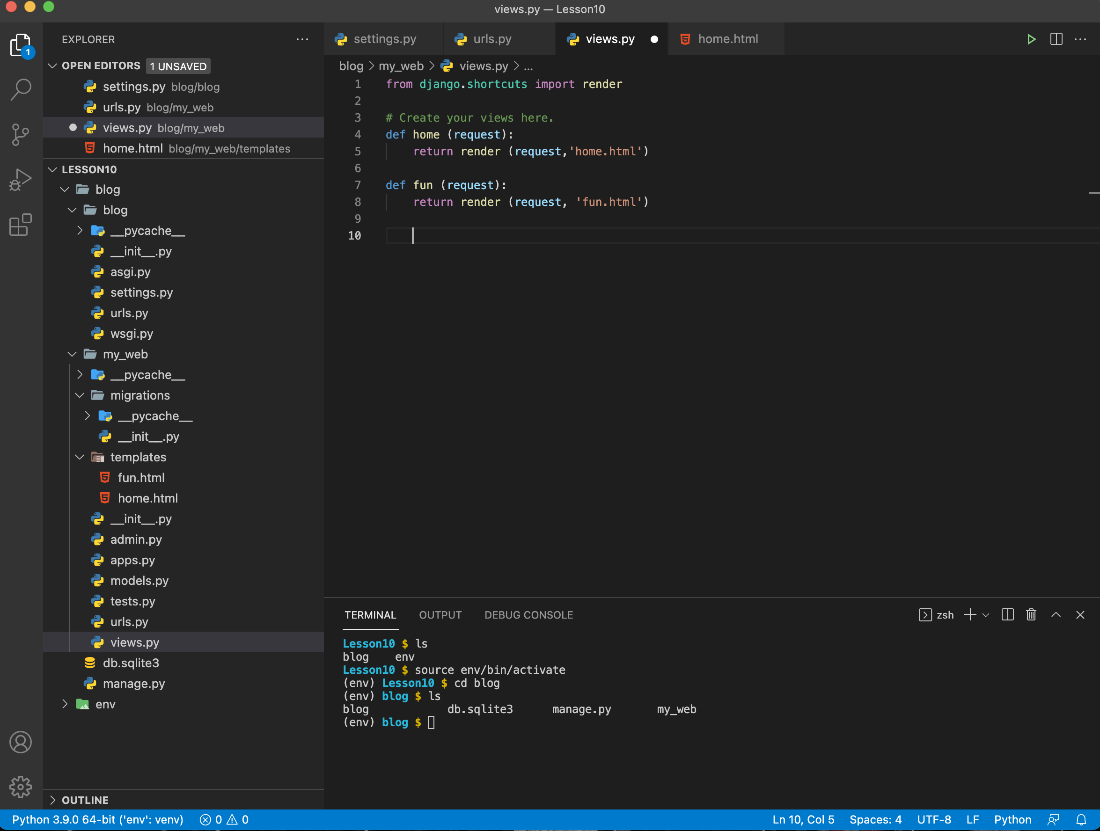
Open the “views.py” file. You should see the following when you open the file:



Step 11:

At line 7, add the following code to link the url to a view function.

def fun (request):  
 return render (request, 'fun.html')



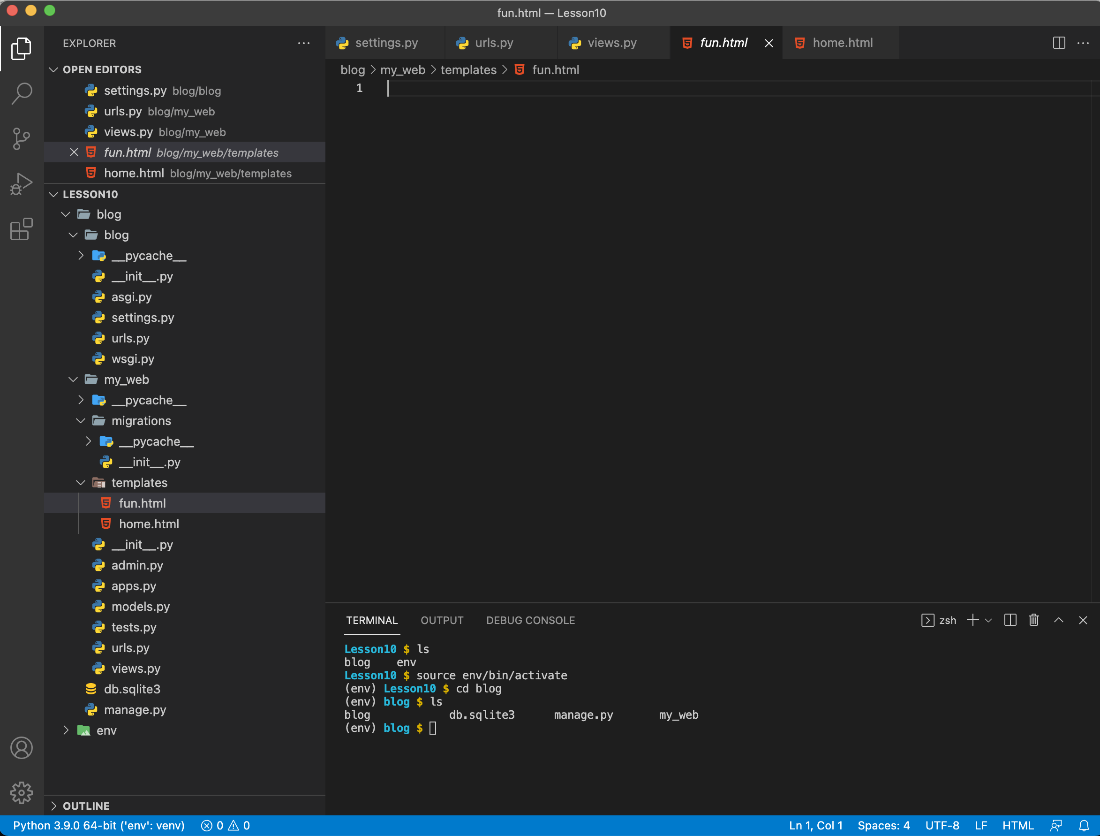
Step 12:

Save the file.

**HTML routing (10 minutes)**

Step 1:

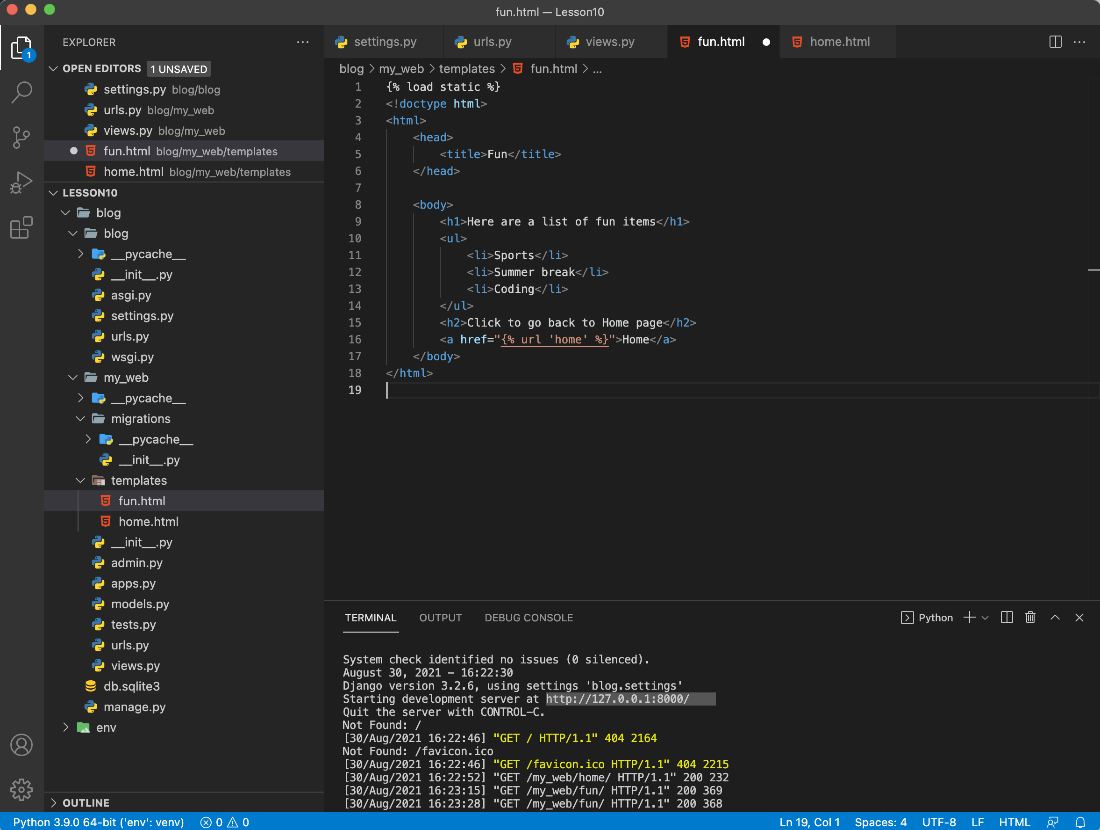
Now we have both the url route and the view function. Next, we will edit the HTML file we created. Open the “fun.html” file. It should be completely empty.



Step 2:

Add the following code:

{% load static %}  
<!doctype html>   
<html>   
 <head>   
 <title>Fun</title>   
 </head>   
 <body>   
 <h1>Here are a list of fun items</h1>  
 <ul>  
 <li>Sports</li>  
 <li>Summer break</li>  
 <li>Coding</li>  
 </ul>  
 <h2>Click to go back to Home page</h2>  
 <a href="{% url 'home' %}">Home</a>  
 </body>   
</html>



Note: Notice that at the top of the HTML file there is a non-HTML tag “**{% load static %}**” This tag tells Django to load static files. This tag is used to load other HTML files, images, and CSS.

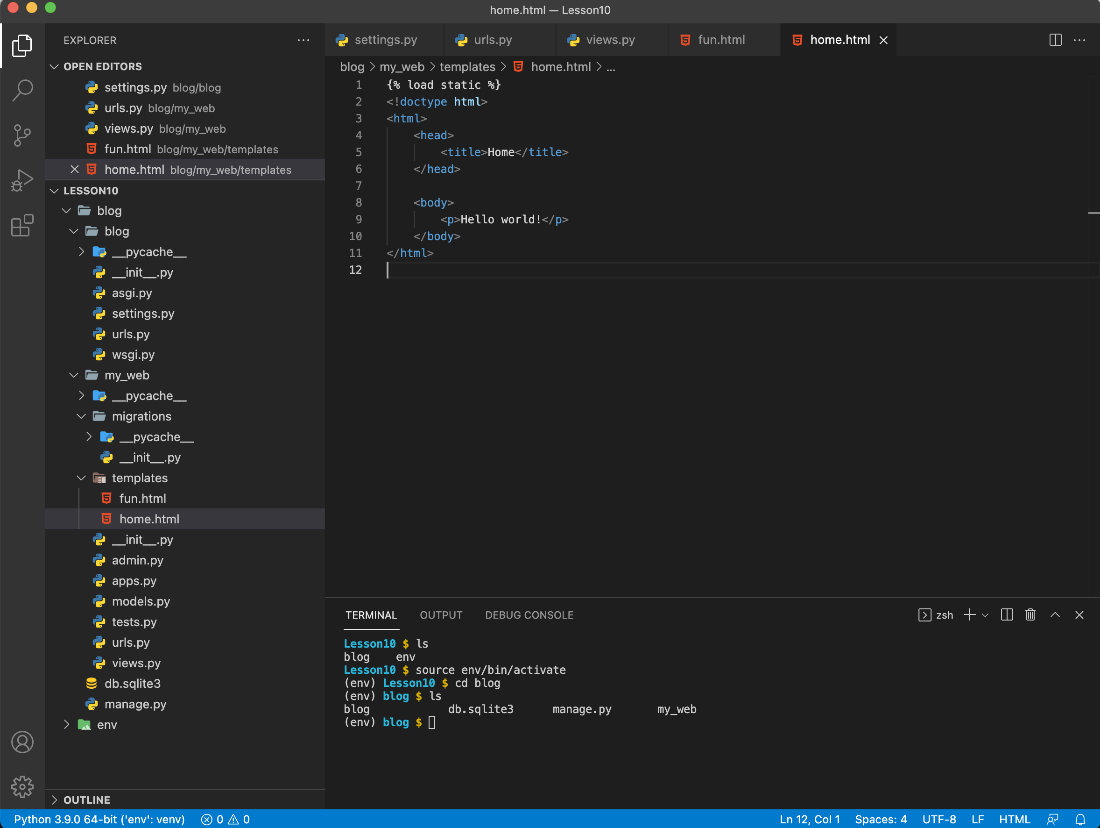
Step 3:

Save the file.

Step 4:

Open the “home.html” file. Add the following to the top of the file:

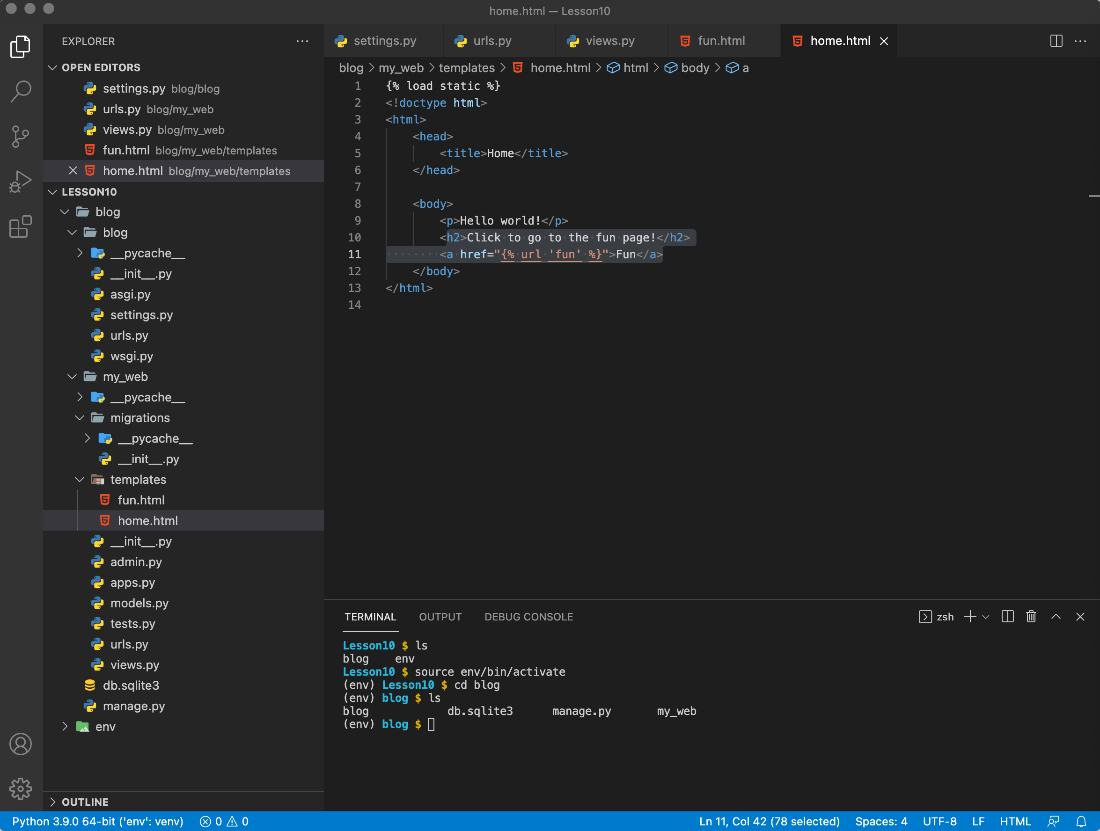
{% load static %}



Step 7:

Now we will add an href tag to link to “fun.html”. Add the following after line 9.

<h2>Click to go to the fun page!</h2>  
<a href="{% url 'fun' %}">Fun</a>



Step 6:

Save the file.

**Run your app (5 minutes)**

Step 1:

Finally, it’s time to run your app. Run the following command in the terminal to run your app. Note: Make sure your env is active and you are inside the blog directory.

python manage.py runserver

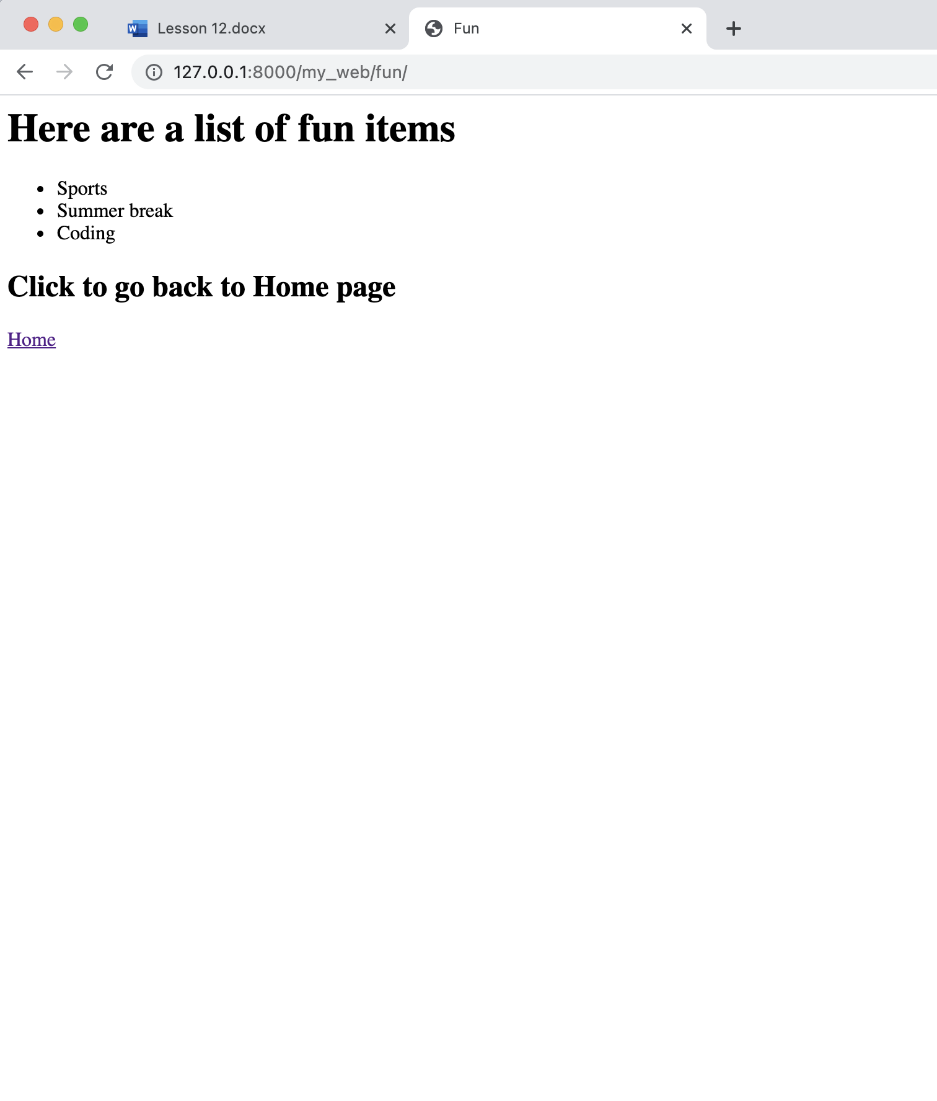
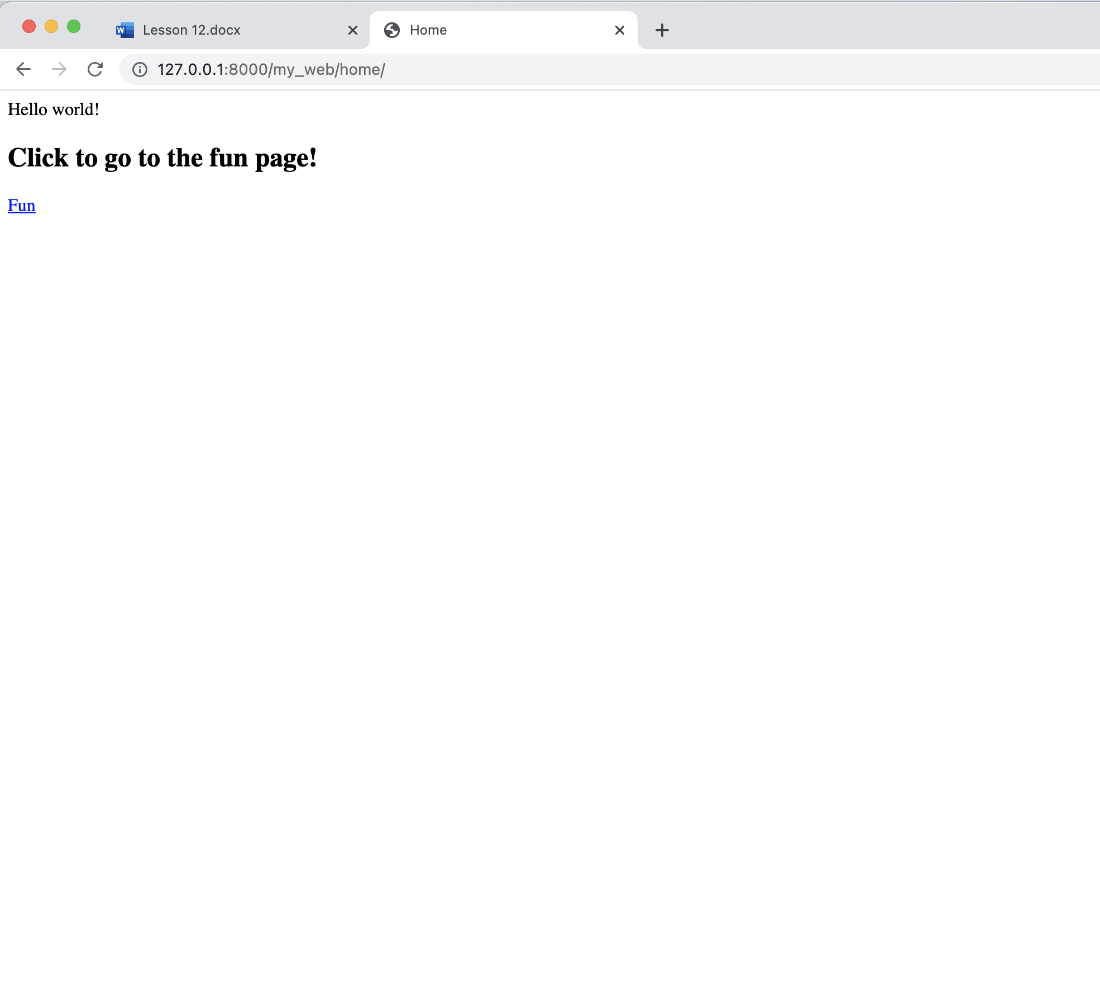
Step 2:

Go to the following link in your preferred browser:

<http://127.0.0.1:8000/my_web/home/>

Step 3:

Test the links between fun and home.



Congratulations you have created links between multiple pages.

In the following lesson you will be introduced to displaying images.