

**MIT AITI**

**Django Lab 3: HTML and CSS**

This is a continuation of Lab 2. In this lab you will make 2 HTML pages for your blog:

1. A homepage
2. A detail page to show details about a specific blog entry

At the end of this week, we will put all of the blogs online and have a contest where students can vote on their favorite blog. So feel free to make your blog look cool with HTML and CSS!

If you get stuck, take a look at these resources:

1. Lecture slides
2. Previous labs
3. Other students
4. Django documentation
5. Google
6. Instructors

**Setup steps:**

First we will add some things to your app so you’ll be able to jump right into coding templates.

1. Open up your blog project

$ cd ~/Desktop/myblog

1. Add the following line to ~/Desktop/myblog/myblog/urls.py

url(r'^blog/', include(‘blog.urls')),

Put it right after the line: urlpatterns = patterns('',

This line tells your myblog project to use the urls from the blog app whenever someone goes to a website beginning with <http://localhost:8000/blog/>

1. Open up your blog app

$ cd blog

1. The urls.py file in the blog app will tell Django what view to use when a person visits a url. Download the urls.py file for the blog app by typing:

$ wget <http://aiti.mit.edu/media/programs/ghana-summer-2013/materials/urls.txt>

$ mv urls.txt urls.py

If the internet is not working, skip to step 6.

1. Download the views.py file for the blog app by typing:

$ wget <http://aiti.mit.edu/media/programs/ghana-summer-2013/materials/views.txt>

$ mv views.txt views.py

6. Open up your blog project

$ cd ~/Desktop/myblog

7. Create directories where you will store templates

$ mkdir templates

$ mkdir templates/blog

8. Edit the settings.py file so that django knows where to look for your templates:

$ gedit ~/Desktop/webnotes/webnotes/settings.py

* 1. At the top of the file, add these two lines:

*import os*

*PROJECT\_ROOT = os.path.realpath(os.path.dirname(*

*os.path.dirname(\_\_file\_\_)))*

* 1. Find the line that starts with “TEMPLATE\_DIRS”. You can search within a file using ctrl-F. Replace the lines with the following:

*TEMPLATE\_DIRS = (*

*os.path.join(PROJECT\_ROOT, 'templates')*

*)*

**HTML steps**

Now you’re ready to write some HTML!

1. Change directory to templates/blog. Figure out how to navigate there using the diagram from lab 2.
2. Make a file called home.html, by opening home.html in gedit. This will be the homepage to your blog. In urls.py, we speficy that if someone navigates to /blog, this page will show up. In home.html write:

**hello world**

1. Open a new Terminal tab by right-clicking and choosing “open tab.”

In this tab, run your server (python manage.py runserver)

1. Open a browser and go to <http://localhost:8000/blog/>

If the internet is not working: Open Firefox. Press ctrl-O to open a file. Navigate to Desktop->myblog->blog and choose home.html

Right now it should just say “hello world.” In the next exercise you will make your homepage look however you want.

1. Edit home.html.This is a chance to get practice using HTML. Make a page about yourself or some of your interests. It can look however you like and include whatever information you want. It must include the following features:

* A <head> section with a title
* A <body> section with content, including:
* Two different header tags
* Some bold text
* An unordered or ordered list
* An image linked from the internet (if internet is working)
* A link to a website

See the HTML reference at the end of this lab. You can refresh <http://localhost:8000/blog/> at any point to see how it looks so far.

1. Add some CSS to your file.

Create a file called home-styles.css

See the css example and reference at the end of this lab.

1. In the <head> section of home.html, add the line:

<link rel="stylesheet" type="text/css" href="home-styles.css">

1. Now we’re going to create a webpage displays a single blog post and all of its comments. This page is called detail.html. In the next lab we will learn how to put real content from any blog post into this HTML template.

Create a new template file detail.html in templates/blog.

In detail.html, type:

**your content here!**

1. In your browser, go to <http://localhost:8000/blog/detail/1>

You should see the name of your blog and “your content here!”

1. Now replace “your content here!” with content displaying a fake blog post. It should include the post’s title, body, date of creation, and date of last update. It should also include comments on the post. For now, make 2 fake comments. The comment should show the comment’s author, body, and date of creation. Think about how you want it to look. What HTML tag should the title go into? How about the body of the post? Can you organize the page into divs?

Here’s an example of what your page might look like:



1. Add some CSS to your file.

Create a file called blog-styles.css

1. In the <head> section of detail.html, add the line:

<link rel="stylesheet" type="text/css" href="blog-styles.css">

In the next lab, we will replace the fake blog post with real data from your database!

**HTML Reference:**

Adapted from <http://www.w3schools.com/tags/default.asp>

|  |  |
| --- | --- |
| <html> | Surrounds all html code |
| <head> | Info about page; does not display in browser |
| <title> | Goes in header, tells browser what to call the page |
| <body> | Surrounds page content that will be displayed |
| <a href=”example.com”>Text of link</a> | link |
| <div> | Division of page, used to divide page into sections |
| <form> | Form for user input |
| <input type=”text” name=”name> | Form element. Types include button, checkbox, date, radio, text, submit. |
| <h1> - <h6> | Headers of different sizes |
| <img src=”imageurl.jpg”> | Image |
| <ul> | Unordered list |
| <ol> | Ordered list |
| <li> | List item |
| <p> | Paragraph |

**CSS Reference:**

Properties:

background-color: color-name Background color

background-image: url('image.gif'); Background image

color: color-name Text color

text-align: center/left/right; Text alignment

font-family:"Times New Roman", Times, serif; Font

font-size:40px; Font size

ul {list-style-type: circle/square;} bullet style

padding: 0px space around element

margin: 0px space around element

Selectors:

#ex1 Selects HTML element with id=ex1

.ex2 Selects HTML elements with class=ex2

li Selects HTML elements that are li’s. You can do this for any element, such as li, p, div, ul...

**CSS example:**

body {

background-color: yellow;

}

p {

color: blue;

background-color: #d0e4fe;

}

h1 {

font-family: “Times New Roman”;

text-align: center;

}