

Lecture 2

John C. Pyun

Spring 2016

1 Before we begin...

My github account: <http://github.com/jcpyun>

I will be uploading all my codes to github for you to download. If you do not know how to use github, please email me at jcp@andrew.cmu.edu

Before we get started, there is one more thing we must do to our virtual environment.

By default, virtual environment is directory dependent. What this means is that the moment you move your virtual environment folder to a different folder, all your packages in your virtual environment will get damaged.

There is a way to fix this, but depending on your machine, it might or might not work. (ex: It does not work for my machine.)

If you are one of the lucky few, try the following command:

```
# In the shell
$ virtualenv --relocatable YOUR_FOLDER_NAME
```

if that works, then great!

Also bear in mind that changing the name of your folder will also affect your virtual env. So, before you start your django project, make sure you name it and store it at a directory that you will never change.

2 What are these files in Django?

Recall that in our last lecture, we have created our own Django folder using the command.

```
# In the shell
# You can substitute "helloworld" with a different name if you want to
$ django-admin startproject helloworld
```

Recall that this created lots of files and folders. Also recall that we have run the following command on the terminal to start our django server:

```
# In the shell
$ python manage.py runserver
```

Recall that this activated the Django server. Now, we will talk about other manage.py commands.

Here are some of the manage.py commands. Please DO NOT try any of these yet. We will go over them in lecture 3.

```
# In the shell
$ python manage.py createsuperuser
$ python manage.py makemigrations
$ python manage.py migrate
$ python manage.py startapp APPNAME
```

Now, before we begin, we must do a few things in our virtual environment folder. In terms of style, django programmers use the base folder as "src". Currently, in our virtual environment folder, we have 4 folders:

bin, include, lib, and helloworld.

let's rename our helloworld folder to "src" instead

So now, we should have 4 folders. bin,include,lib, and src.

Now, drag and drop this "src" folder to the "sublime text" icon. This should automatically open the sublime text program with all the components inside your src folder.

Now, lets go inside our src folder. You should have a helloworld folder and a file named manage.py.

Now, we are going to create our very first html site to our django framework. to do that, we must create our own django app within our django project. Let's create our frontpage django app by typing the following command:

```
# In the shell
$ python manage.py startapp frontpage
```

Now, this should create a folder named "frontpage" in our src folder. So, in our src folder, we must have 2 folders: frontpage and helloworld and it should also have a file named manage.py

We have covered how MVC (Model, View, Controller) works in standard webdev. Also, we have covered how MVC works in Django format. I will just briefly explain how everything works.

urls.py stores all the list of url you will be using for you django project. So, if you want to set up a url called "home" or url called "yourprofile" (for example <https://www.facebook.com/yourprofile>). Unless you specifically state

the list of urls to this file, your django app will have no idea that these urls exist.

views.py control the template arrangement. Once it receives the valid url from urls.py, it tries to find the corresponding template to show. We will be using views.py to load up the correct html/css/javascript code to upload.

We will later go over models.py so don't worry about it yet.

Now, let's setup our first urls.py code.

go inside helloworld folder and open up urls.py through text editor. Now, include the following lines of code:

```
# inside urlpatterns, add the following.
url(r'^$', 'homepage.views.home', name='home'),

#as a result, it should look like the following :
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', 'homepage.views.home', name='home'),
]
```

What "`^$`" means is that it is a wildcard url.

This means that any unknown url will also be directed to your homepage.

What `homepage.views.home` does is that it opens the directory "homepage" and opens the "views.py" file and opens the "home" function that we will code after this.

now, let's edit views.py file that is located inside our homepage directory.

```
#inside views.py
def home(request):
    template="home.html"
    context={
    }
    return render(request,template,context)
```

Aside: make sure you either stick with tab format or spacing format. Also, make sure you save all these files.

Now, let us try to run our code. Go to the folder that contains `manage.py`. and run the server again with following command:

```
#on the shell
python manage.py runserver
```

Now, go to `http://127.0.0.1:8000/` and you will notice that you have encountered "TemplateDoesNotExist" error.

We will go over how to read these error messages in more detail next class, but if you look carefully you will realize that this error is caused because our django server cannot find where our home.html is located.

We have not created home.html file yet, so let us create one right now.

Go to our src folder. Currently, we have 2 folders: helloworld and homepage. We will now create another folder named "templates". Create a folder named "templates"

now, go inside templates folder, and create a file named "home.html" using your text editor. We will now edit this file.

in home.html, write the following html code:

```
#html code
<h1> Hello World! </h1>
```

Now, click save. Let us now test this again! Go back to the folder that contains manage.py and run the server again (do the whole python manage.py runserver thingy...) and test it on 127.0.0.1:8000 again.

You will notice again that it has created an error AGAIN (oh no!)

This is because although we have created a "home.html" file, our django project has no idea where this file is located!

Let's help our django project locate home.html. To do so, we must go to settings.py. This file is located inside our helloworld folder. Open this file. Let's go one by one and see which line we must edit to tell django where our templates files are located.

```
#settings.py
BASE_DIR: #this line of code tells our django app where our "manage.py"
          file is located.

SECRET_KEY: #this is some magical witchcraft secret stuff that django
            does. Don't touch it and don't share it!

DEBUG: #All your error messages that you've seen happens because this
       variable is set as "True". In the production phase, you will turn
       this False.

ALLOWED_HOSTS: #When you deploy your webapp and you want to use your own
               Domain name, you will add them here. But for now, leave it as it is.
```


That's it for this lecture!

If you are stuck, please write a piazza post and write an e-mail me at jcp@andrew.cmu.edu