Todd Smith
tbsmith@fullsail.com

Welcome to SSL Day 7!

Intro to Frameworks

Day 7

Server-Side Languages

PHP: CodeIgniter

EllisLab

clone this:

https://github.com/EllisLab/CodeIgniter.git

The files in the CodeIgniter folder will become your new web root

make day7.php.com now

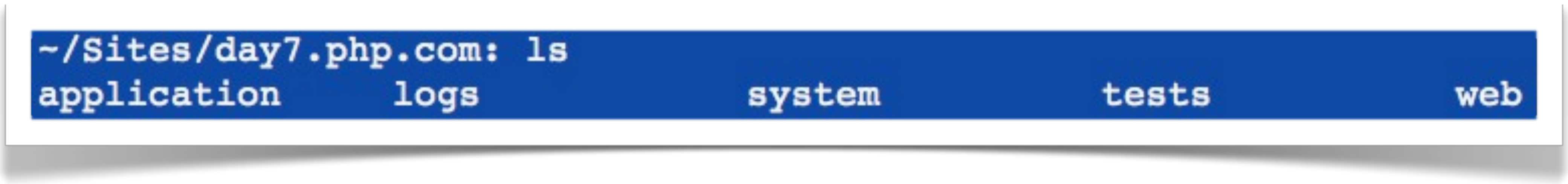
Day 7

Server-Side Languages

Isolating the web user

Delete the extra stuff and make the folders web/ and logs/

Move the index.php file into web/



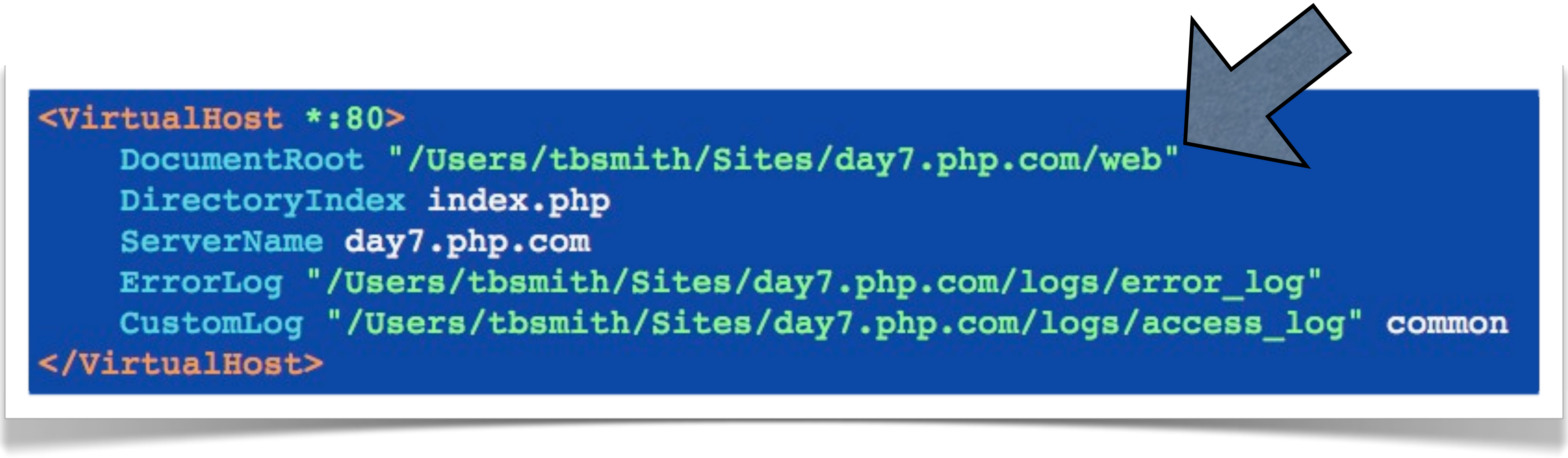
```
~/Sites/day7.php.com: ls
application       logs              system            tests             web
```

Day 7

Server-Side Languages

Isolating the web user

In your virtual hosting file, set the webroot to /web



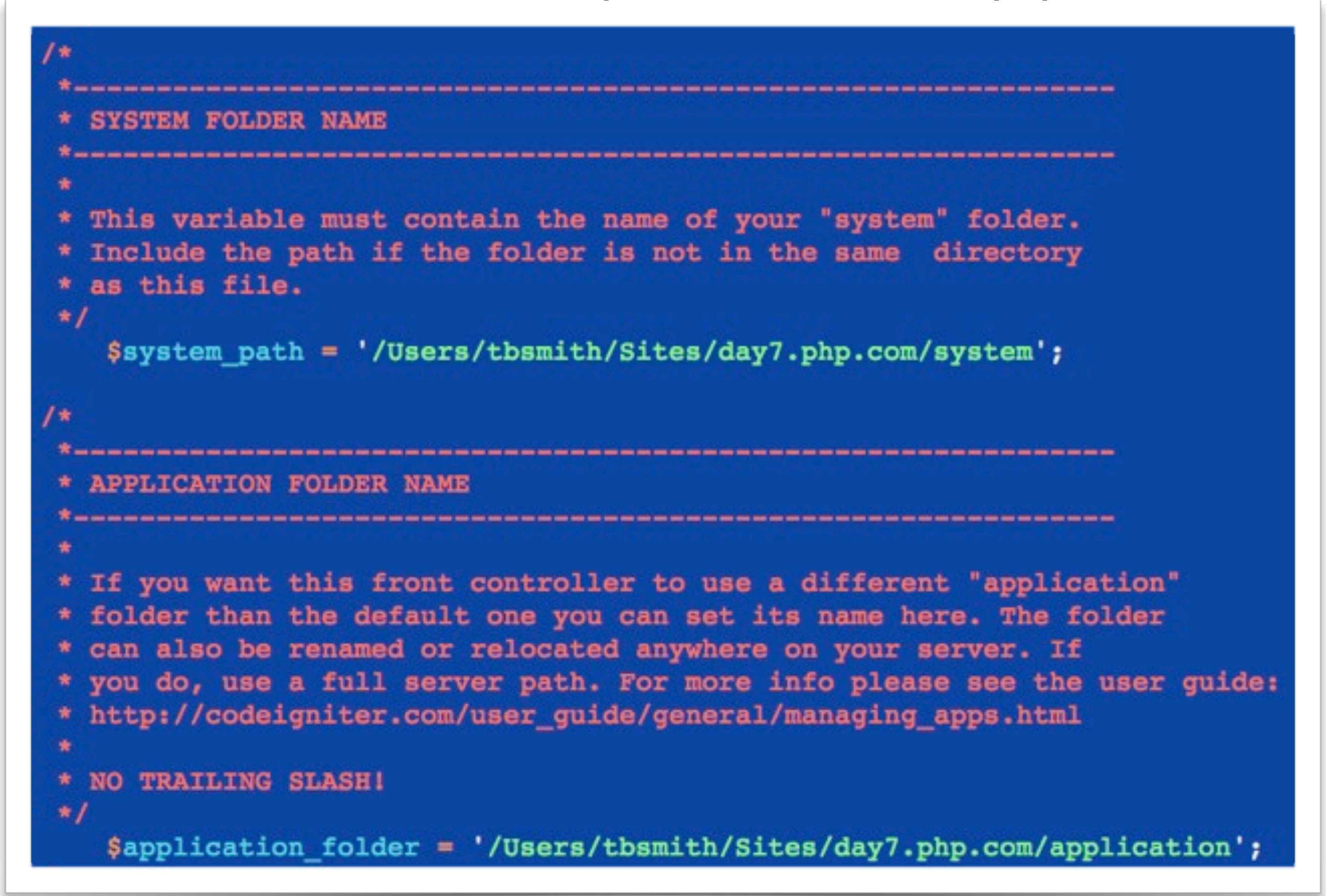
```
<VirtualHost *:80>
    DocumentRoot "/Users/tbsmith/Sites/day7.php.com/web"
    DirectoryIndex index.php
    ServerName day7.php.com
    ErrorLog "/Users/tbsmith/Sites/day7.php.com/logs/error_log"
    CustomLog "/Users/tbsmith/Sites/day7.php.com/logs/access_log" common
</VirtualHost>
```

Day 7

Server-Side Languages

Isolating the web user

Hard-code these paths in /web/index.php

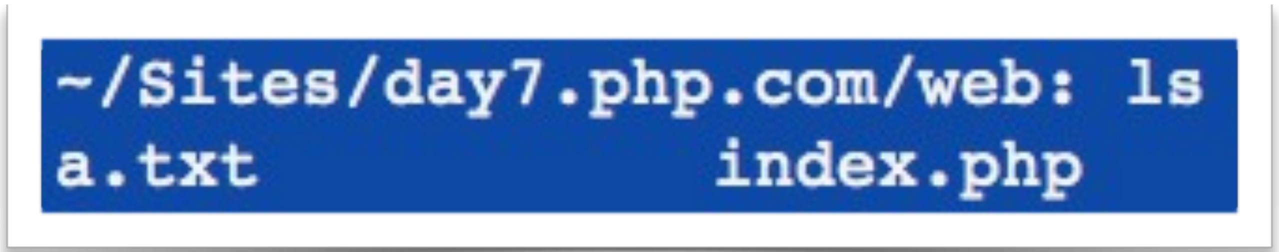


```
/*
 *---------------------------------------------------------------
 * SYSTEM FOLDER NAME
 *---------------------------------------------------------------
 *
 * This variable must contain the name of your "system" folder.
 * Include the path if the folder is not in the same  directory
 * as this file.
 */
    $system_path = '/Users/tbsmith/Sites/day7.php.com/system';

/*
 *---------------------------------------------------------------
 * APPLICATION FOLDER NAME
 *---------------------------------------------------------------
 *
 * If you want this front controller to use a different "application"
 * folder than the default one you can set its name here. The folder
 * can also be renamed or relocated anywhere on your server. If
 * you do, use a full server path. For more info please see the user guide:
 * http://codeigniter.com/user_guide/general/managing_apps.html
 *
 * NO TRAILING SLASH!
 */
    $application_folder = '/Users/tbsmith/Sites/day7.php.com/application';
```

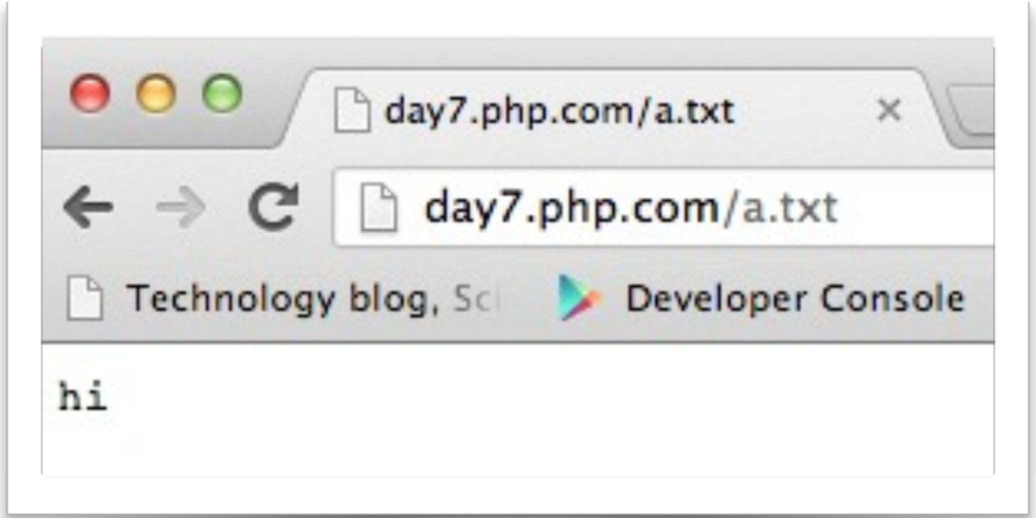
Day 7

Isolating the web user

Now users at day7.php.com/ will be restricted to the web directory.



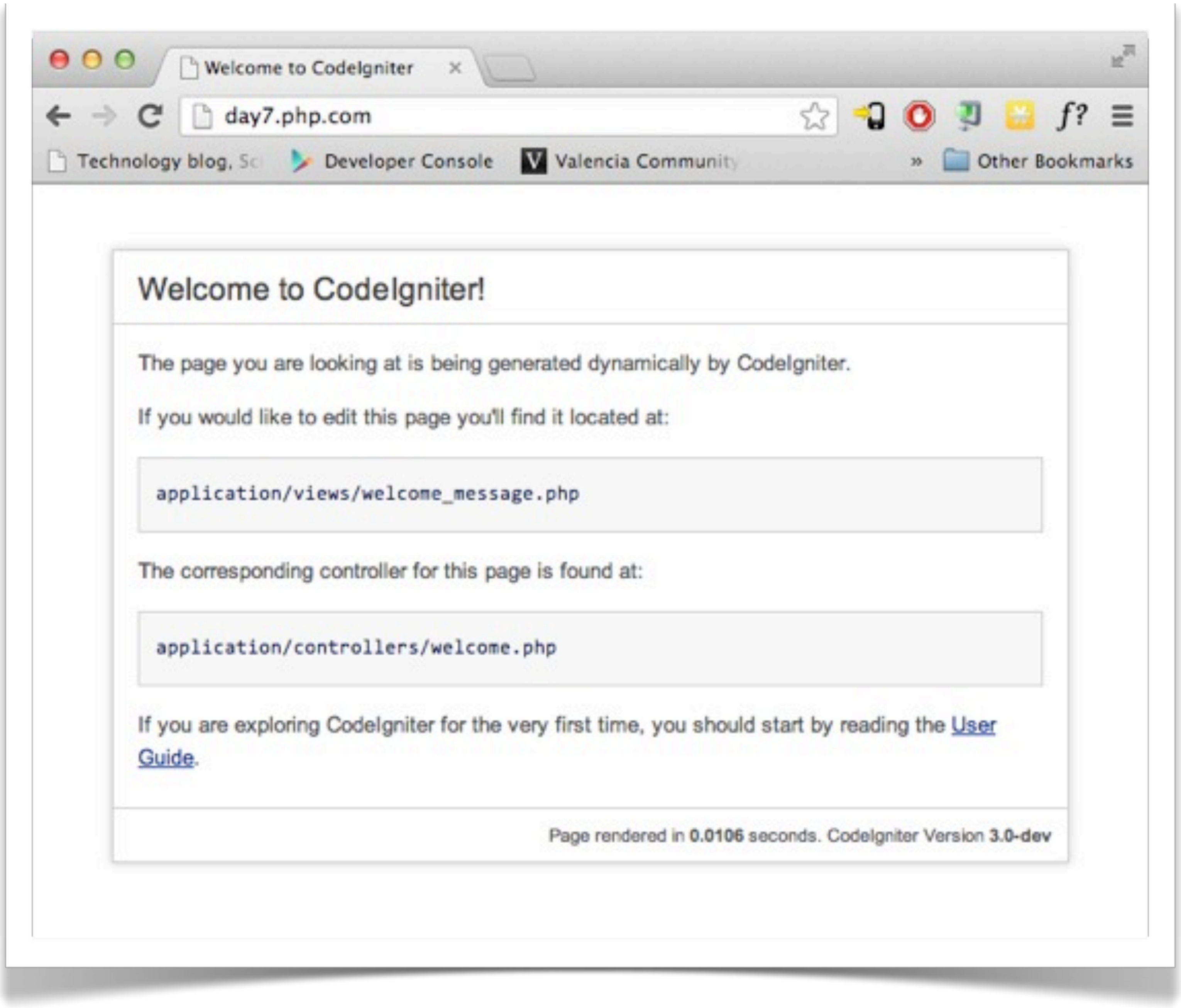
```
~/Sites/day7.php.com/web: ls
a.txt                  index.php
```



```
hi
```

Day 7

Test the site



Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

```
application/views/welcome_message.php
```

The corresponding controller for this page is found at:

```
application/controllers/welcome.php
```

If you are exploring CodeIgniter for the very first time, you should start by reading the User Guide.

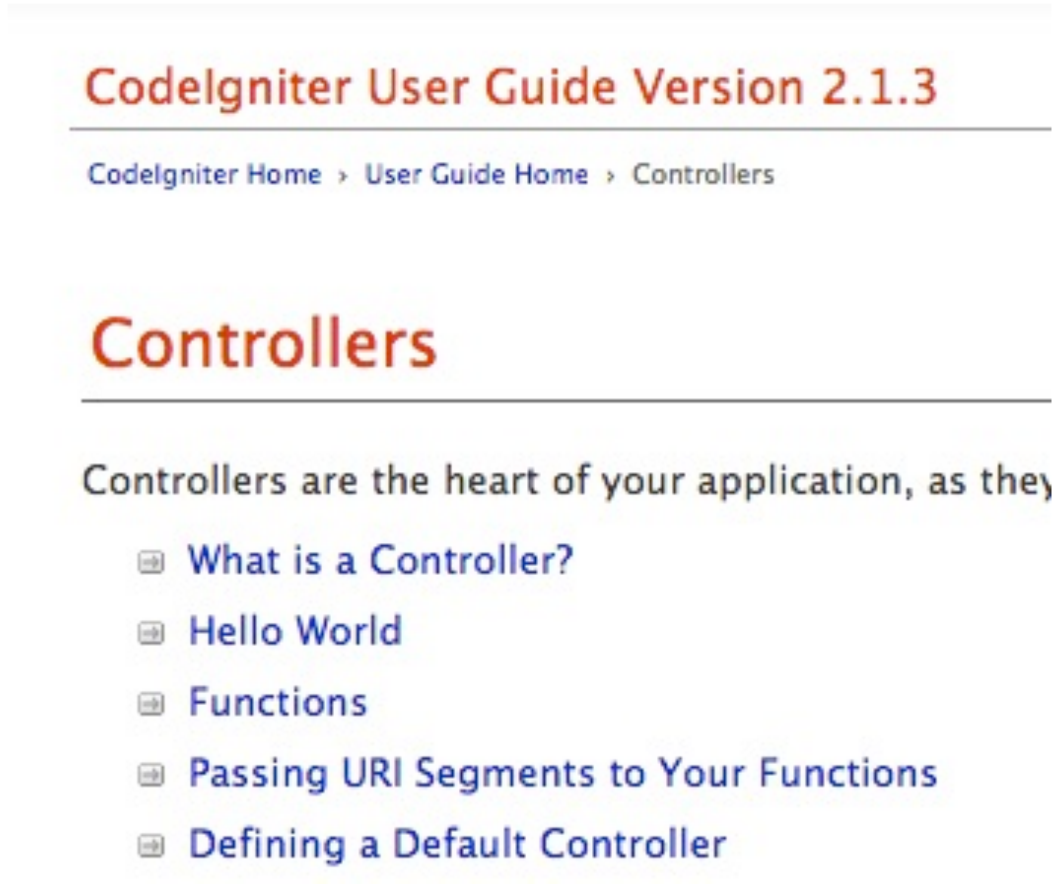
Page rendered in **0.0106** seconds. CodeIgniter Version **3.0-dev**

Server-Side Languages

Understanding CI Controllers

Search the web for "codeigniter controllers"

Follow the first 5 sections of this tutorial.



CodeIgniter User Guide Version 2.1.3

CodeIgniter Home › User Guide Home › Controllers

Controllers

Controllers are the heart of your application, as they

- What is a Controller?
- Hello World
- Functions
- Passing URI Segments to Your Functions
- Defining a Default Controller

Day 7

Server-Side Languages

Python: Django

PyCharm

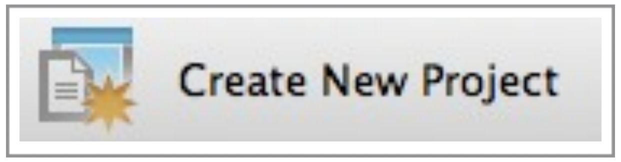
download this:

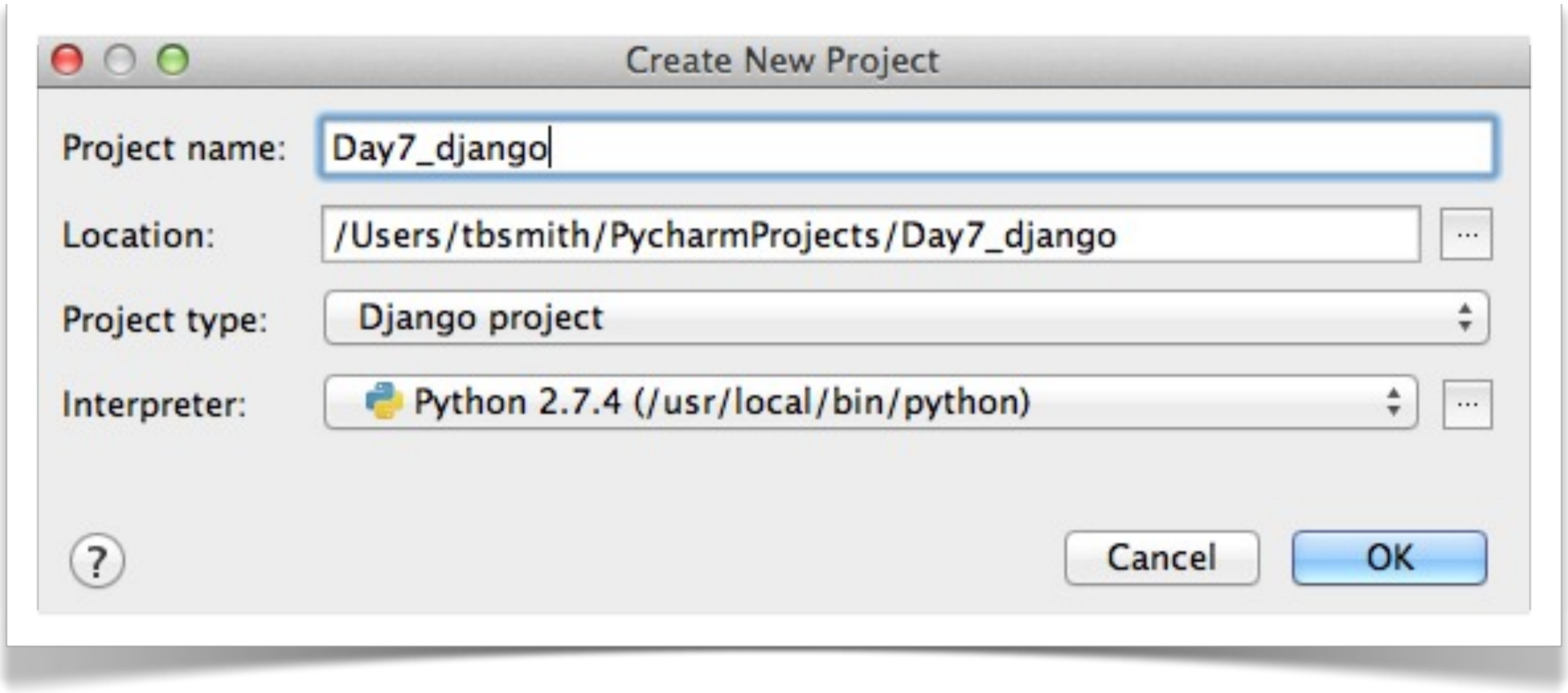
http://www.jetbrains.com/pycharm/download/index.html

The PyCharm IDE will handle our directory structure

Day 7

Starting the PyCharm Project





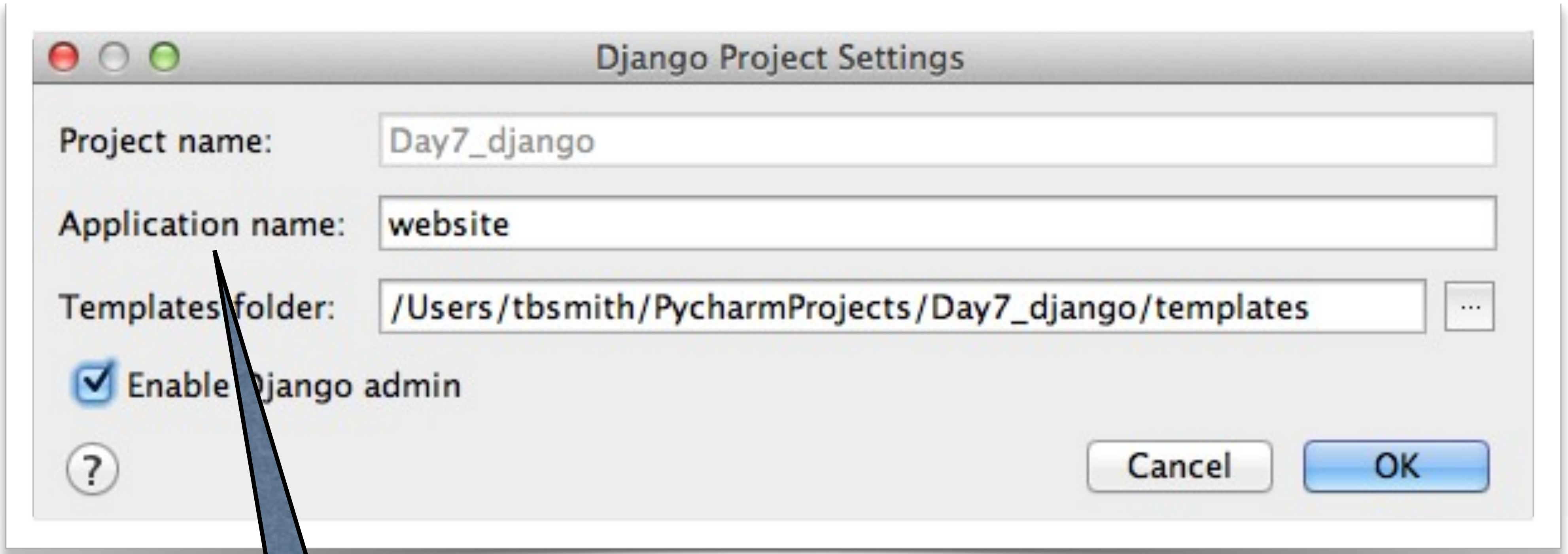
Project name: Day7_django

Location: /Users/tbsmith/PycharmProjects/Day7_django

Project type: Django project

Interpreter: Python 2.7.4 (/usr/local/bin/python)

Starting the PyCharm Project



Project name: Day7_django

Application name: website

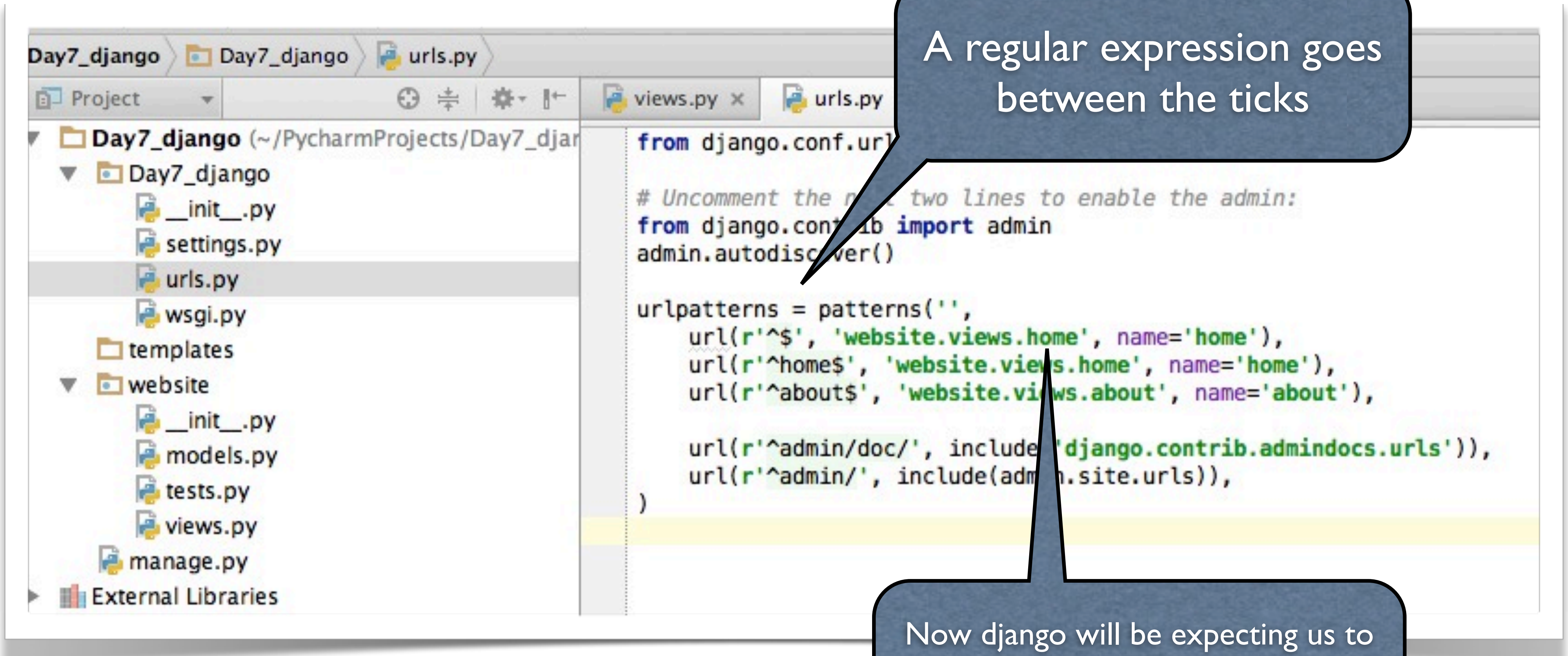
Templates folder: /Users/tbsmith/PycharmProjects/Day7_django/templates

[✓] Enable Django admin

You may set up different apps for a website's blog, JSON api, admin section, shopping cart, analytics, user accounts...

The idea is to keep your code modular and reusable.

Set up your urls



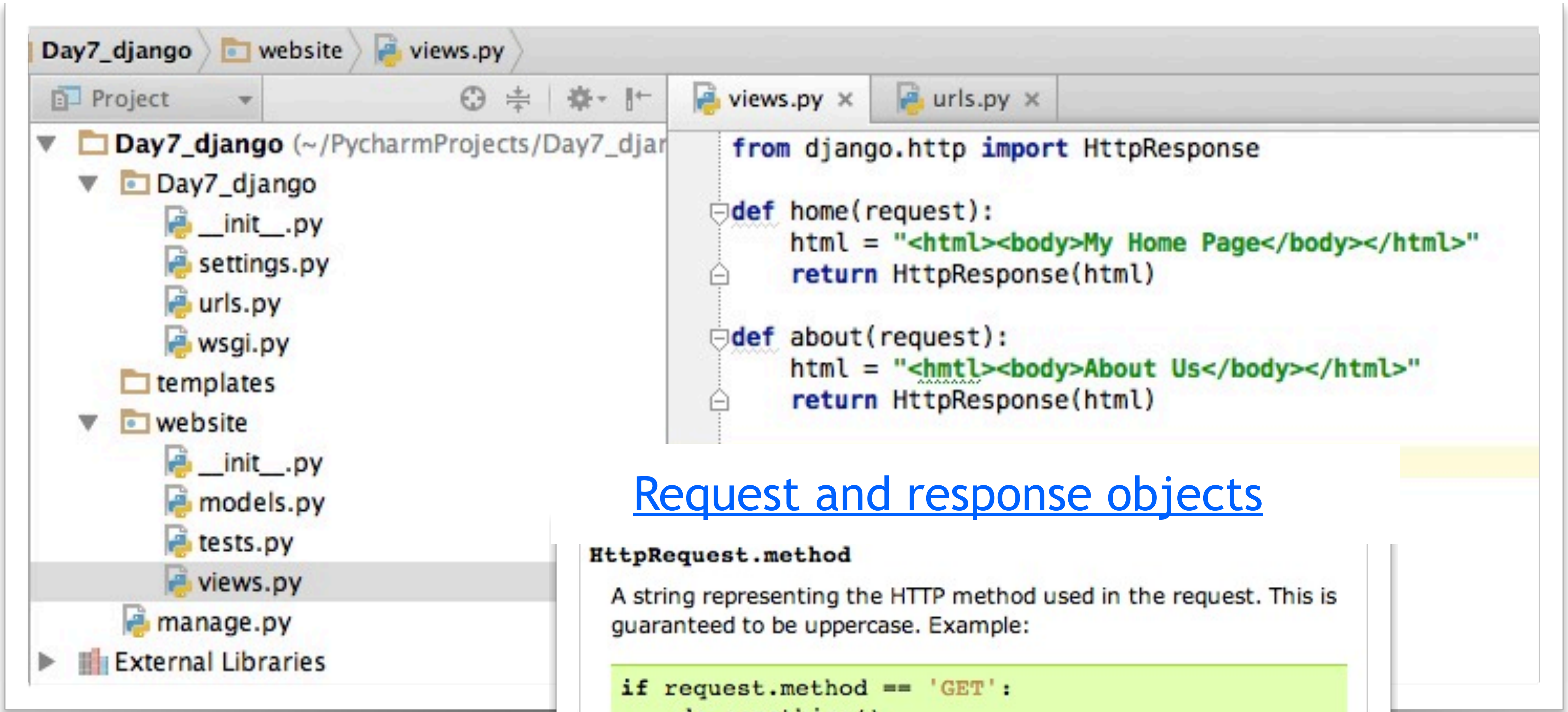
```
from django.conf.url

# Uncomment the n    two lines to enable the admin:
from django.con   ib import admin
admin.autodisc  ver()

urlpatterns = patterns('',
    url(r'^$', 'website.views.home', name='home'),
    url(r'^home$', 'website.vie s.home', name='home'),
    url(r'^about$', 'website.vi ws.about', name='about'),

    url(r'^admin/doc/', include 'django.contrib.admindocs.urls')),
    url(r'^admin/', include(adm .site.urls)),
)
```

A regular expression goes between the ticks

Now django will be expecting us to have a home class in the views.py file in the website app

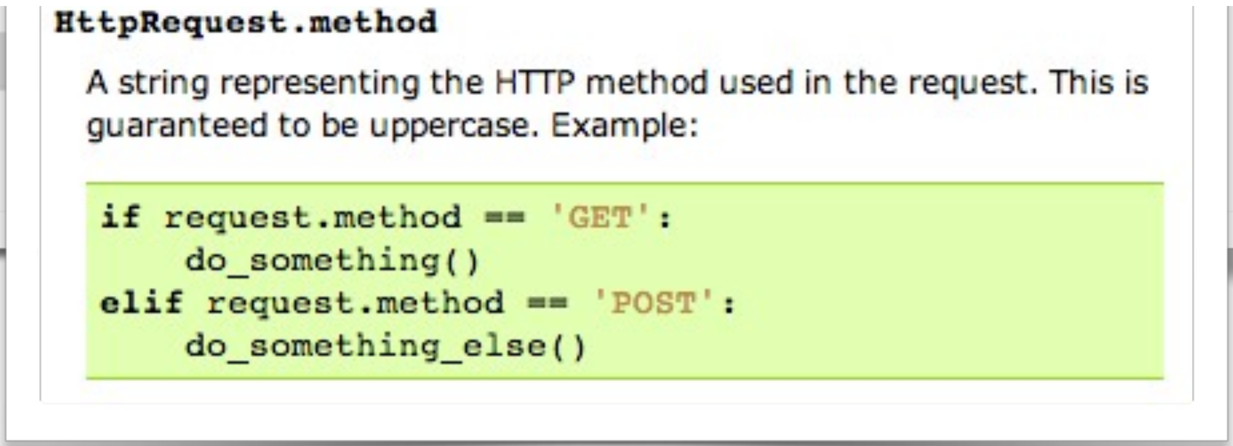
Server-Side Languages

Set up your views



```
from django.http import HttpResponse

def home(request):
    html = "<html><body>My Home Page</body></html>"
    return HttpResponse(html)

def about(request):
    html = "<hmtl><body>About Us</body></html>"
    return HttpResponse(html)
```

[Request and response objects]



HttpRequest.method

A string representing the HTTP method used in the request. This is guaranteed to be uppercase. Example:

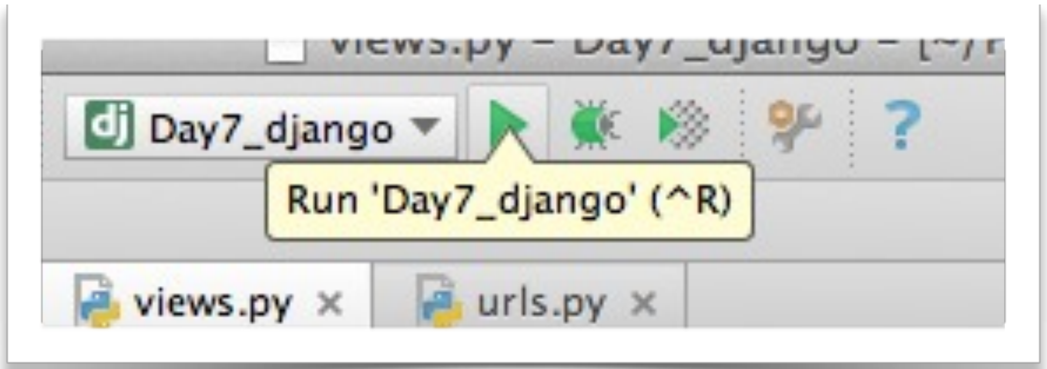
```
if request.method == 'GET':
    do_something()
elif request.method == 'POST':
    do_something_else()
```

Day 7

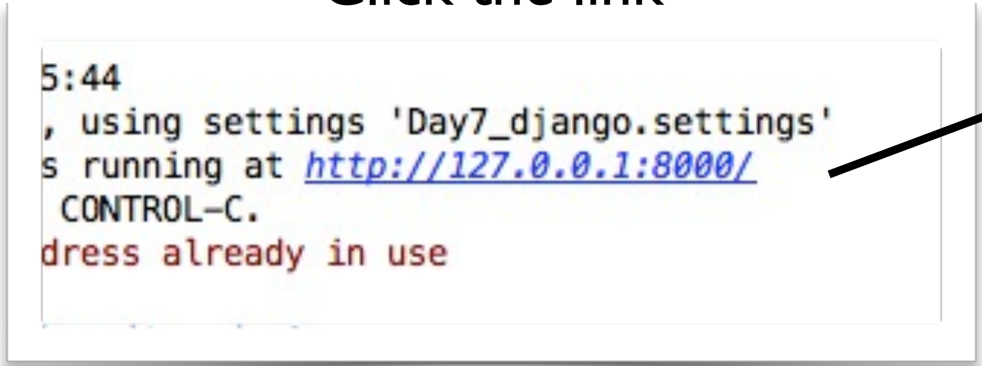
Server-Side Languages

Test your views

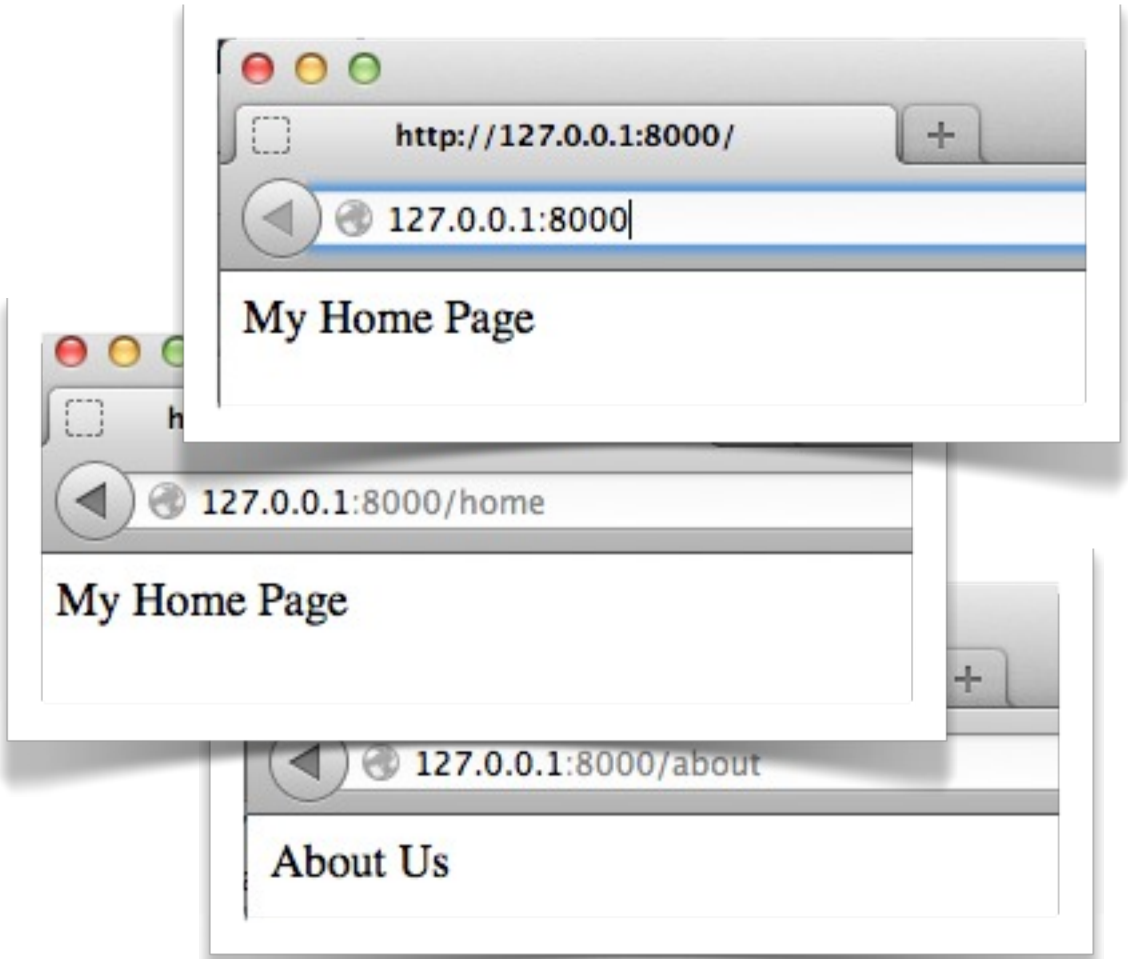
Click play



Click the link

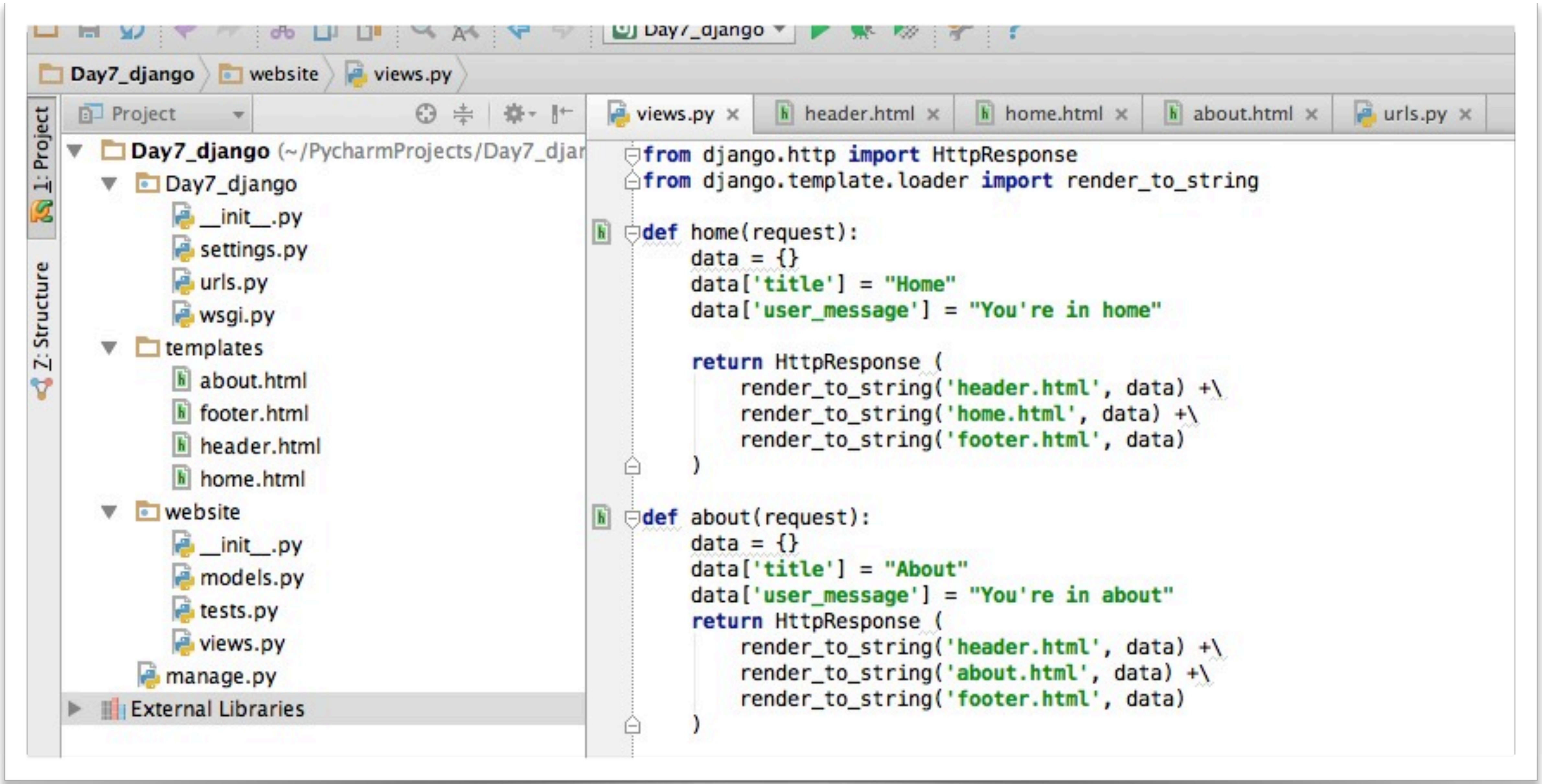


```
5:44
, using settings 'Day7_django.settings'
s running at http://127.0.0.1:8000/
 CONTROL-C.
dress already in use
```



Day 7

The Templating System

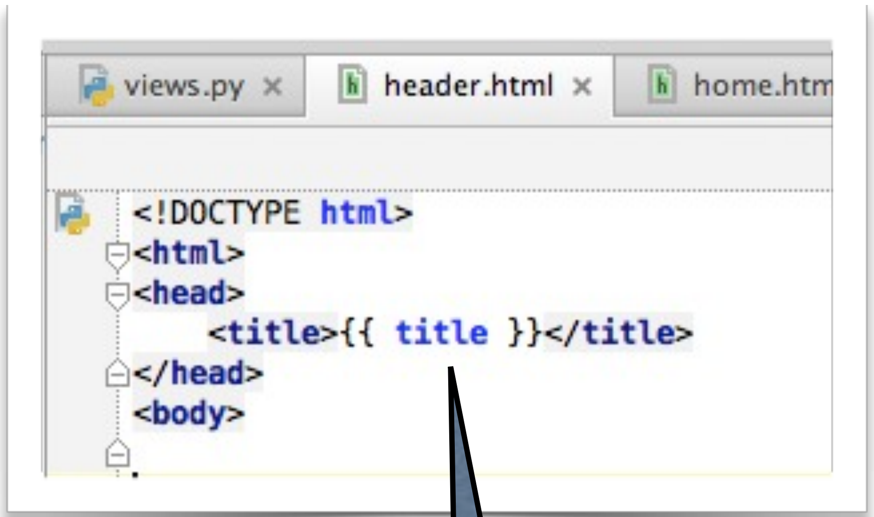


```python
from django.http import HttpResponse
from django.template.loader import render_to_string

def home(request):
    data = {}
    data['title'] = "Home"
    data['user_message'] = "You're in home"

    return HttpResponse (
        render_to_string('header.html', data) +\
        render_to_string('home.html', data) +\
        render_to_string('footer.html', data)
    )

def about(request):
    data = {}
    data['title'] = "About"
    data['user_message'] = "You're in about"
    return HttpResponse (
        render_to_string('header.html', data) +\
        render_to_string('about.html', data) +\
        render_to_string('footer.html', data)
    )
```

Day 7

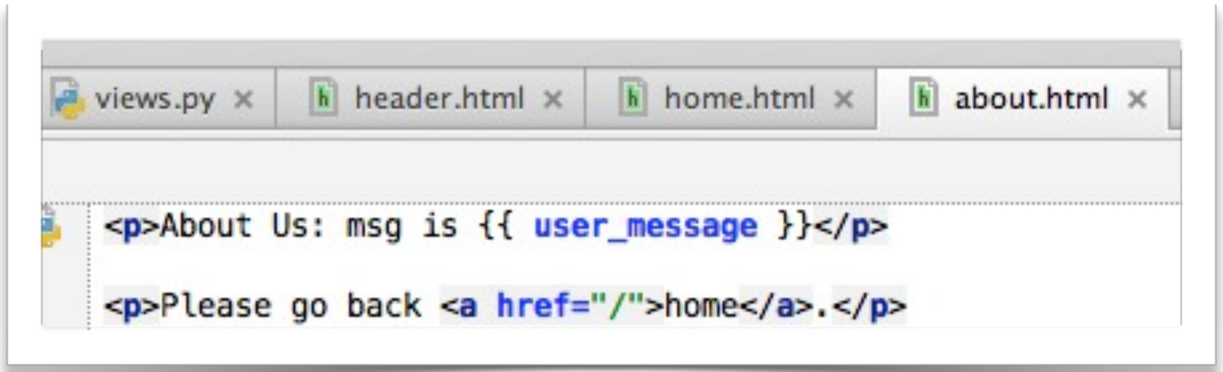
Server-Side Languages

The Templating System

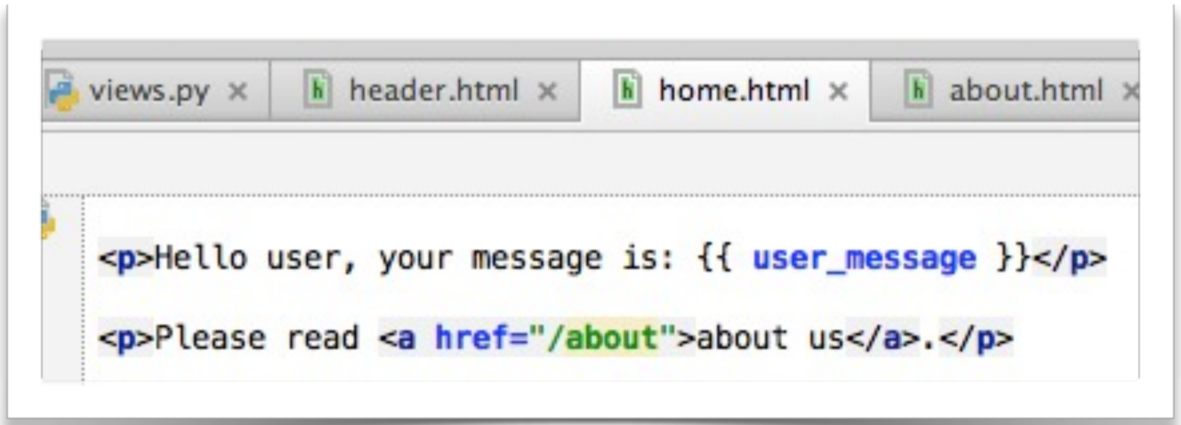


```
<!DOCTYPE html>
<html>
<head>
    <title>{{ title }}</title>
</head>
<body>
```

These are the keys in the dictionary that was passed



```
<p>About Us: msg is {{ user_message }}</p>

<p>Please go back <a href="/">home</a>.</p>
```



```
<p>Hello user, your message is: {{ user_message }}</p>

<p>Please read <a href="/about">about us</a>.</p>
```

Link to docs

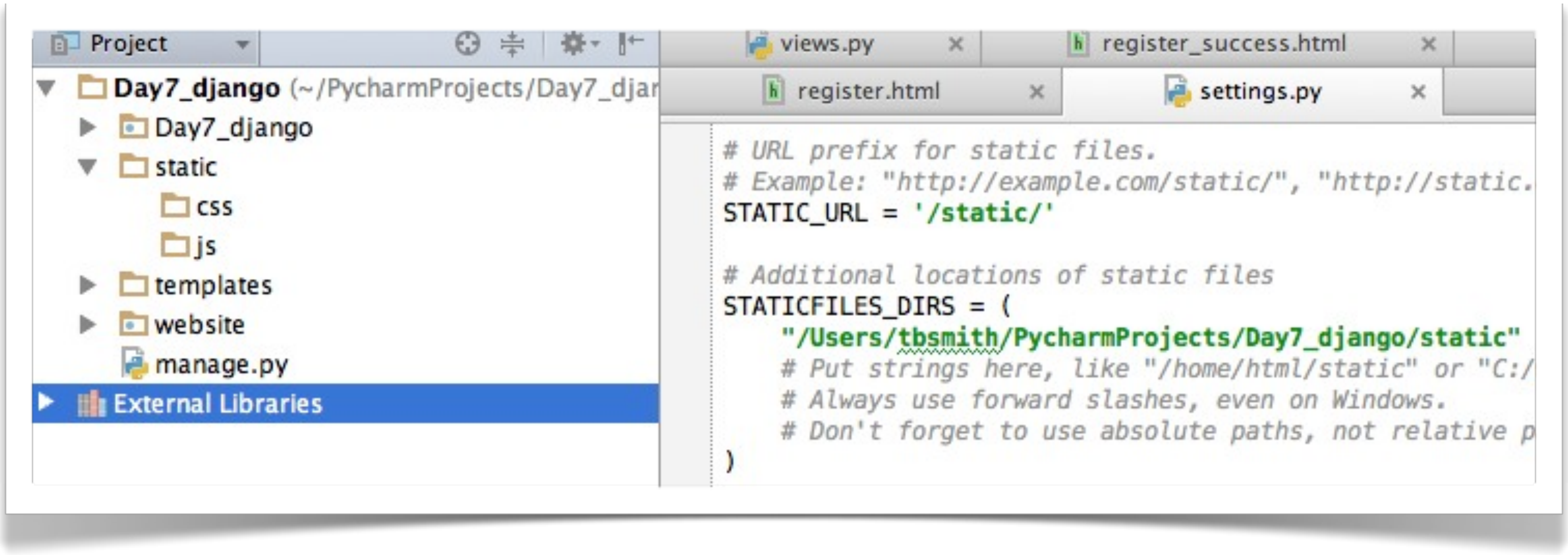
Day 7

Server-Side Languages

Static Files

[docs]

Set your static url and directories
in the settings.py file



```
# URL prefix for static files.
# Example: "http://example.com/static/", "http://static.
STATIC_URL = '/static/'

# Additional locations of static files
STATICFILES_DIRS = (
    "/Users/tbsmith/PycharmProjects/Day7_django/static"
    # Put strings here, like "/home/html/static" or "C:/
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative p
)
```

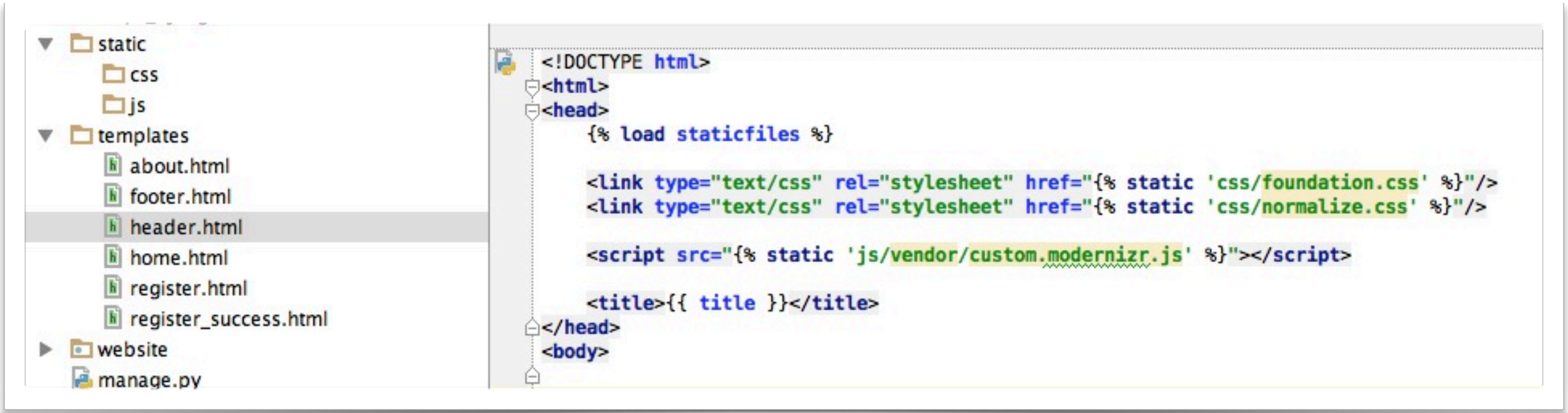
Day 7

Server-Side Languages

Static Files

docs

Use static template tags

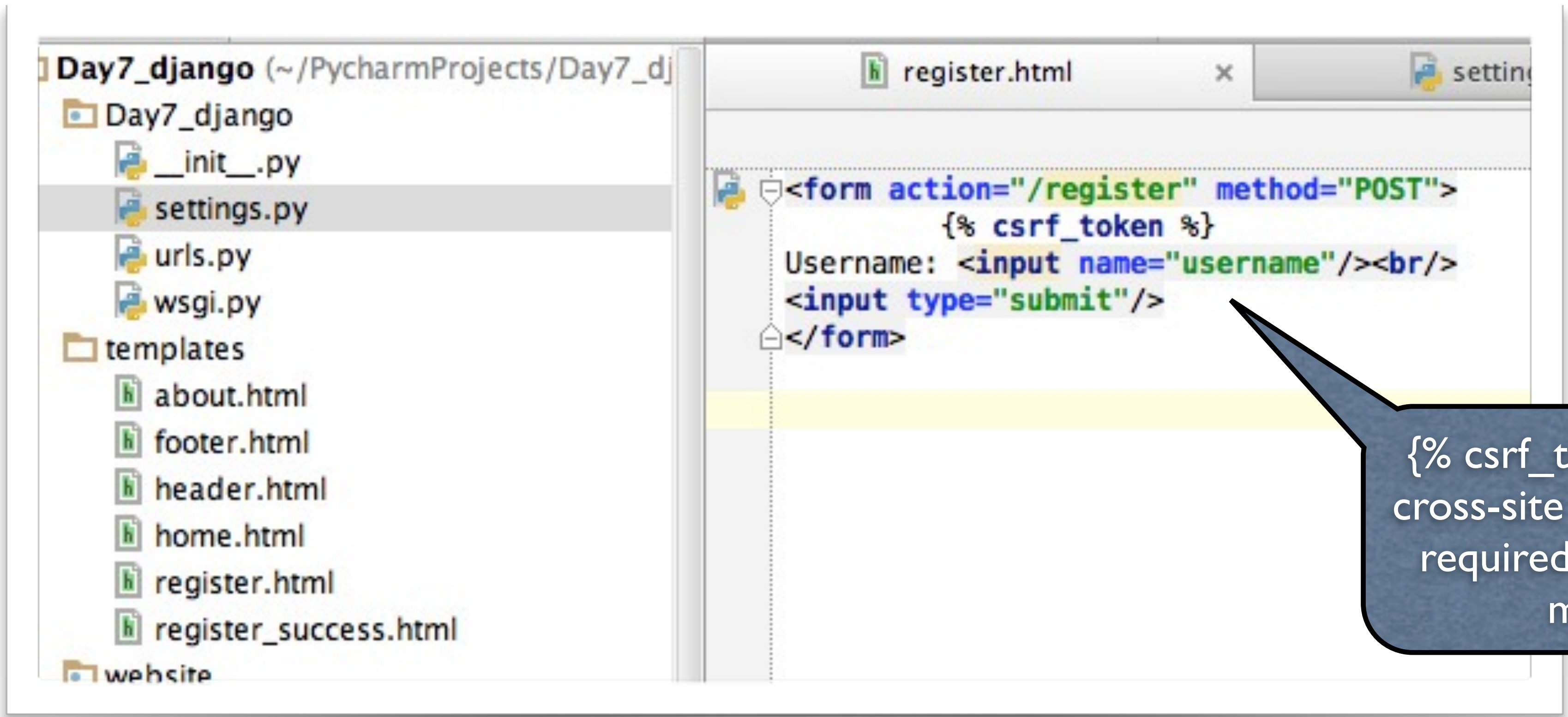


```
▼ static
    css
    js
▼ templates
    about.html
    footer.html
    header.html
    home.html
    register.html
    register_success.html
▶ website
  manage.py
```

```
<!DOCTYPE html>
<html>
<head>
    {% load staticfiles %}

    <link type="text/css" rel="stylesheet" href="{% static 'css/foundation.css' %}"/>
    <link type="text/css" rel="stylesheet" href="{% static 'css/normalize.css' %}"/>

    <script src="{% static 'js/vendor/custom.modernizr.js' %}"></script>

    <title>{{ title }}</title>
</head>
<body>
```

Day 7

A Simple Form: templates

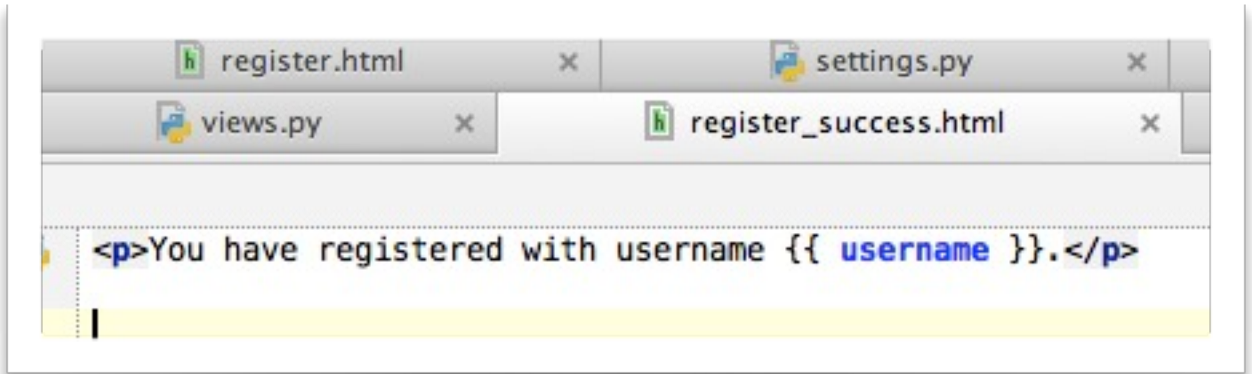


```
Day7_django (~/PycharmProjects/Day7_dj
  Day7_django
    __init__.py
    settings.py
    urls.py
    wsgi.py
  templates
    about.html
    footer.html
    header.html
    home.html
    register.html
    register_success.html
  website
```

register.html

```
<form action="/register" method="POST">
        {% csrf_token %}
Username: <input name="username"/><br/>
<input type="submit"/>
</form>
```

{% csrf_token %} will prevent cross-site request forgeries. It is required for every form with method POST.

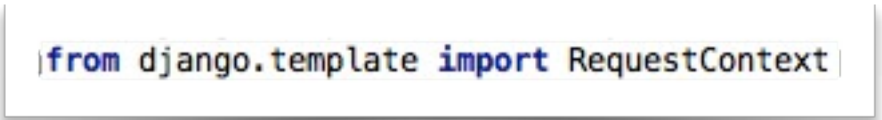


register_success.html

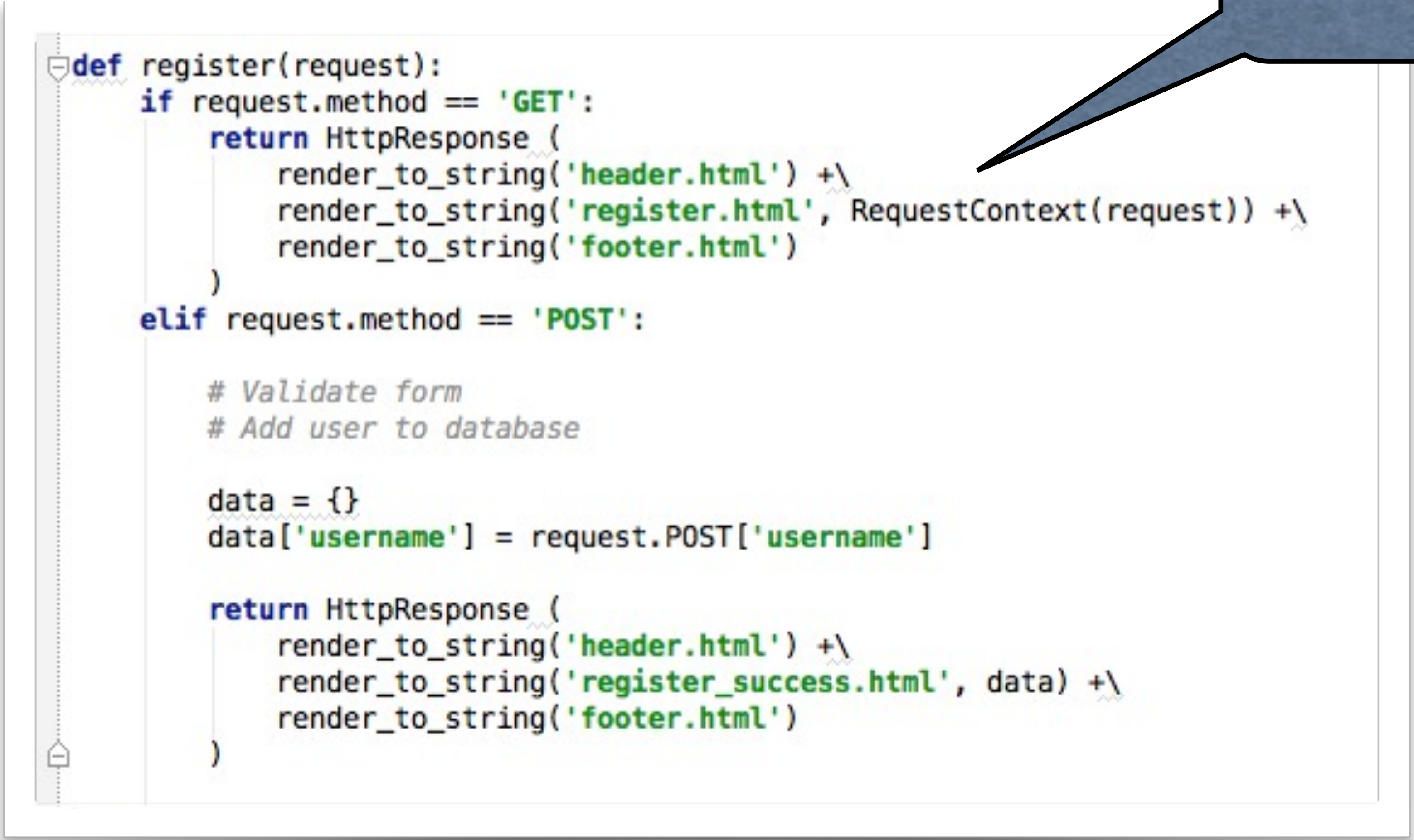
```
<p>You have registered with username {{ username }}.</p>
```

Day 7

A Simple Form: views



```python
from django.template import RequestContext
```

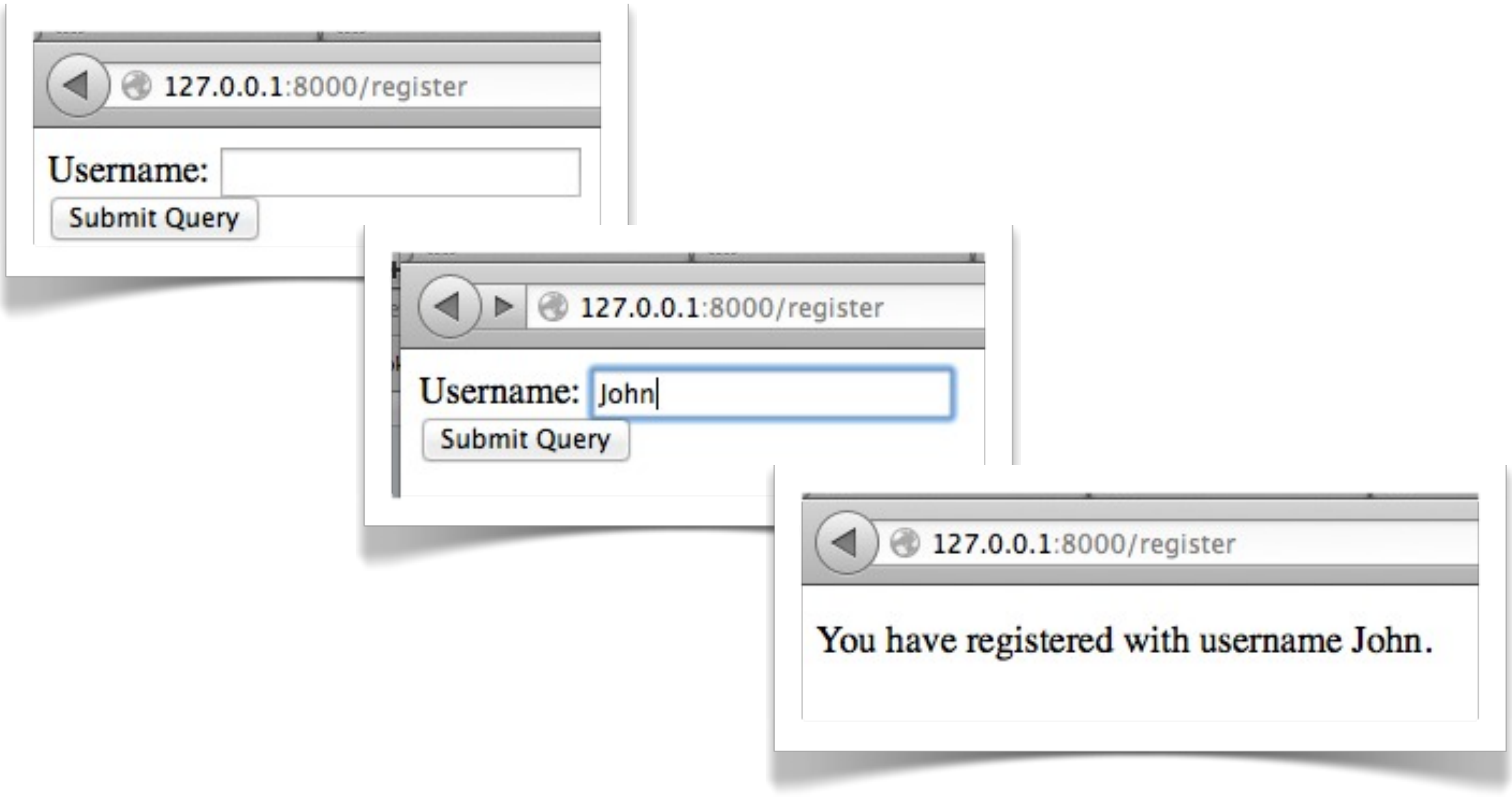


```python
def register(request):
    if request.method == 'GET':
        return HttpResponse (
            render_to_string('header.html') +\
            render_to_string('register.html', RequestContext(request)) +\
            render_to_string('footer.html')
        )
    elif request.method == 'POST':

        # Validate form
        # Add user to database

        data = {}
        data['username'] = request.POST['username']

        return HttpResponse (
            render_to_string('header.html') +\
            render_to_string('register_success.html', data) +\
            render_to_string('footer.html')
        )
```

This should be passed to the templates with forms with method POST.

Day 7

A Simple Form: testing


127.0.0.1:8000/register

Username:

Submit Query

127.0.0.1:8000/register

Username: John

Submit Query

127.0.0.1:8000/register

You have registered with username John.

Day 7

Server-Side Languages

Lab

Do this Django tutorial:

http://lightbird.net/dbe/blog.html

Day 7