CM3035 Advanced Web Development

Lesson 2

Created by Ben Gay

Template folder

Under the Django framework, html files should be stored inside the Templates folder

Activate the Virtual Environment

Create a new project L2proj

Refer to Lesson 1

Use the same **Virtual Environment**

Structure:

Myvenv → **L2proj**

Create a new application L2app

Refer to Lesson 1

Use the same **Virtual Environment**

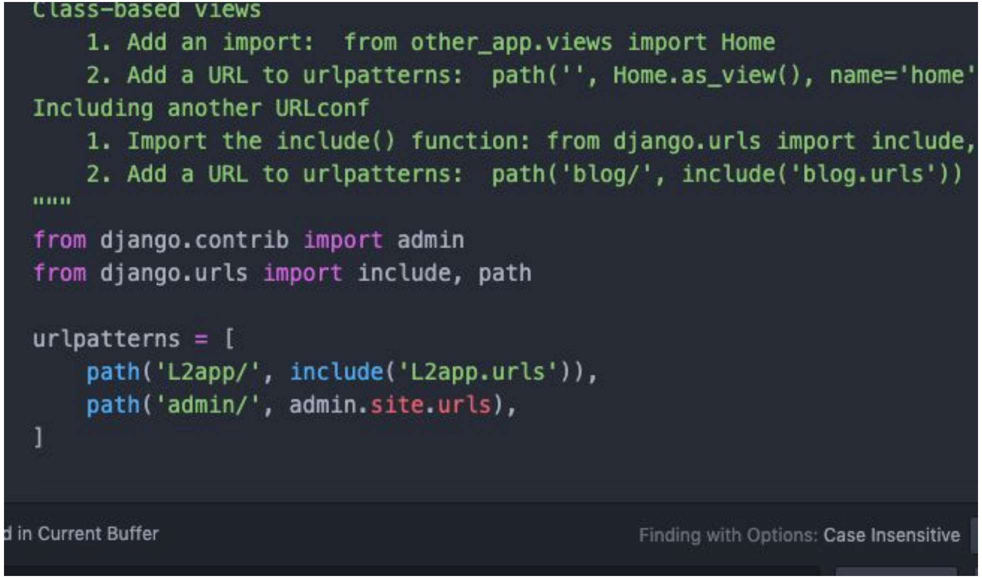
Structure:

Myvenv → L2proj → **L2app**

Update project path

Include L2app.urls into the file L2proj/urls.py

This is to point the project url to application url

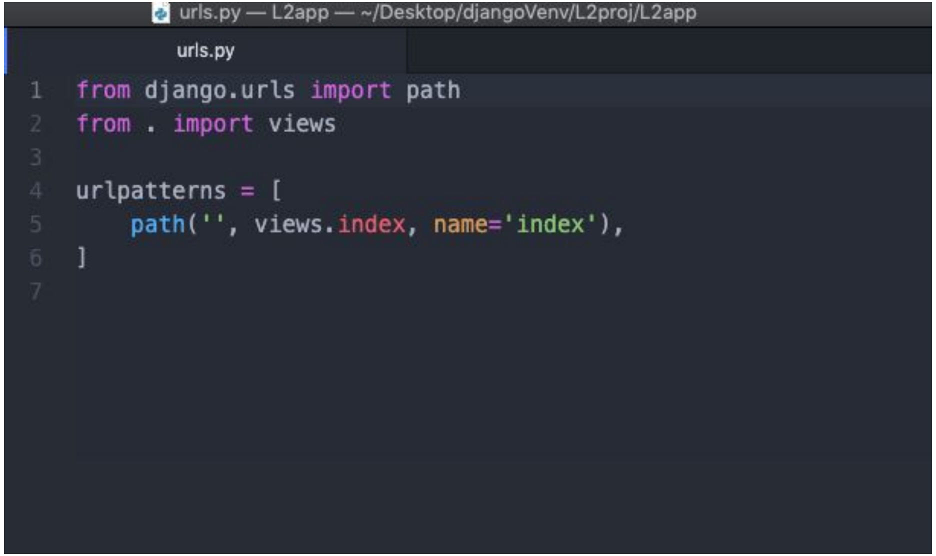


```
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home'
Including another URLconf
    1. Import the include() function: from django.urls import include,
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('L2app/', include('L2app.urls')),
    path('admin/', admin.site.urls),
]
```

Set application path to views

Point url inside file L2app/urls.py to the index function inside L2app/views.py

This is to point the application url to the views



```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
]
```

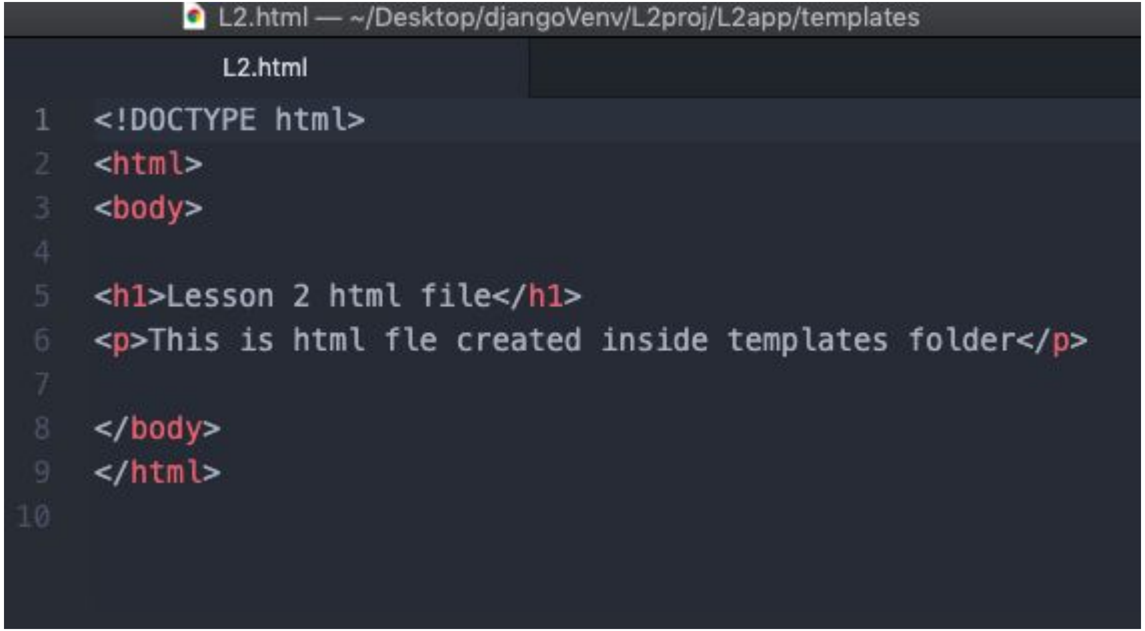
Create html file inside templates

Under Django framework all html files should be stored in the templates folder

Create templates folder inside L2app

Add L2.html file inside templates folder

Content inside L2.html



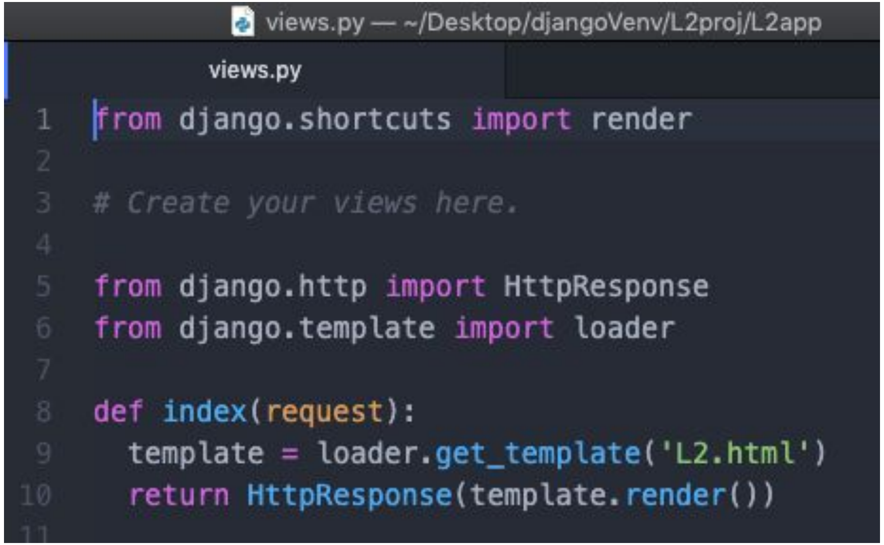
```
<!DOCTYPE html>
<html>
<body>

<h1>Lesson 2 html file</h1>
<p>This is html fle created inside templates folder</p>

</body>
</html>
```

views.py

Set L2.html to a variable

Return the http response to the browser

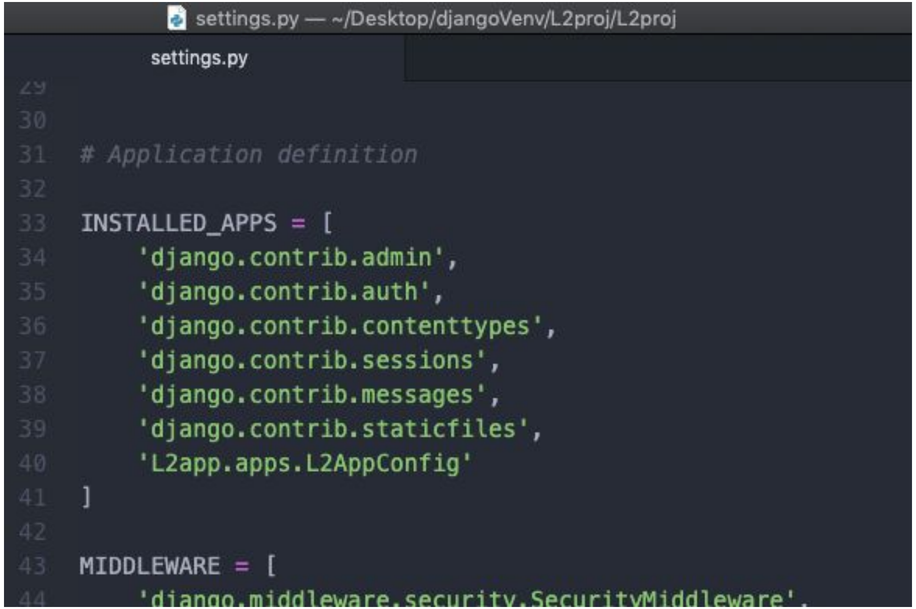
Update to L2app/views.py



```python
from django.shortcuts import render

# Create your views here.

from django.http import HttpResponse
from django.template import loader

def index(request):
  template = loader.get_template('L2.html')
  return HttpResponse(template.render())
```

Configure L2proj/settings.py

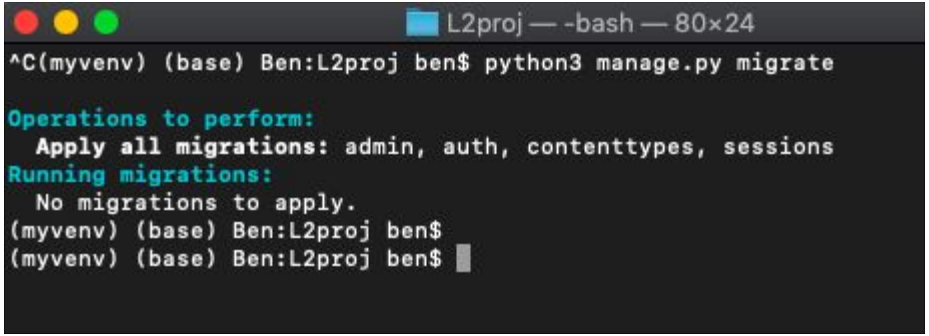
Configure L2app.app.L2AppConfig object into the array called INSTALLED_APPS under file L2proj/settings.py



```


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'L2app.apps.L2AppConfig'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
```

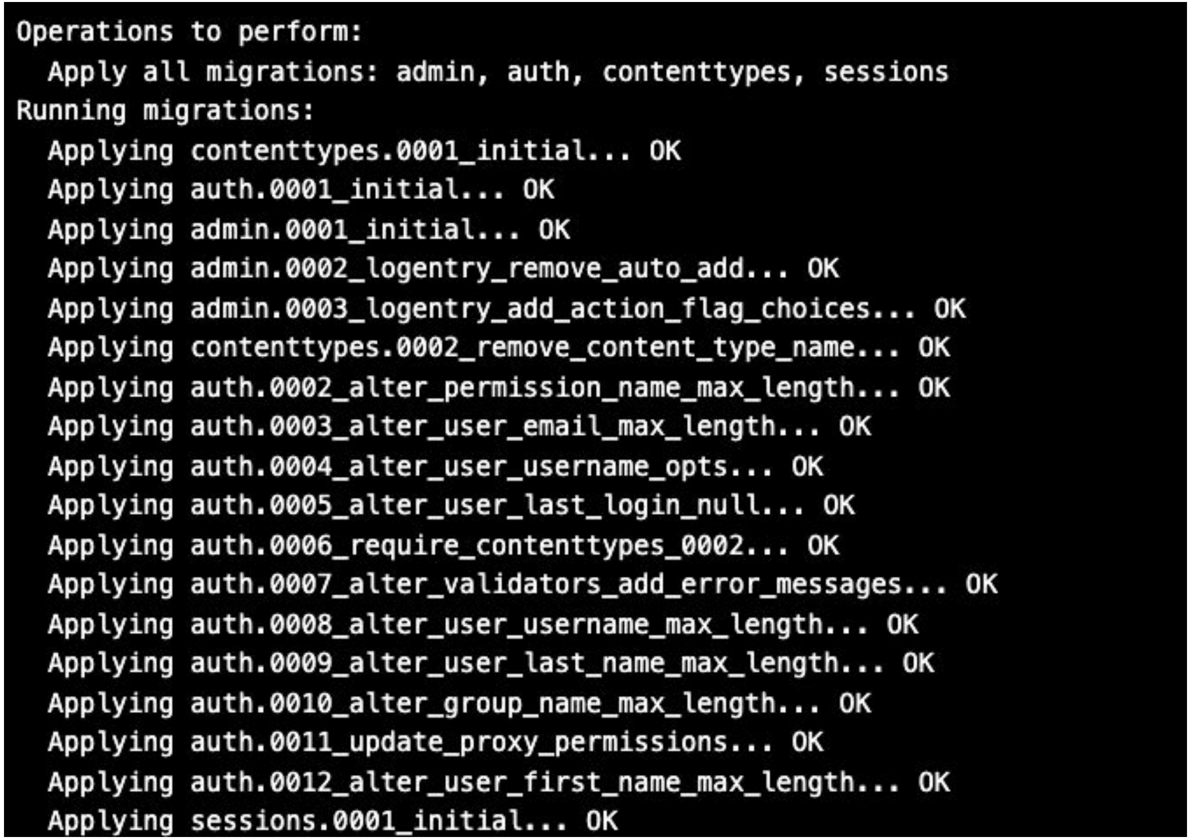
Propagating Changes

>>>python3 manage.py migrate



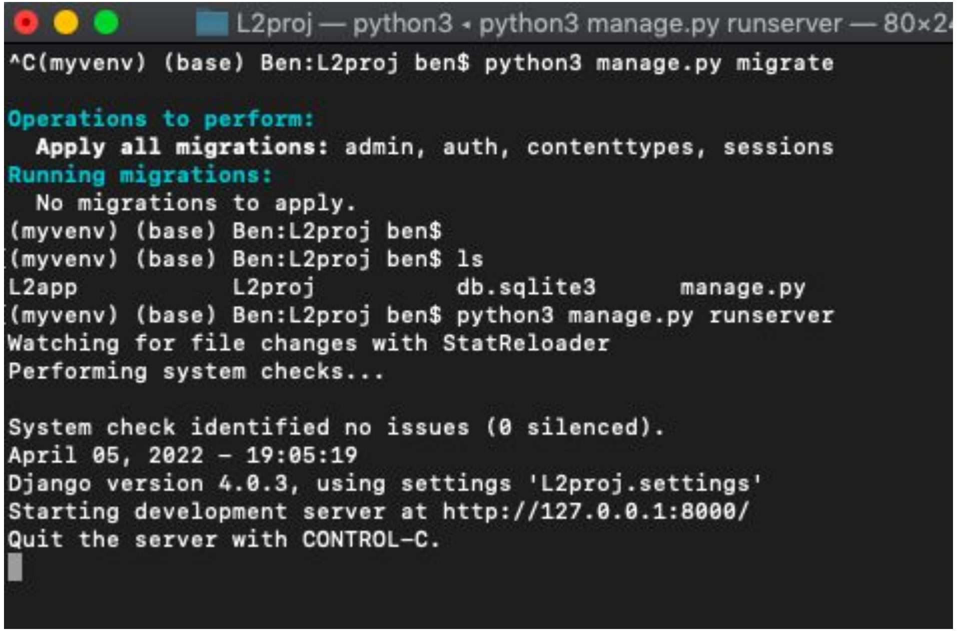
```
^C(myvenv) (base) Ben:L2proj ben$ python3 manage.py migrate

Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
(myvenv) (base) Ben:L2proj ben$
(myvenv) (base) Ben:L2proj ben$
```

Successful propagation



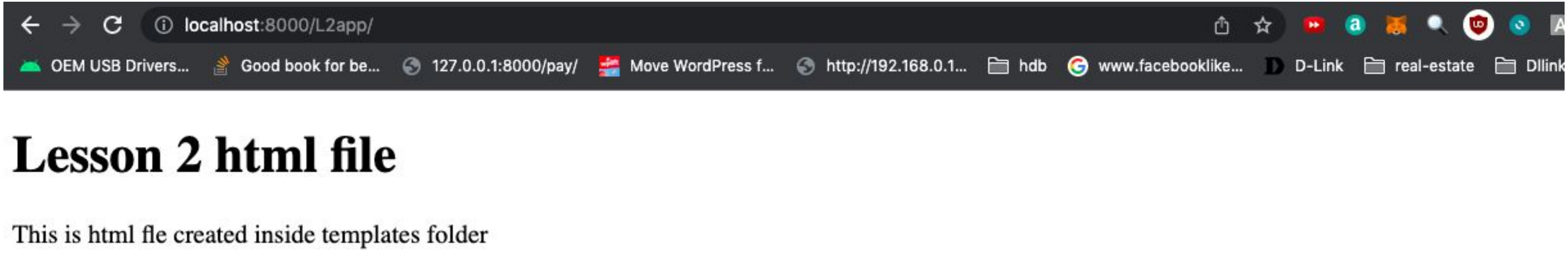
```
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

Run server



```
^C(myvenv) (base) Ben:L2proj ben$ python3 manage.py migrate

Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
(myvenv) (base) Ben:L2proj ben$
(myvenv) (base) Ben:L2proj ben$ ls
L2app           L2proj          db.sqlite3      manage.py
(myvenv) (base) Ben:L2proj ben$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 05, 2022 - 19:05:19
Django version 4.0.3, using settings 'L2proj.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Successful html output

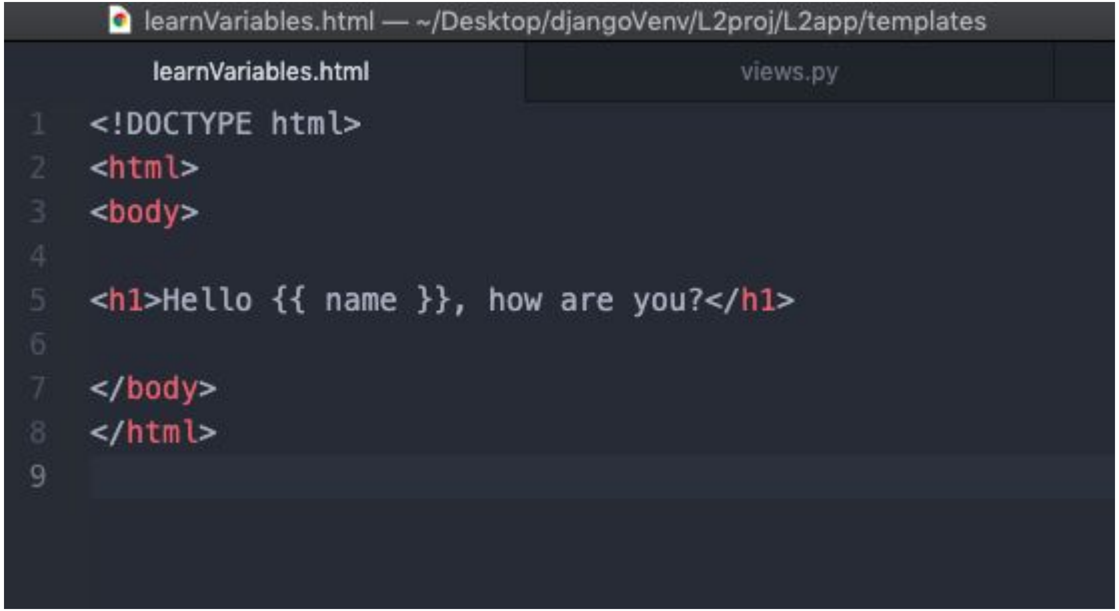


Lesson 2 html file

This is html fle created inside templates folder

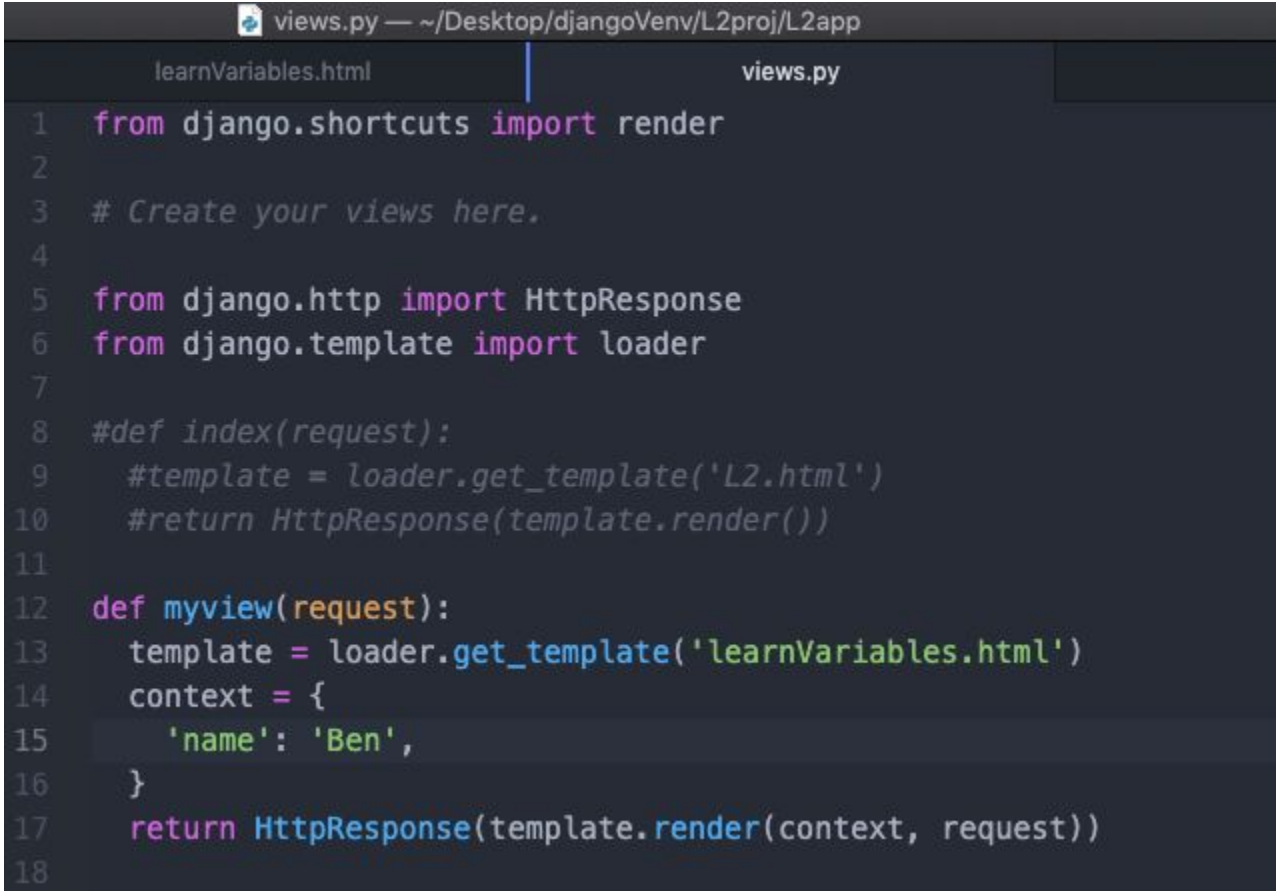
Template Variables

In Django you can insert dynamic objects such as text into the html file


learnVariables.html — ~/Desktop/djangoVenv/L2proj/L2app/templates
learnVariables.html
views.py

```
<!DOCTYPE html>
<html>
<body>

<h1>Hello {{ name }}, how are you?</h1>

</body>
</html>

```

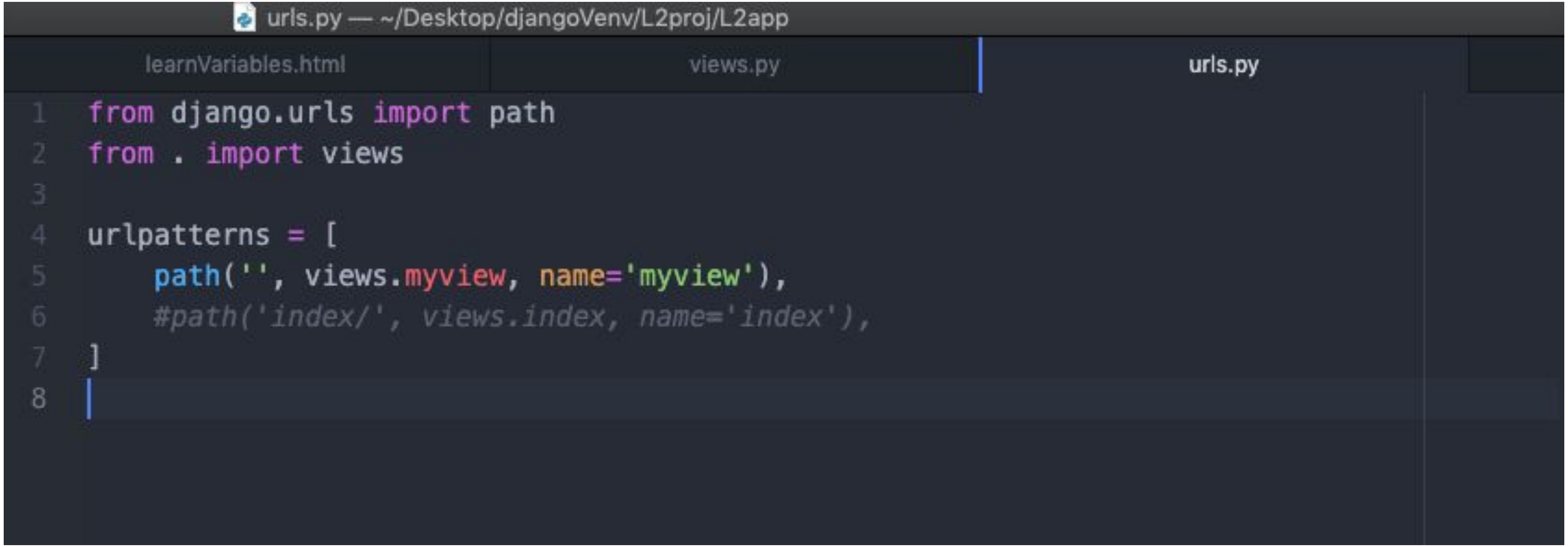
Assigning variable



```python
from django.shortcuts import render

# Create your views here.

from django.http import HttpResponse
from django.template import loader

#def index(request):
  #template = loader.get_template('L2.html')
  #return HttpResponse(template.render())

def myview(request):
  template = loader.get_template('learnVariables.html')
  context = {
    'name': 'Ben',
  }
  return HttpResponse(template.render(context, request))

```

URLpatterns and views

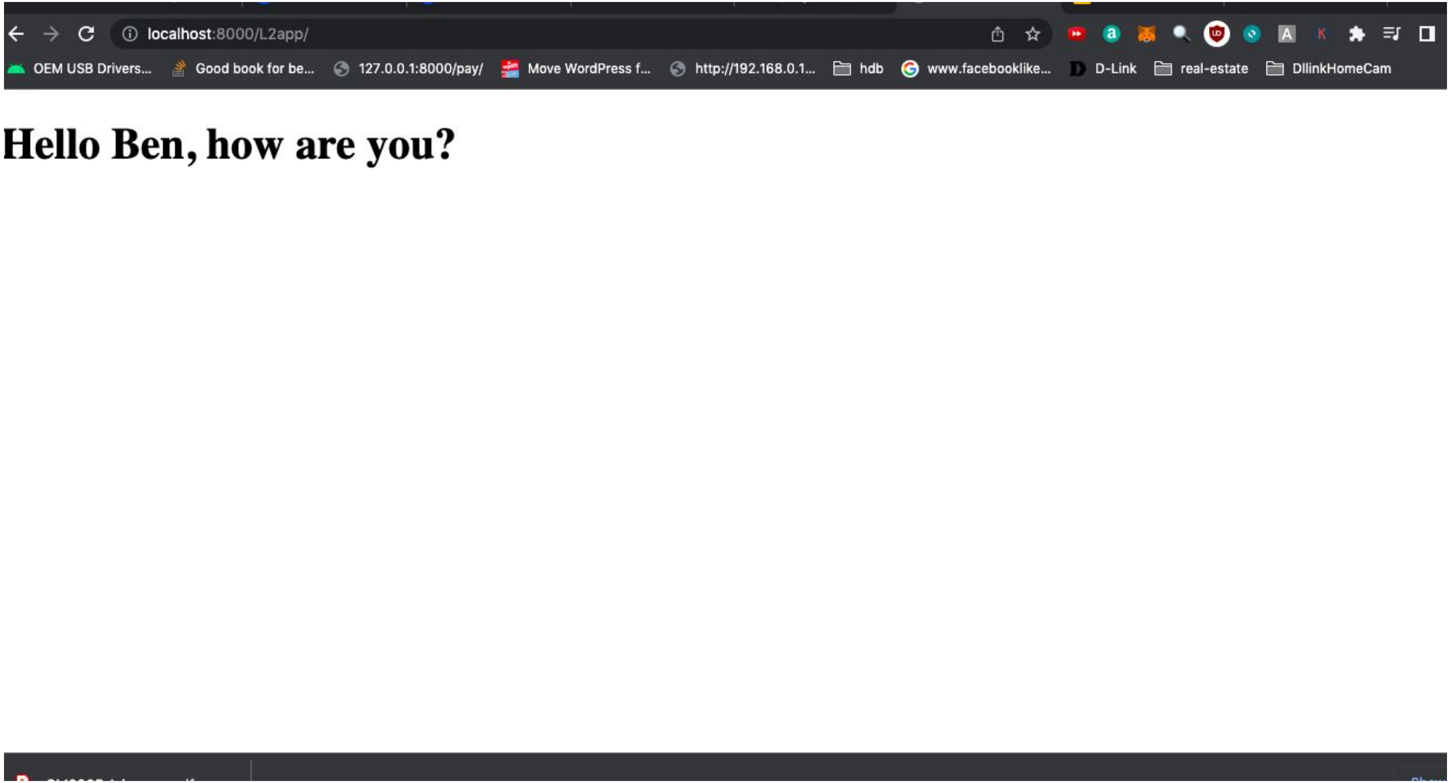
Ensure urlpatterns array has the path to myview function



```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.myview, name='myview'),
    #path('index/', views.index, name='index'),
]
```

Output

Run server using manage.py



Hello Ben, how are you?

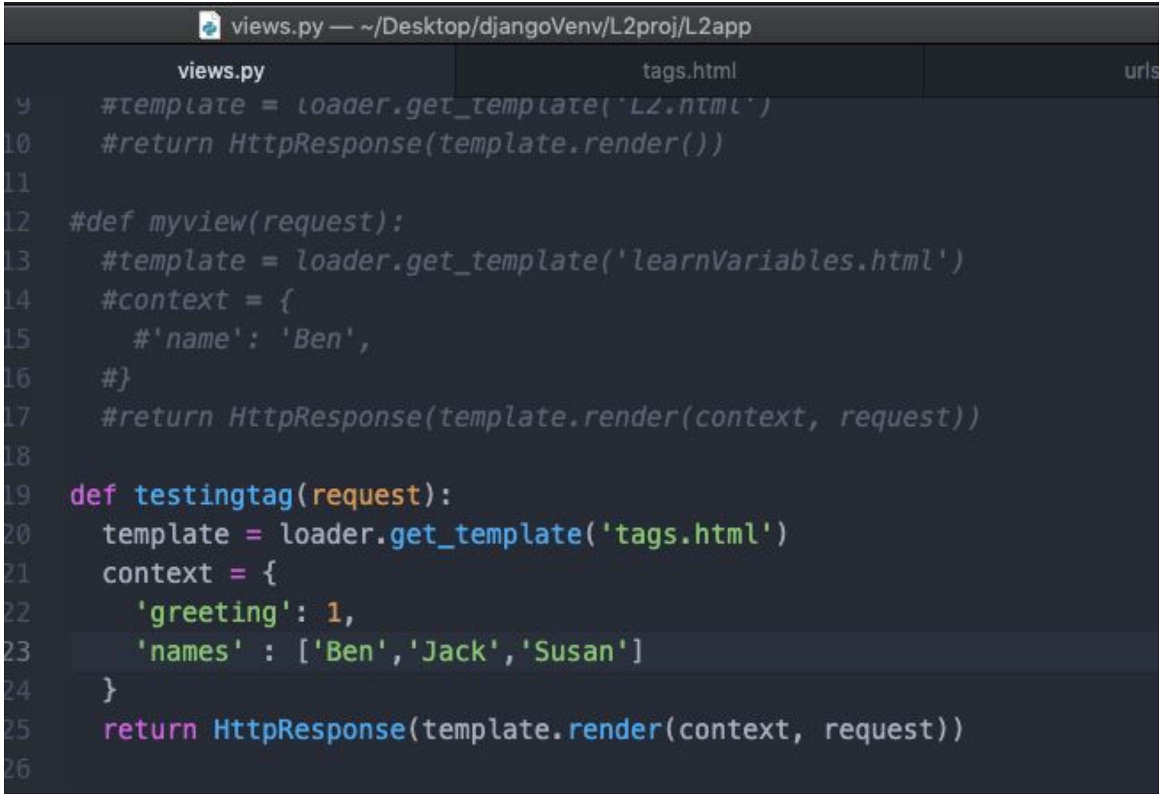
Template Tags

If statements

For Loops

You can wrap template tags with {% %} in html files

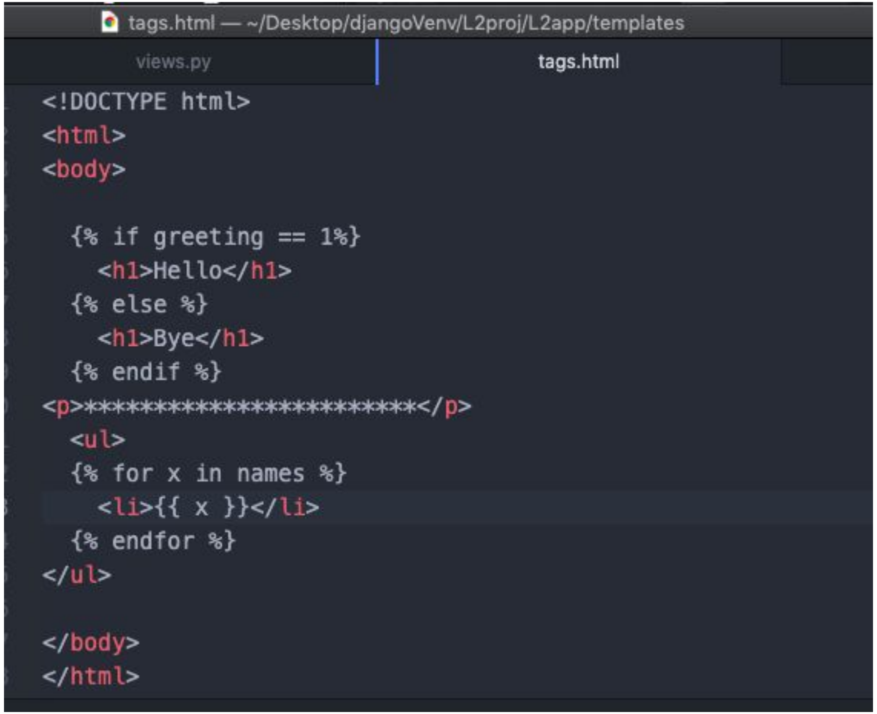
Assign variables in views



```
  #template = loader.get_template('L2.html')
  #return HttpResponse(template.render())

#def myview(request):
  #template = loader.get_template('learnVariables.html')
  #context = {
    #'name': 'Ben',
  #}
  #return HttpResponse(template.render(context, request))

def testingtag(request):
  template = loader.get_template('tags.html')
  context = {
    'greeting': 1,
    'names' : ['Ben','Jack','Susan']
  }
  return HttpResponse(template.render(context, request))

```

Template Tags in html file

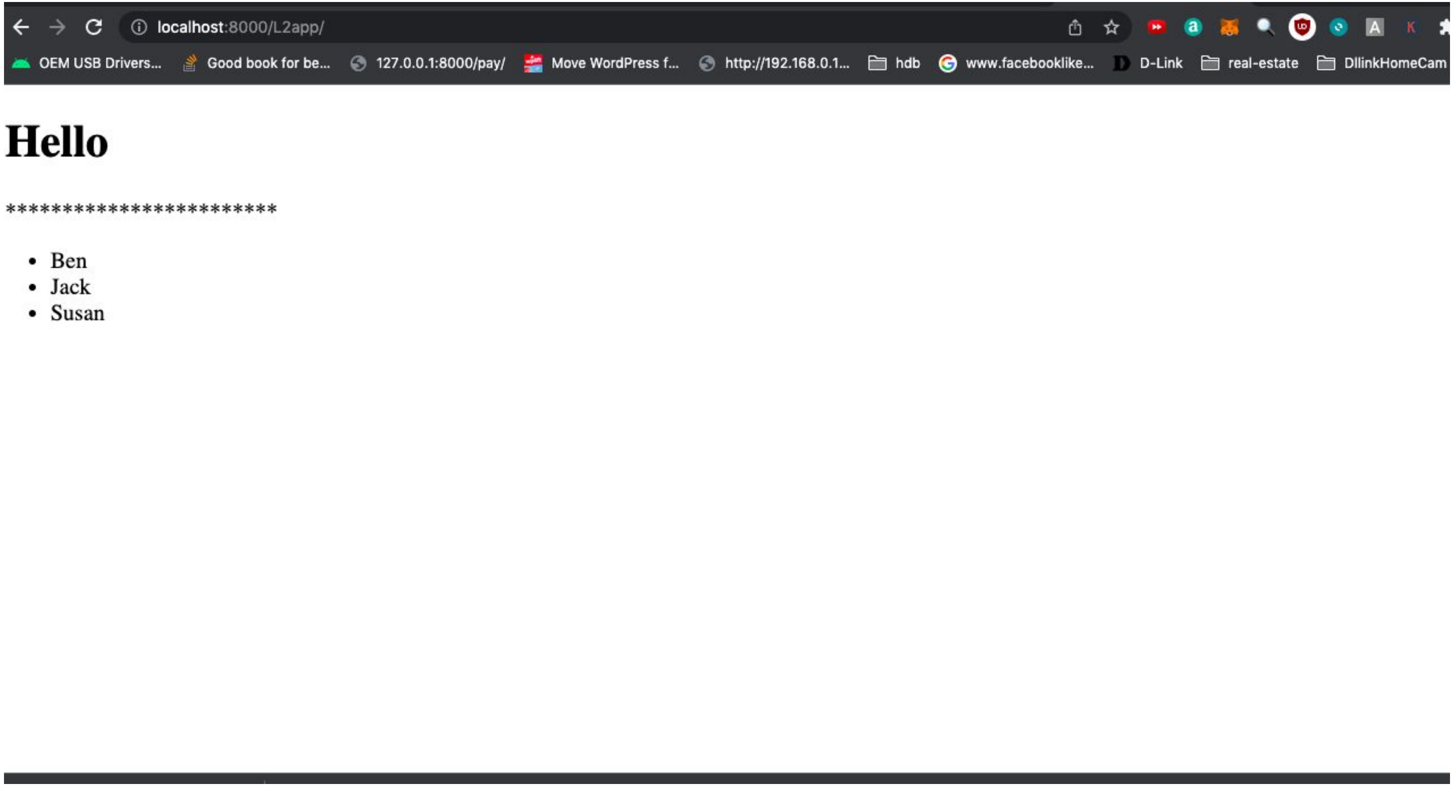
Location : templates/tags.html



tags.html — ~/Desktop/djangoVenv/L2proj/L2app/templates

```
<!DOCTYPE html>
<html>
<body>

  {% if greeting == 1%}
    <h1>Hello</h1>
  {% else %}
    <h1>Bye</h1>
  {% endif %}
<p>************************</p>
  <ul>
  {% for x in names %}
    <li>{{ x }}</li>
  {% endfor %}
</ul>

</body>
</html>
```

Run Server



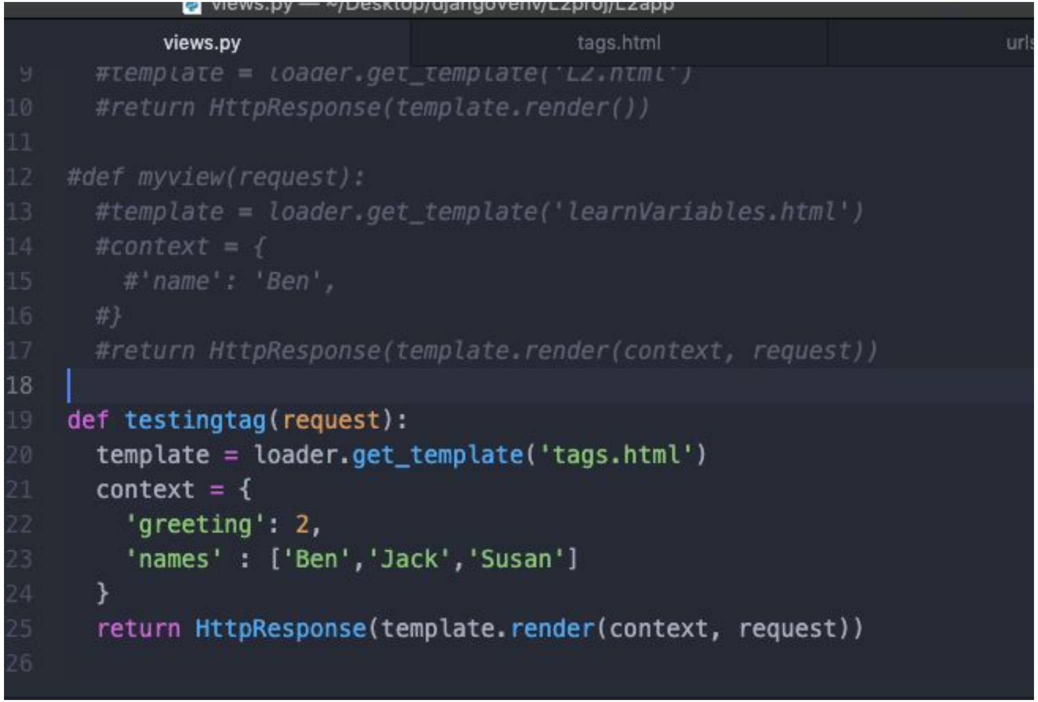
Hello

****

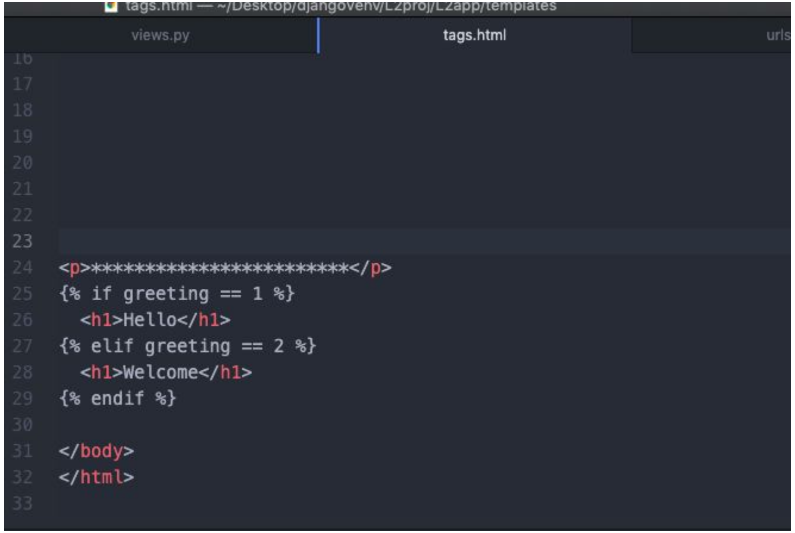
- Ben
- Jack
- Susan

Django If Else

Change variable greeting value to 2



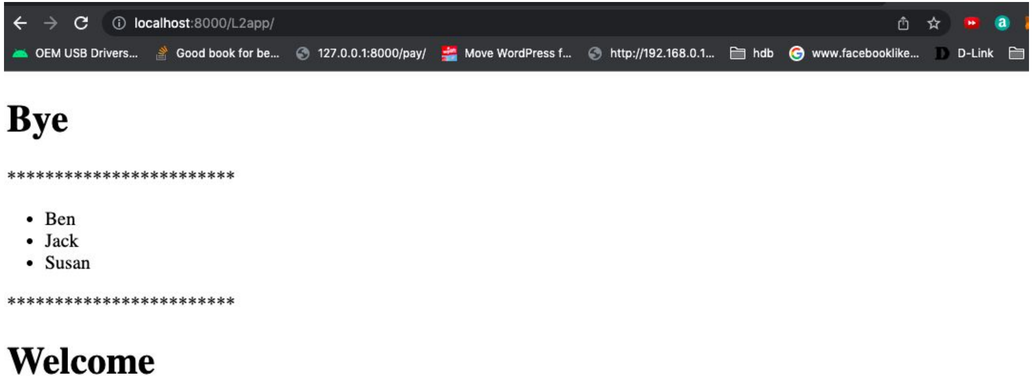
```
  #template = loader.get_template('L2.html')
  #return HttpResponse(template.render())

#def myview(request):
  #template = loader.get_template('learnVariables.html')
  #context = {
    #'name': 'Ben',
  #}
  #return HttpResponse(template.render(context, request))

def testingtag(request):
  template = loader.get_template('tags.html')
  context = {
    'greeting': 2,
    'names' : ['Ben','Jack','Susan']
  }
  return HttpResponse(template.render(context, request))
```



```
<p>************************</p>
{% if greeting == 1 %}
  <h1>Hello</h1>
{% elif greeting == 2 %}
  <h1>Welcome</h1>
{% endif %}

</body>
</html>
```

Django IF Else Output

Refresh browser



```
Bye

************************

  • Ben
  • Jack
  • Susan

************************

Welcome
```

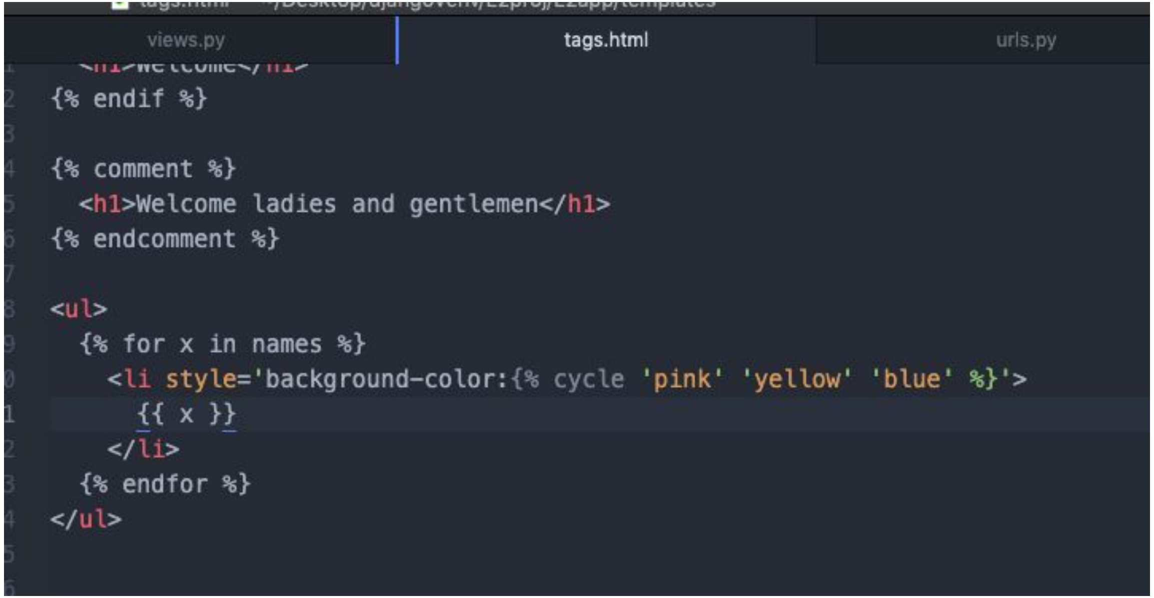
Django Comments

```
{% comment %}
<h1>Anything inside here will not be shown on browser</h1>
{% endcomment %}
```

Try it!!!

Cycles tags

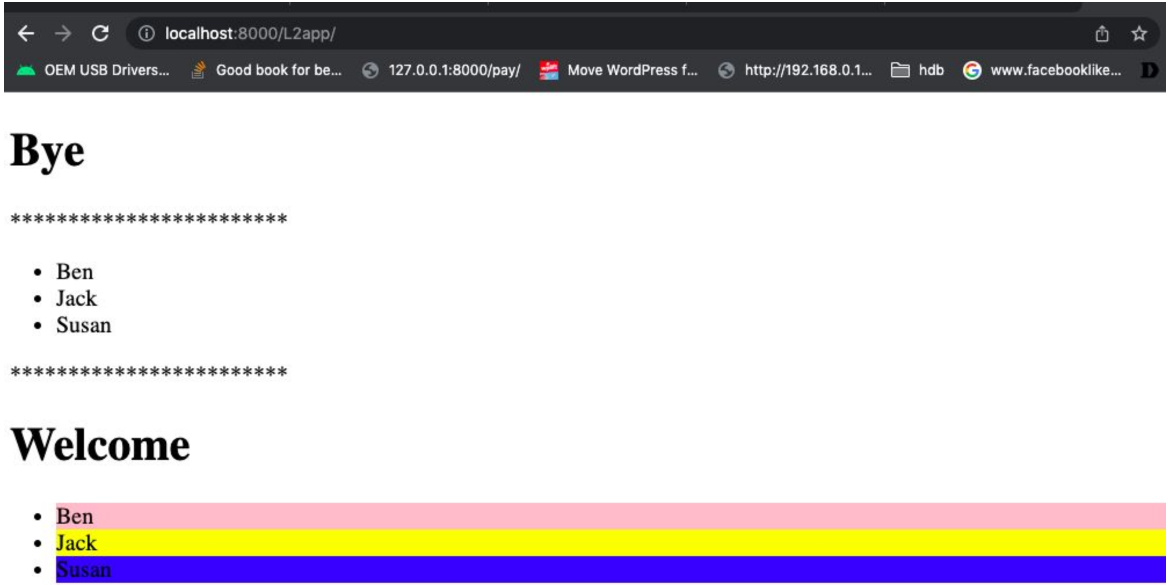
Cycle tag takes arguments, the first iteration uses the first argument, the second iteration uses the second argument etc.



```
  <h1>Welcome</h1>
{% endif %}

{% comment %}
  <h1>Welcome ladies and gentlemen</h1>
{% endcomment %}

<ul>
  {% for x in names %}
    <li style='background-color:{% cycle 'pink' 'yellow' 'blue' %}'>
      {{ x }}
    </li>
  {% endfor %}
</ul>
```

Output

Refresh browser



Bye

****

- Ben
- Jack
- Susan

****

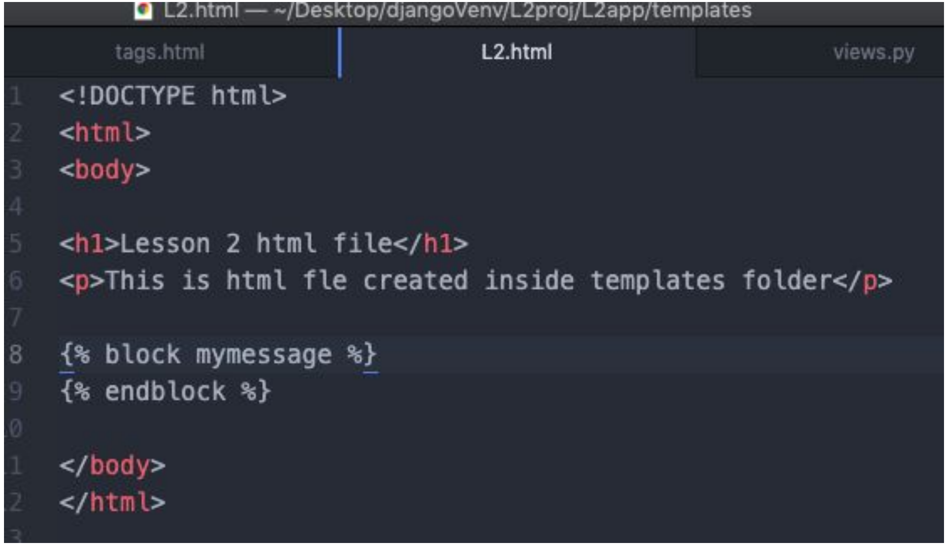
Welcome

- Ben
- Jack
- Susan

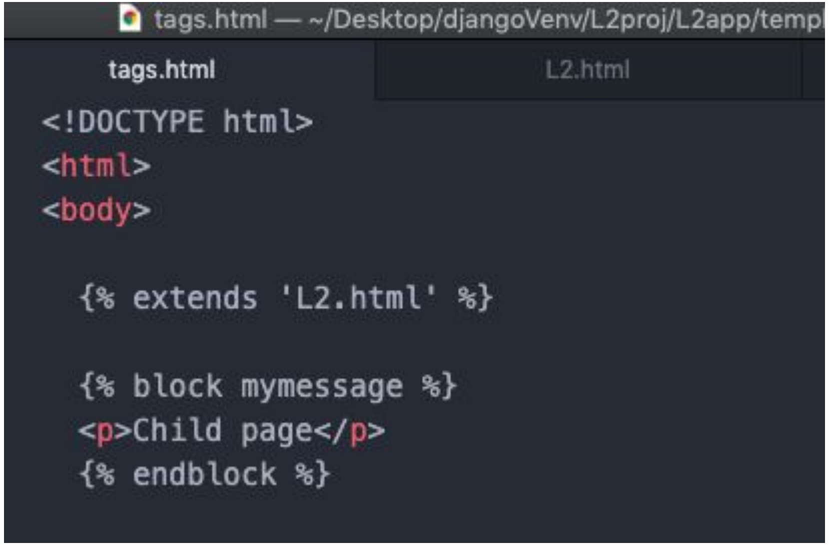
Extends Tag

The extends tag can add a parent template to a child template.

Parent page acts as a skeleton for all other pages (child pages).



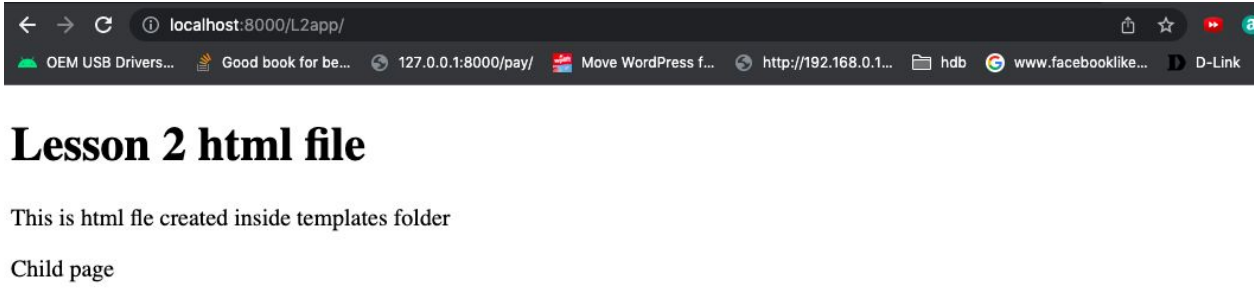
```
<!DOCTYPE html>
<html>
<body>

<h1>Lesson 2 html file</h1>
<p>This is html fle created inside templates folder</p>

{% block mymessage %}
{% endblock %}

</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>

  {% extends 'L2.html' %}

  {% block mymessage %}
  <p>Child page</p>
  {% endblock %}
```

Extends Tag Output

Refresh browser



Lesson 2 html file

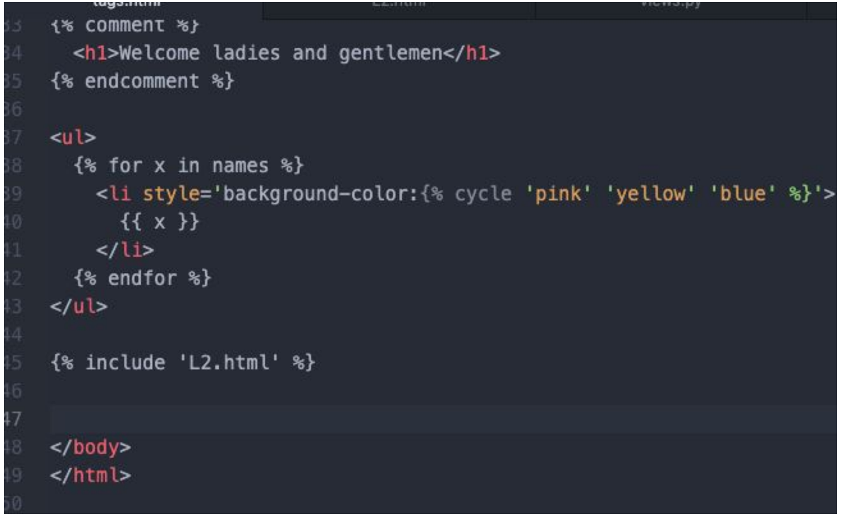
This is html fle created inside templates folder

Child page

Include tag

The include tag allows you include a template inside the current template.

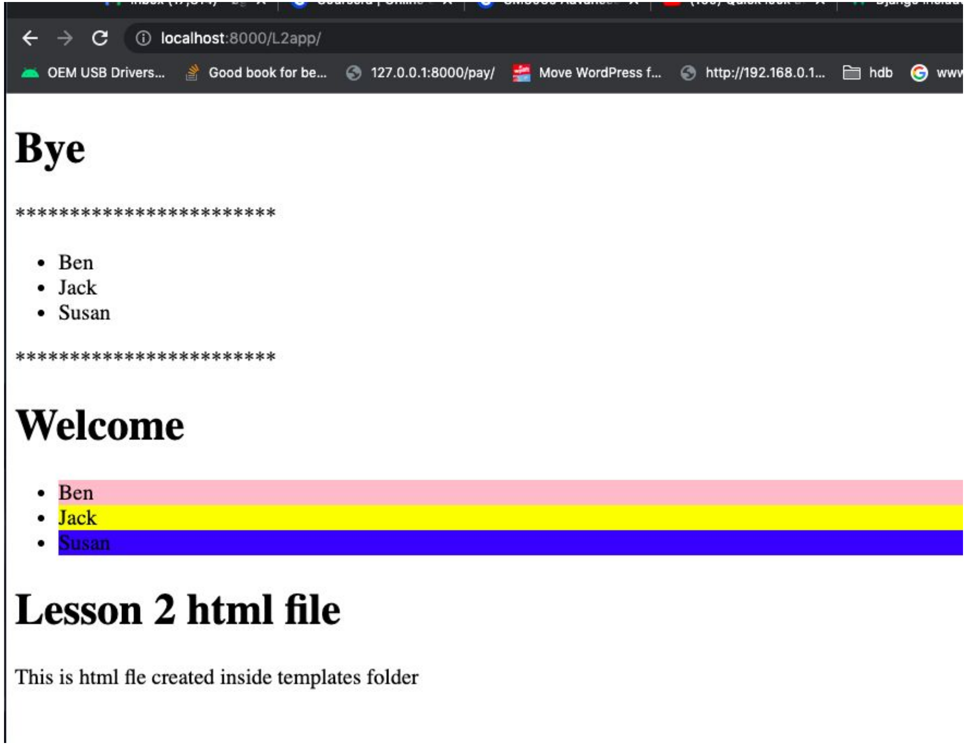
Good for block of content that are the same for many pages.



```
{% comment %}
  <h1>Welcome ladies and gentlemen</h1>
{% endcomment %}

<ul>
  {% for x in names %}
    <li style='background-color:{% cycle 'pink' 'yellow' 'blue' %}'>
      {{ x }}
    </li>
  {% endfor %}
</ul>

{% include 'L2.html' %}


</body>
</html>
```

Include tag output

Refresh browser



Bye

****

- Ben
- Jack
- Susan

****

Welcome

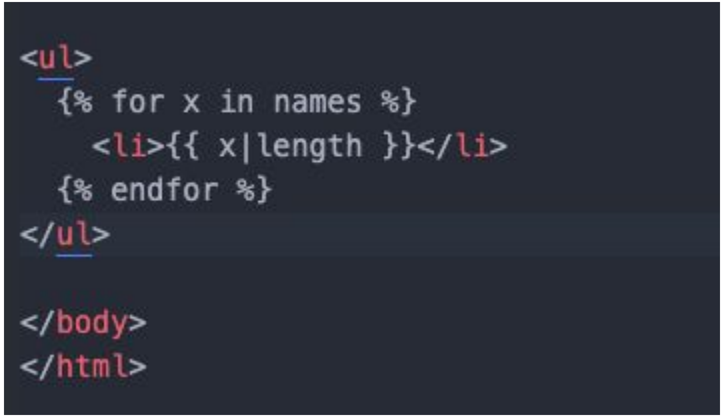
- Ben
- Jack
- Susan

Lesson 2 html file

This is html fle created inside templates folder

Filter tag

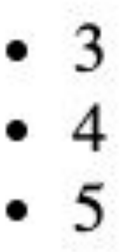
The filter tag allows you to run a value through a filter before returning it.



```
<ul>
  {% for x in names %}
    <li>{{ x|length }}</li>
  {% endfor %}
</ul>

</body>
</html>
```

Filter tag output

Refresh browser



- 3
- 4
- 5

Additional django references

Template tag -

https://www.djangotemplatetagsandfilters.com/tags/

Filter reference -

https://www.djangotemplatetagsandfilters.com/filters/

End of Lesson 2