

# 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),
]
```

d in Current Buffer

Finding with Options: Case Insensitive

# 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

A screenshot of a code editor window. The title bar at the top reads "urls.py — L2app — ~/Desktop/djangoVenv/L2proj/L2app". Below the title bar, the file name "urls.py" is displayed. The code is as follows:

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name='index'),
6 ]
7
```

# Create html file inside templates

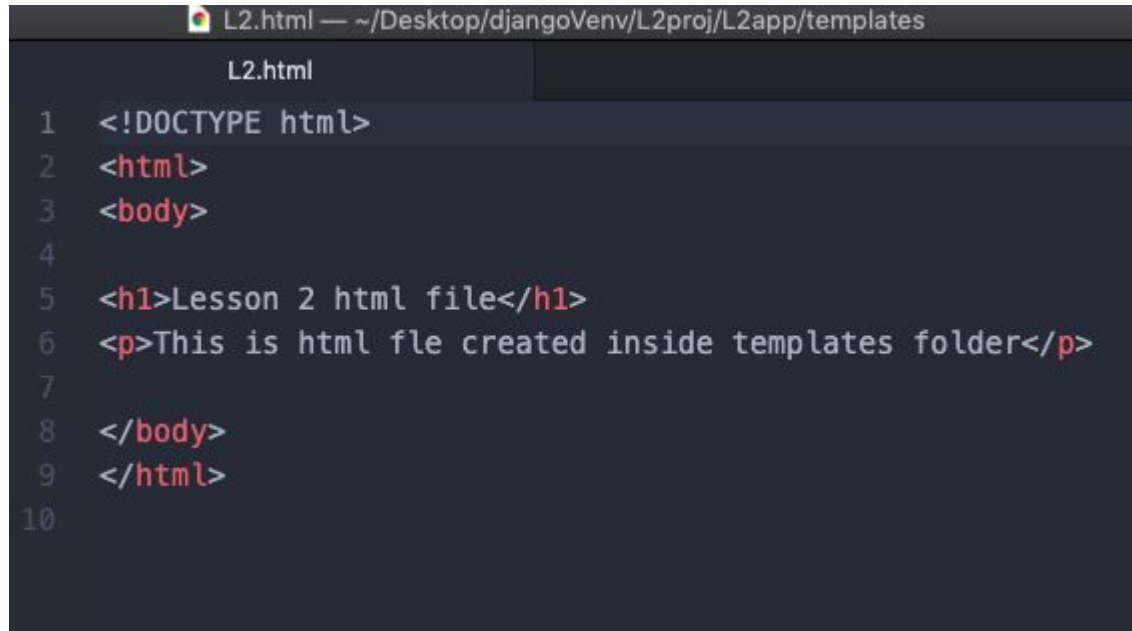
Under Django framework all html files should be stored in the  
templates folder

Create templates folder inside L2app

Add L2.html file inside templates folder



# Content inside L2.html

A screenshot of a code editor window. The title bar at the top reads "L2.html — ~/Desktop/djangoVenv/L2proj/L2app/templates". Below the title bar, there is a tab labeled "L2.html". The main area of the editor shows the following HTML code with line numbers 1 through 10 on the left:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Lesson 2 html file</h1>
6 <p>This is html file created inside templates folder</p>
7
8 </body>
9 </html>
10
```

# views.py

Set L2.html to a variable

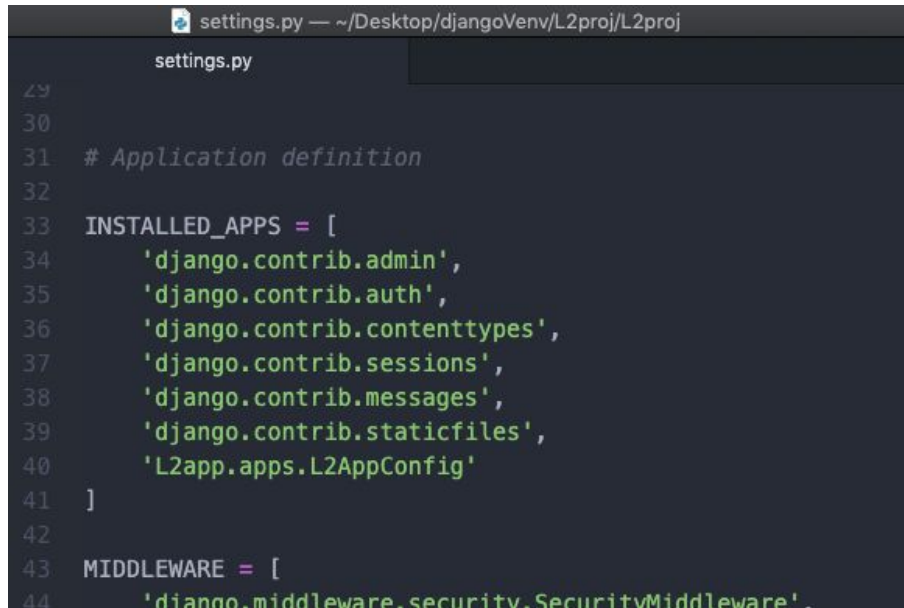
Return the http response to the browser

Update to L2app/views.py

```
views.py — ~/Desktop/djangoVenv/L2proj/L2app
views.py
1  from django.shortcuts import render
2
3  # Create your views here.
4
5  from django.http import HttpResponse
6  from django.template import loader
7
8  def index(request):
9      template = loader.get_template('L2.html')
10     return HttpResponse(template.render())
11
```

# Configure L2proj/settings.py

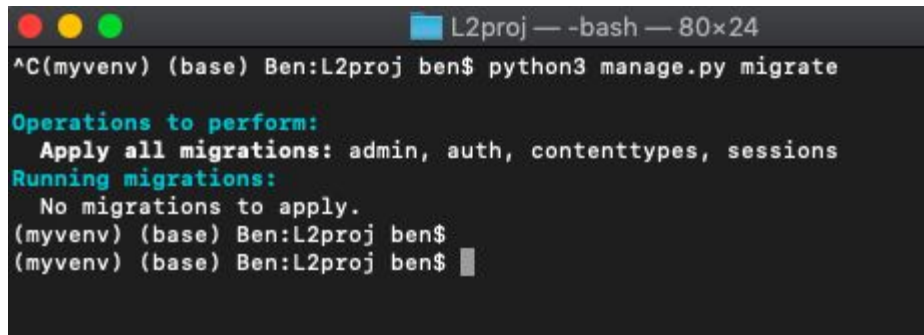
Configure L2app.app.L2AppConfig object into the array called  
INSTALLED\_APPS under file L2proj/settings.py



```
settings.py
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'L2app.apps.L2AppConfig'
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
```

# Propagating Changes

>>>python3 manage.py migrate

A screenshot of a terminal window with a dark background. The window title bar shows three colored circles (red, yellow, green) on the left, followed by a blue folder icon and the text "L2proj — -bash — 80x24". The terminal content shows a command prompt where the user has entered "python3 manage.py migrate". The output displays "Operations to perform:" followed by "Apply all migrations: admin, auth, contenttypes, sessions". Then it shows "Running migrations:" followed by "No migrations to apply." The prompt then changes from "(myvenv) (base) Ben:L2proj ben\$" to "(myvenv) (base) Ben:L2proj ben\$".

```
^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

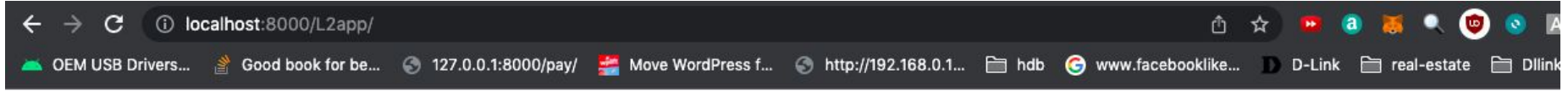
Applying sessions.0001\_initial... OK

# Run server

```
L2proj — python3 - python3 manage.py runserver — 80x24
^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 file 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
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Hello {{ name }}, how are you?</h1>
6
7 </body>
8 </html>
9
```



# Assigning variable

```
views.py — ~/Desktop/djangoVenv/L2proj/L2app
learnVariables.html | views.py
1  from django.shortcuts import render
2
3  # Create your views here.
4
5  from django.http import HttpResponse
6  from django.template import loader
7
8  #def index(request):
9      #template = loader.get_template('L2.html')
10     #return HttpResponse(template.render())
11
12  def myview(request):
13     template = loader.get_template('learnVariables.html')
14     context = {
15         'name': 'Ben',
16     }
17     return HttpResponse(template.render(context, request))
18
```

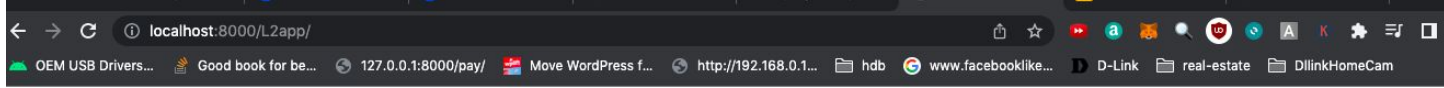
# URL patterns and views

Ensure urlpatterns array has the path to myview function

```
urls.py — ~/Desktop/djangoVenv/L2proj/L2app
learnVariables.html  views.py  urls.py
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.myview, name='myview'),
6      #path('index/', views.index, name='index'),
7  ]
8  |
```

# 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

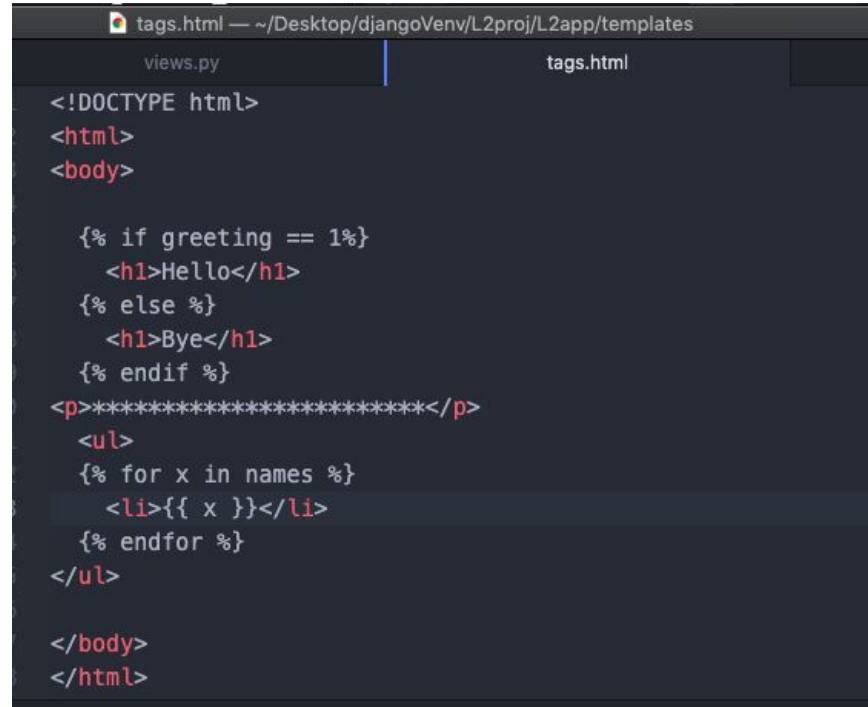
```
views.py — ~/Desktop/djangoVenv/L2proj/L2app

views.py    tags.html    urls

9  #template = loader.get_template('L2.html')
10 #return HttpResponse(template.render())
11
12 #def myview(request):
13     #template = loader.get_template('learnVariables.html')
14     #context = {
15         #'name': 'Ben',
16     #}
17     #return HttpResponse(template.render(context, request))
18
19 def testingtag(request):
20     template = loader.get_template('tags.html')
21     context = {
22         'greeting': 1,
23         'names' : ['Ben', 'Jack', 'Susan']
24     }
25     return HttpResponse(template.render(context, request))
26
```

# Template Tags in html file

Location : templates/tags.html

A screenshot of a code editor window. The title bar shows 'tags.html' and the file path '~/.Desktop/djangoVenv/L2proj/L2app/templates'. The editor has two tabs: 'views.py' and 'tags.html', with 'tags.html' being the active tab. The code is written in a dark theme with syntax highlighting. It starts with a DOCTYPE declaration, followed by HTML tags for the body. Inside the body, there is a conditional block for 'greeting' and a loop for 'names'.

```
tags.html — ~/.Desktop/djangoVenv/L2proj/L2app/templates
views.py | tags.html
<!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>
```

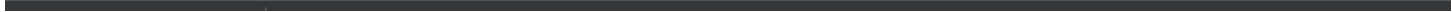
# Run Server



## Hello

\*\*\*\*\*

- Ben
- Jack
- Susan



# Django If Else

Change variable greeting value to 2

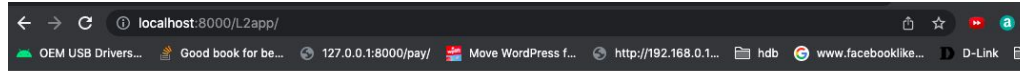
```
views.py — ~/Desktop/djangovenw/L2proj/L2app
views.py      tags.html      urls
9  #template = loader.get_template('L2.html')
10 #return HttpResponse(template.render())
11
12 #def myview(request):
13 #template = loader.get_template('learnVariables.html')
14 #context = {
15 #    #'name': 'Ben',
16 #}
17 #return HttpResponse(template.render(context, request))
18
19 def testingtag(request):
20     template = loader.get_template('tags.html')
21     context = {
22         'greeting': 2,
23         'names' : ['Ben', 'Jack', 'Susan']
24     }
25     return HttpResponse(template.render(context, request))
26
```

```
tags.html — ~/Desktop/djangovenw/L2proj/L2app/templates
views.py      tags.html      urls
16
17
18
19
20
21
22
23
24 <p>*****</p>
25 {% if greeting == 1 %}
26     <h1>Hello</h1>
27 {% elif greeting == 2 %}
28     <h1>Welcome</h1>
29 {% endif %}
30
31 </body>
32 </html>
33
```



# Django IF Else Output

Refresh browser



**Bye**

\*\*\*\*\*

- Ben
- Jack
- Susan

\*\*\*\*\*

**Welcome**

# Django Comments

{% comment %}

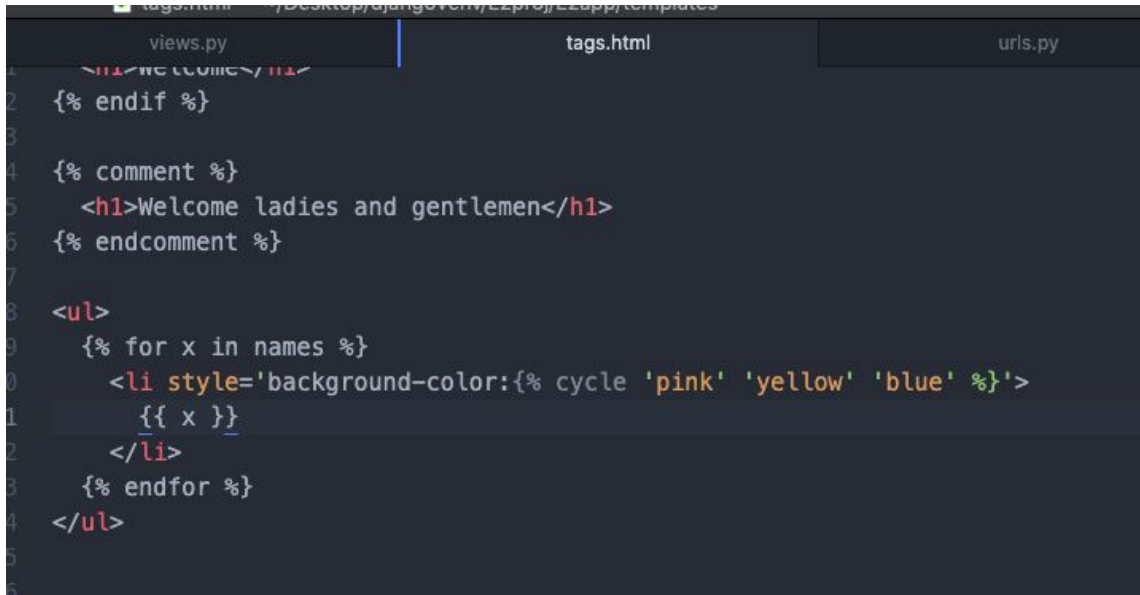
<h1>Anything inside here will not be shown on browser</h1>

{% endcomment %}

**Try it!!!**

# Cycles tags

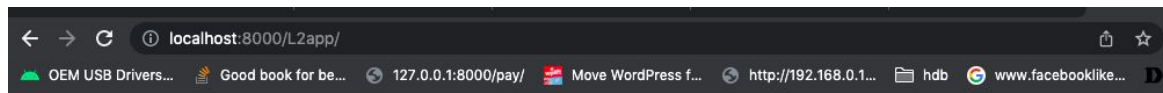
Cycle tag takes arguments, the first iteration uses the first argument, the second iteration uses the second argument etc.



```
views.py | tags.html | urls.py
1 <h1>Welcome</h1>
2 {% endif %}
3
4 {% comment %}
5 <h1>Welcome ladies and gentlemen</h1>
6 {% endcomment %}
7
8 <ul>
9   {% for x in names %}
10     <li style='background-color:{% cycle 'pink' 'yellow' 'blue' %}'>
11       {{ x }}
12     </li>
13   {% endfor %}
14 </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).

```
L2.html — ~/Desktop/djangoVenv/L2proj/L2app/templates
tags.html  L2.html  views.py
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Lesson 2 html file</h1>
6 <p>This is html file created inside templates folder</p>
7
8 {% block mymessage %}
9 {% endblock %}
10
11 </body>
12 </html>
13
```

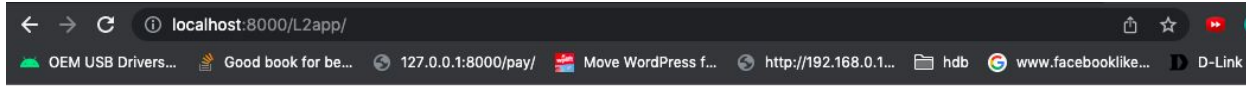
```
tags.html — ~/Desktop/djangoVenv/L2proj/L2app/temp
tags.html  L2.html
<!DOCTYPE html>
<html>
<body>

{% extends 'L2.html' %}

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

# Extends Tag Output

Refresh browser



## Lesson 2 html file

This is html file created inside templates folder

Child page

# Include tag

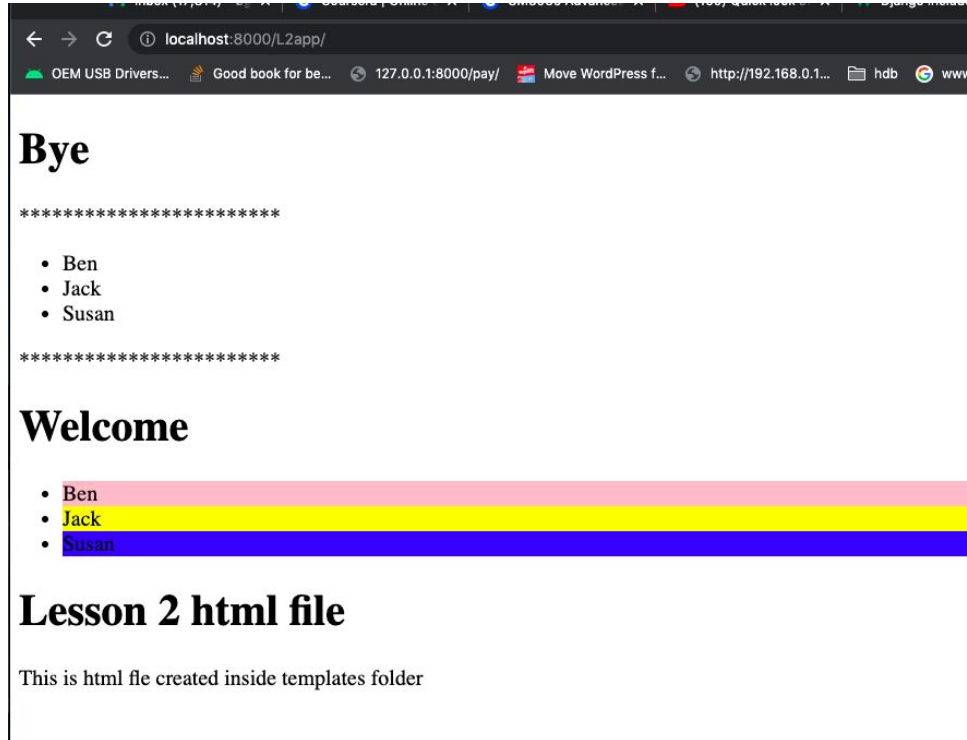
The include tag allows you include a template inside the current template.

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

```
33 {% comment %}  
34     <h1>Welcome ladies and gentlemen</h1>  
35 {% endcomment %}  
36  
37 <ul>  
38     {% for x in names %}  
39         <li style='background-color:{% cycle 'pink' 'yellow' 'blue' %}'>  
40             {{ x }}  
41         </li>  
42     {% endfor %}  
43 </ul>  
44  
45 {% include 'L2.html' %}  
46  
47  
48 </body>  
49 </html>  
50
```

# Include tag output

Refresh browser





# 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