**Create a folder with two HTML files**

We start with creating a folder “formedium”, and in this folder, we create two HTML files:

1. index.html

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Page 1</**title**>  
</**head**>  
<**body**>  
 <**H1**>This is our first page</**H1**>  
 <**p**>You may add a picture of a cat here.</**p**>  
</**body**>  
</**html**>

2. page2.html

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Page 2</**title**>  
</**head**>  
<**body**>  
 <**H1**>This is our second page</**H1**>  
 <**p**>You may add a picture of a house here.</**p**>  
</**body**>  
</**html**>

If you have some imagination, you don’t even have to run these files in a browser.

**Create a Django project**

Now, we create a new Django project.

This is done with several steps:

1. Create a virtual environment, install Django
2. Create a project
3. Create an app

We start with the virtual environment. This step is optional. This is done in the Terminal:

$ python -m venv venv  
$ source ./venv/bin/activate

Install Django:

$ pip install django

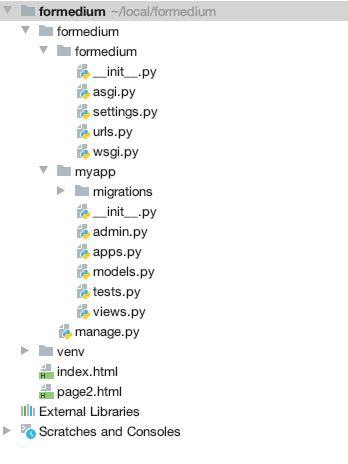
You should get a message like that:

Successfully installed asgiref-3.2.10 django-3.1.2 pytz-2020.1 sqlparse-0.4.1

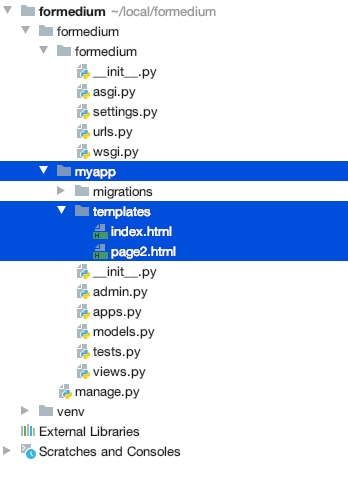
If so, create a project and an app:

$ django-admin startproject formedium  
$ cd formedium  
$ django-admin startapp myapp

So, our folder should look as follows:sf



We are almost done. Let’s now create a “template” folder inside “myapp”, and move our HTML files into it. We should now have the following:



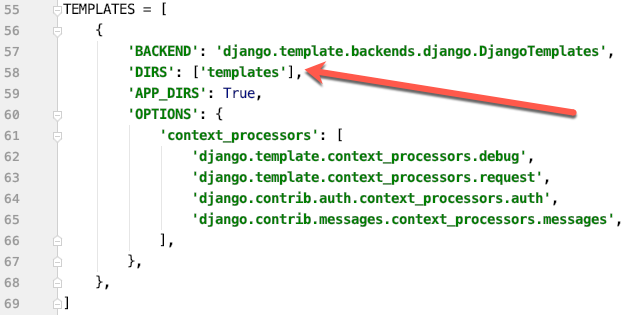
But how should our “formedium” project know where to look for the templates? We help it by modifying the **setting.py** file:

Add our app:



Add this line

Set the name of the template folder:



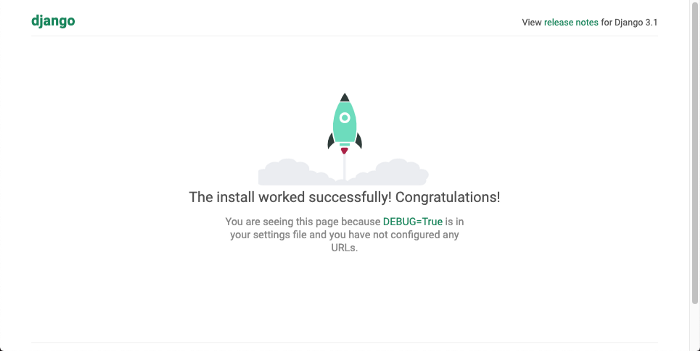
Add this item inside the DIRS list

Good. We are ready with setting up our projects!

Try running it with

$ python manage.py runserver

and running “localhost:8000” in your browser.



localhost:8000

**Point empty URL to index.html**

What is an URL?

Say, you have a website called medium.com.

If the user types “medium.com” in the browser, then this is a root URL. We say that the URL is empty in this case.

If the user types “medium.com/help”, then the URL is “/help”.

If the user types “medium.com/login?user=jimcarrey&password=monkey”, then the URL is “/login”, and all the rest we can get from the GET request, and that will be covered below.

So. We want to show our index.html whenever the URL is empty. How do we do that? In the “formedium” folder, there is a file called “urls.py”, and this file is called first when there is a request from the user.

Inside this file, we change two things:

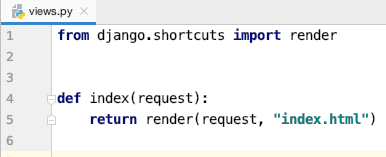
1. Import our app views.
2. Say, that if the URL is empty, edit the **views.index** function.



When the URL is empty, run the views.index function

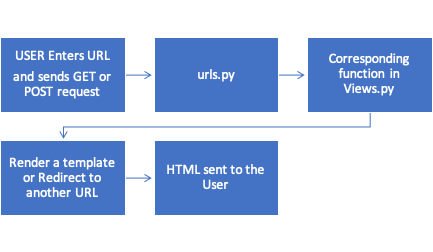
And what about this “**views.index**” function? Sure, we need to provide it.

Edit the “myapp/views.py” file as follows:



Now, try to navigate “localhost:8000”, and you should see the content of the “index.html” file.

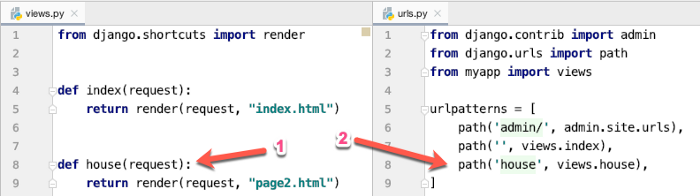
In other words, this is how Django handles requests from the user:



**Point /house URL to page2.html**

This should be an exercise for you.

All you need is to add a new URL line and a new function.



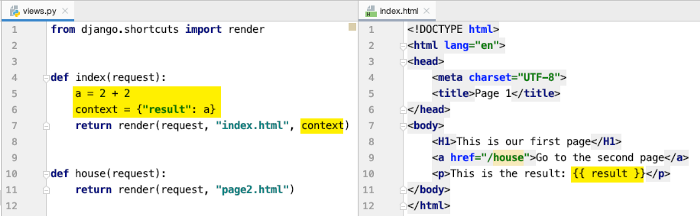
This way you may add as many URLs and HTML views as you need. To make your application more user-friendly, try to add the address of the second page into index.html:

<a href="/house">See the second page</a>

**Pass some parameters to the HTML file (Context)**

Say, your server is so clever that it can calculate sophisticated mathematical expressions and now we need to put these results into our HTML file.

For this, there is a nice thing called **context:**



Edit your files as above

A Context is a normal Python dictionary that contains all the objects you need to pass to your HTML. In our case, we have passed the value of “a” using a “result” key.

