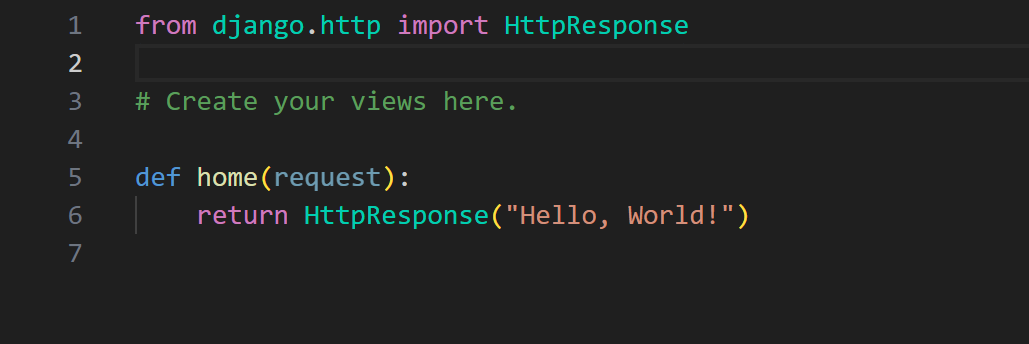
Django for web development

According to the django documentations, a view is a type of web page in your django application that generally serves a specific function and has a specific template.

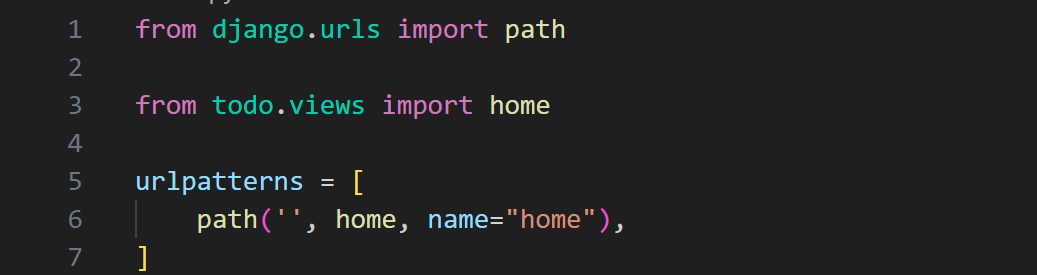
In django, web pages are delivered by a function.

Example:

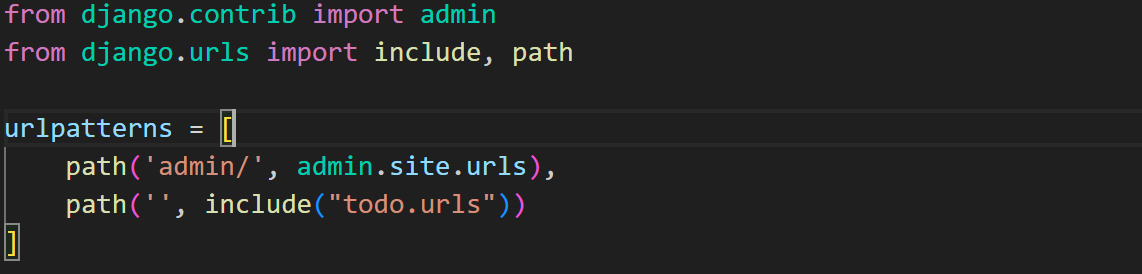
1. “todo/views.py”



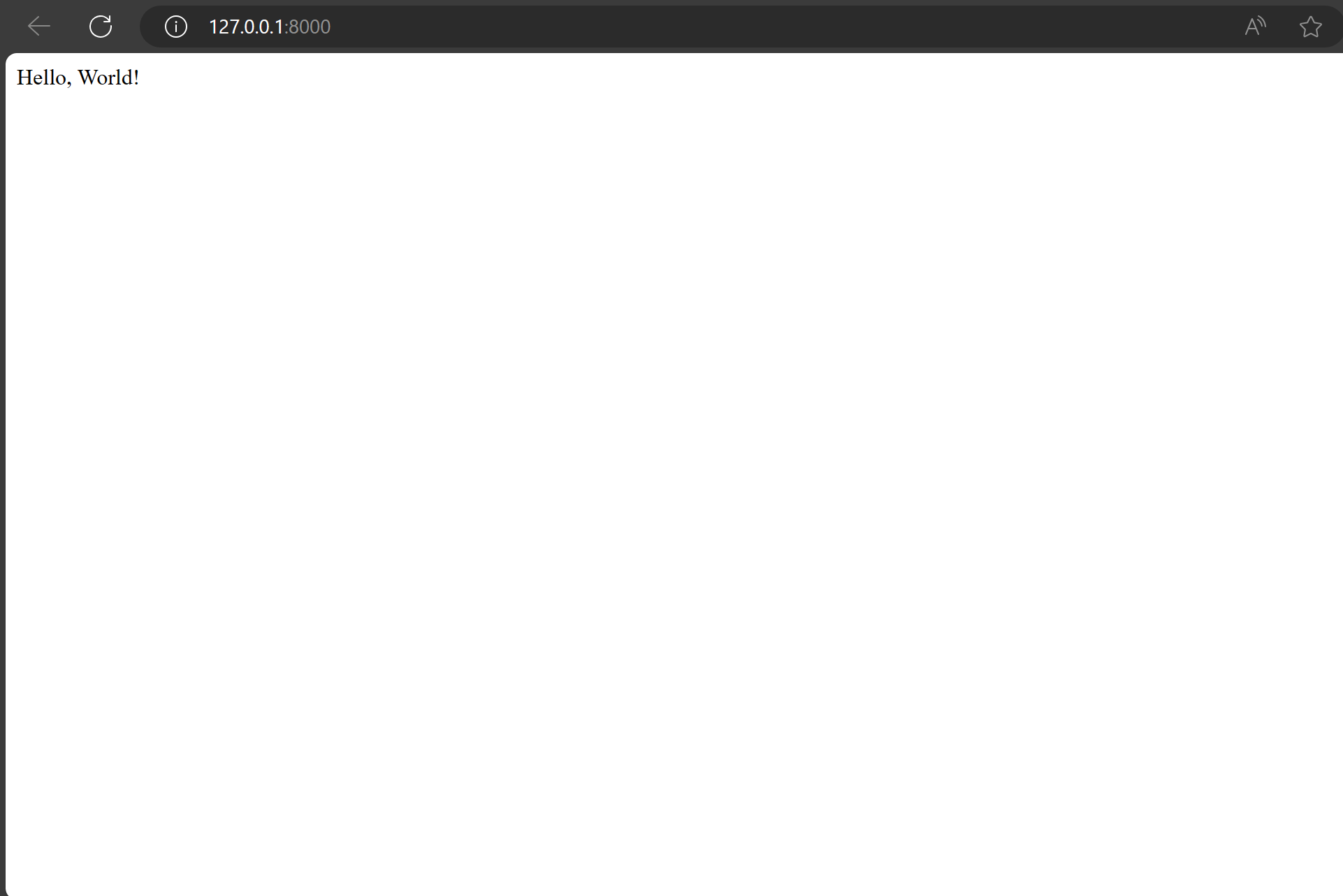
1. In the todo create a python file called urls.py
2. “todo/urls.py”



1. “saturdayproject.urls.py”

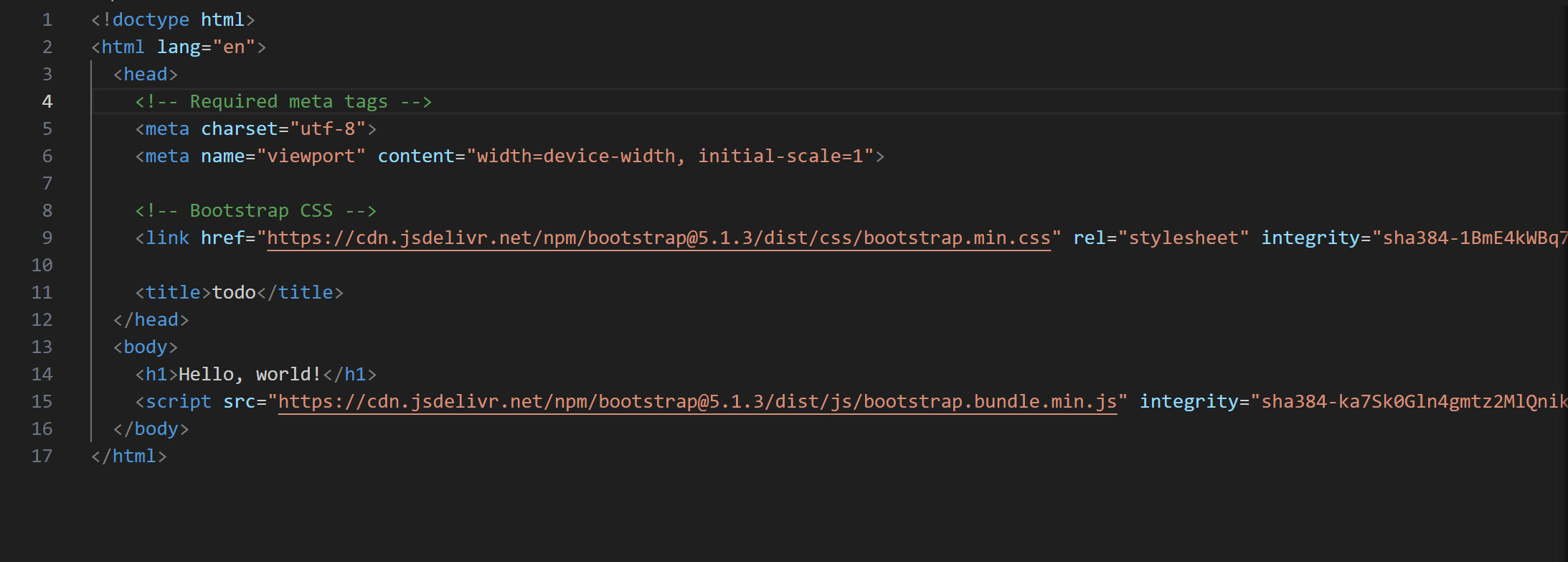


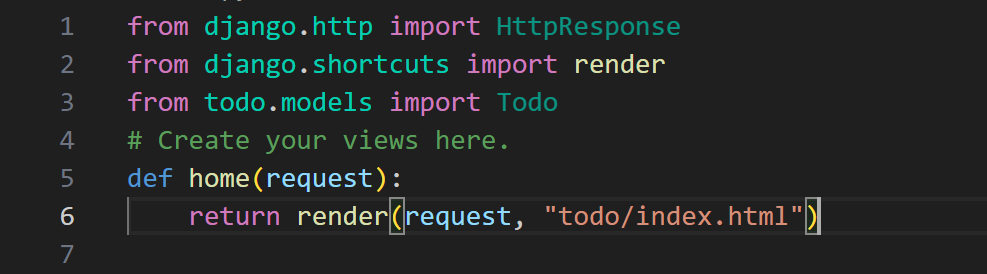
1. “Start the development server”
2. Go to http://127.0.0.1:8000



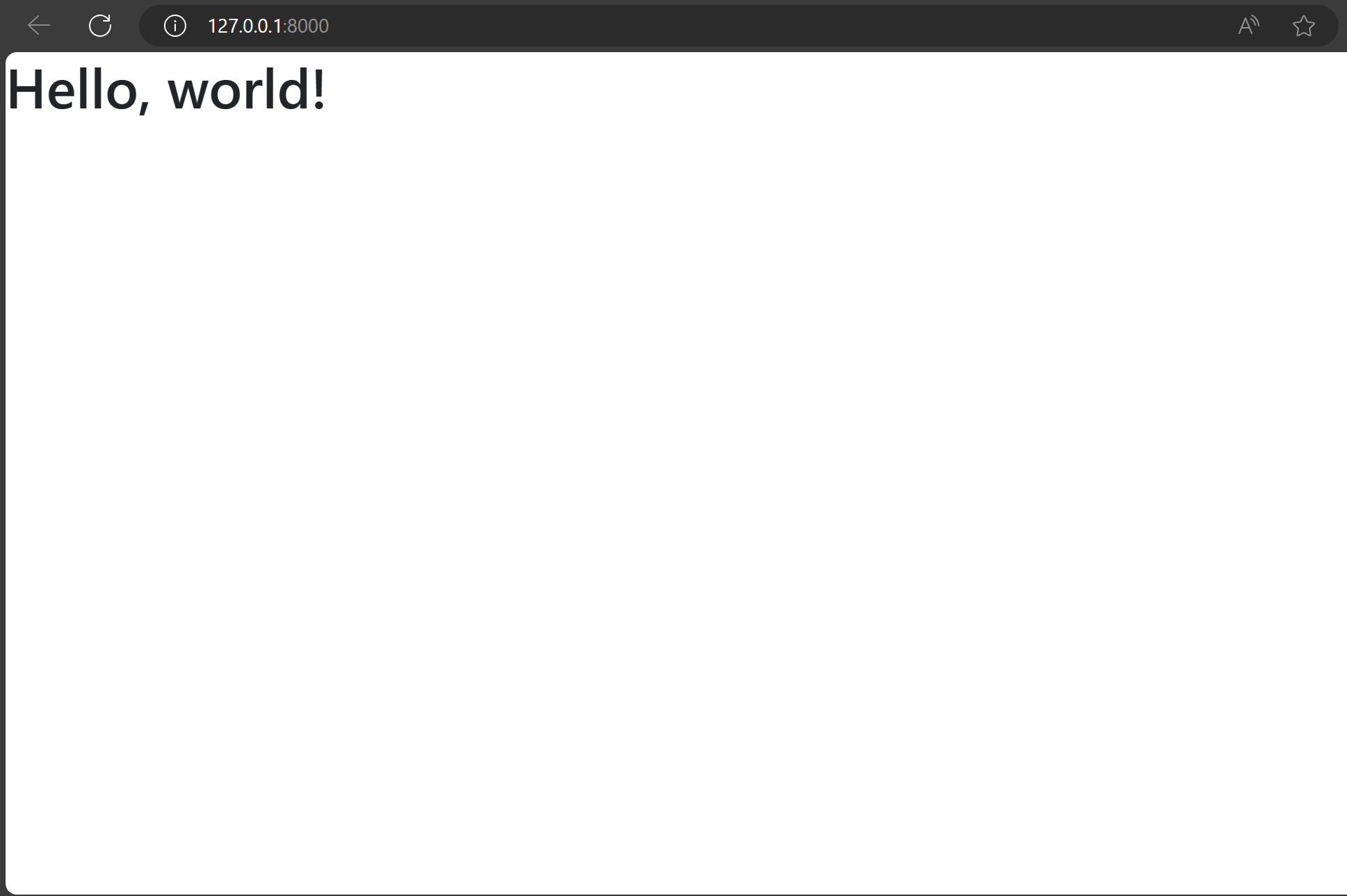
In the example given above the page design is hard-coded in the view. If we want to change the way the page looks, we will have to edit the python code. Django has a template system to separate the design from python by creating a template that the view can use.

Using a template in a view:

1. Create a directory called templates in todo directory; todo/templates
2. Create a directory in templates called todo; todo/templates/todo
3. Create a html file called index.html; todo/templates/todo/index.html
4. “todo/templates/todo/index.html”
5. “todo/views.py”



1. Make sure the development server is on and go to <http://127.0.0.1:8000>



Templating language - [Templates | Django documentation | Django (djangoproject.com)](https://docs.djangoproject.com/en/5.0/topics/templates/)

1. Variables

Variables outputs data from the view, a variable is a dict-like object mapping keys to values. Variables are surrounded by {{ first\_name }}.

1. Tags

Tags provide arbitrary logic in the rendering process. A tag can output content, serve as a control structure e.g if statements.

Tags are surrounded by {% %}

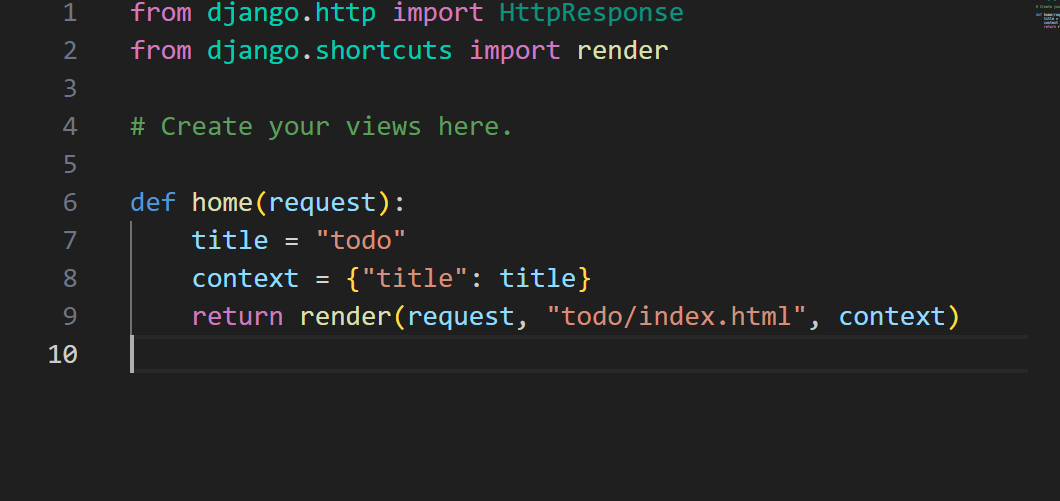
1. Filters

Filters transform the values of variables and tag arguments e.g

{{ first\_name|capitalize}}

Example of using django templating language

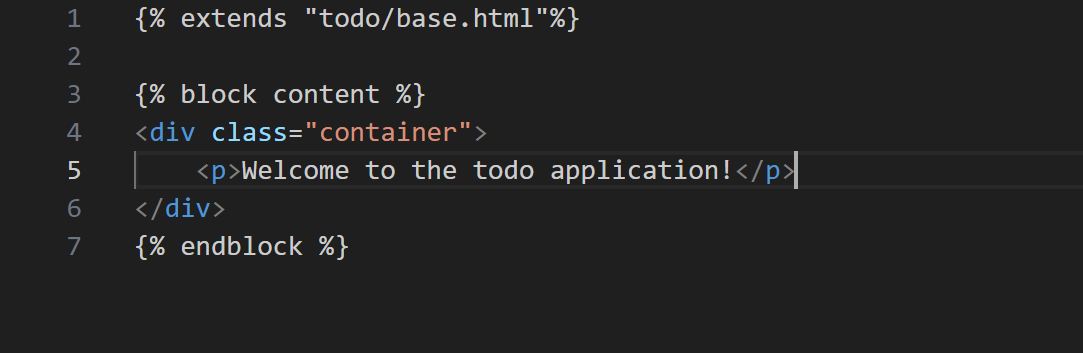
1. “todo/views.py”



1. “todo/templates/todo/base.html”



1. “todo/templates/todo/index.html”



1. “If everything is the same the page should show this:”

