
ВВЕДЕНИЕ В
DJANGO

План занятия

1) Знакомство с Django
2) Первый проект
3) Папка templates

Знакомство с Django

Django — веб-фреймворк для создания сайтов, написанный на языке Python. Язык и является одним из основных его преимуществ, так как обеспечивает быстрое написание кода.

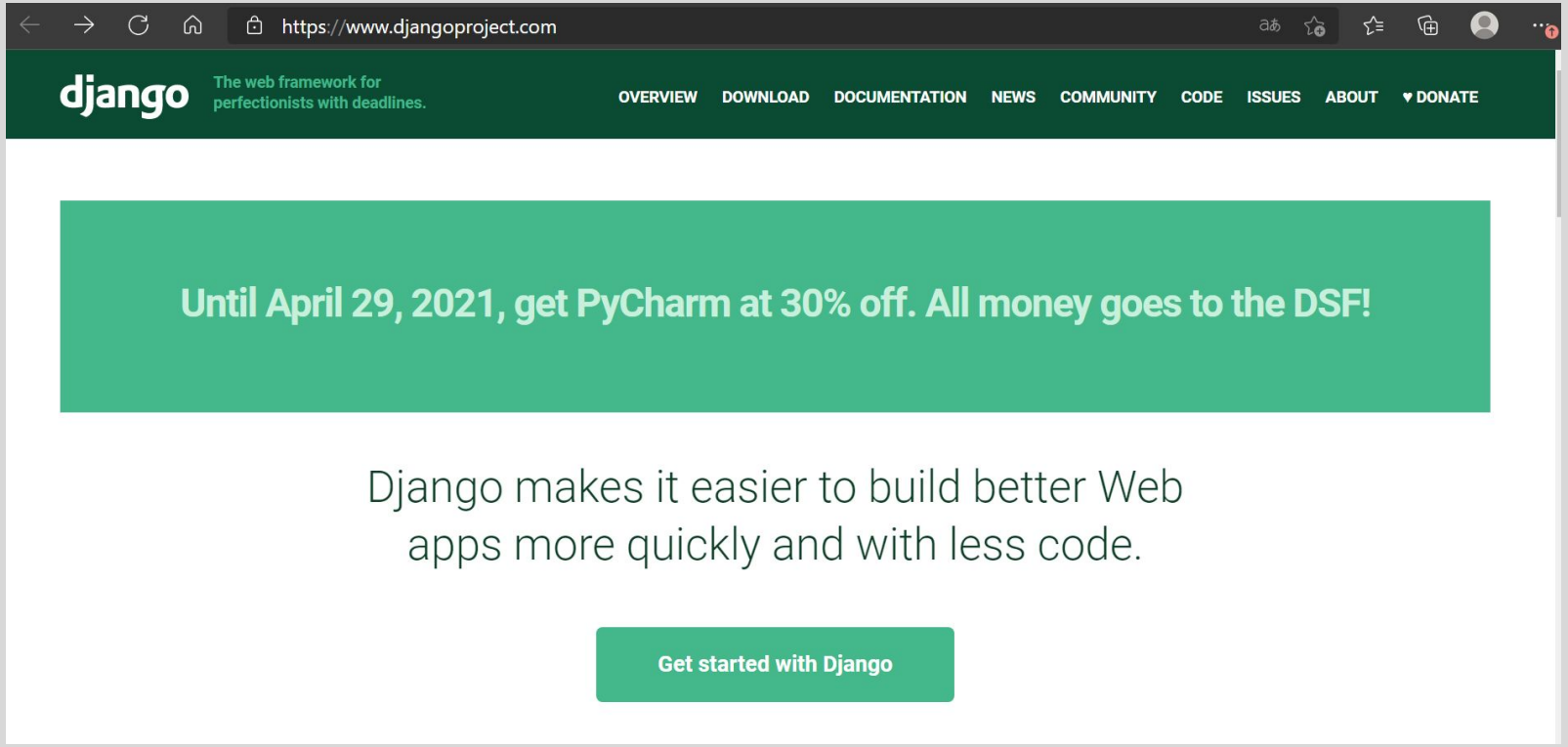
Также Django имеет "батарейки в комплекте", что означает, что для многих распространённых задач уже есть написанная библиотека. На текущий момент Django считается основным фреймворком Python для разработки веб-сайтов и веб-сервисов

Почему Джанго?

1. Это очень легко переключить базу данных в фреймворке Django.
2. Он имеет встроенный интерфейс администратора, который облегчает работу с ним.
3. Django — это полностью функциональная структура, которая больше ничего не требует.
4. Имеются тысячи дополнительных пакетов.
5. Это очень масштабируемо.

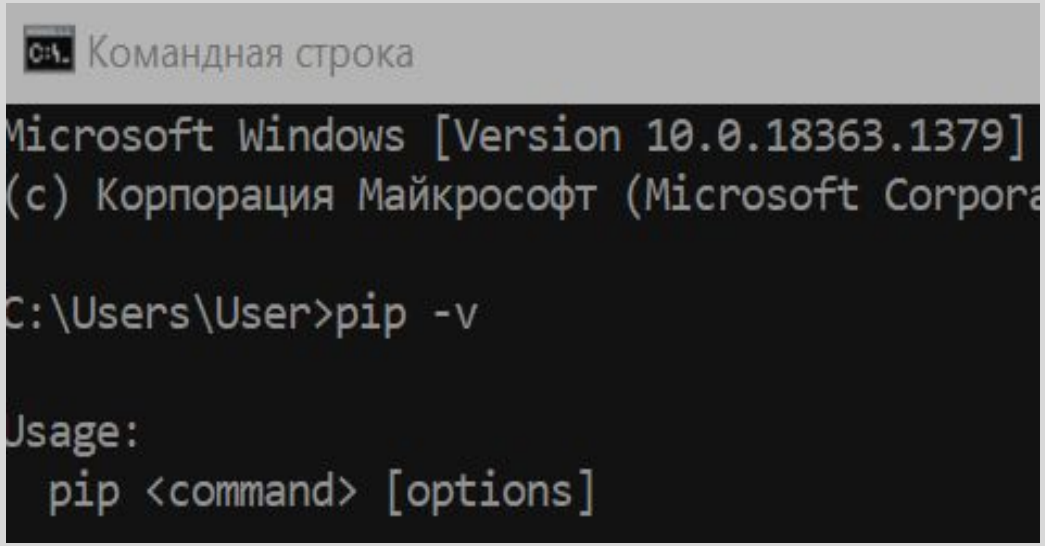
По ссылке ниже вы попадёте на официальный сайт Django. Там вы найдёте своеобразный туториал по установке Джанго, документация Джанго, новости и т.д.

[The Web framework for perfectionists with deadlines | Django (djangoproject.com)]

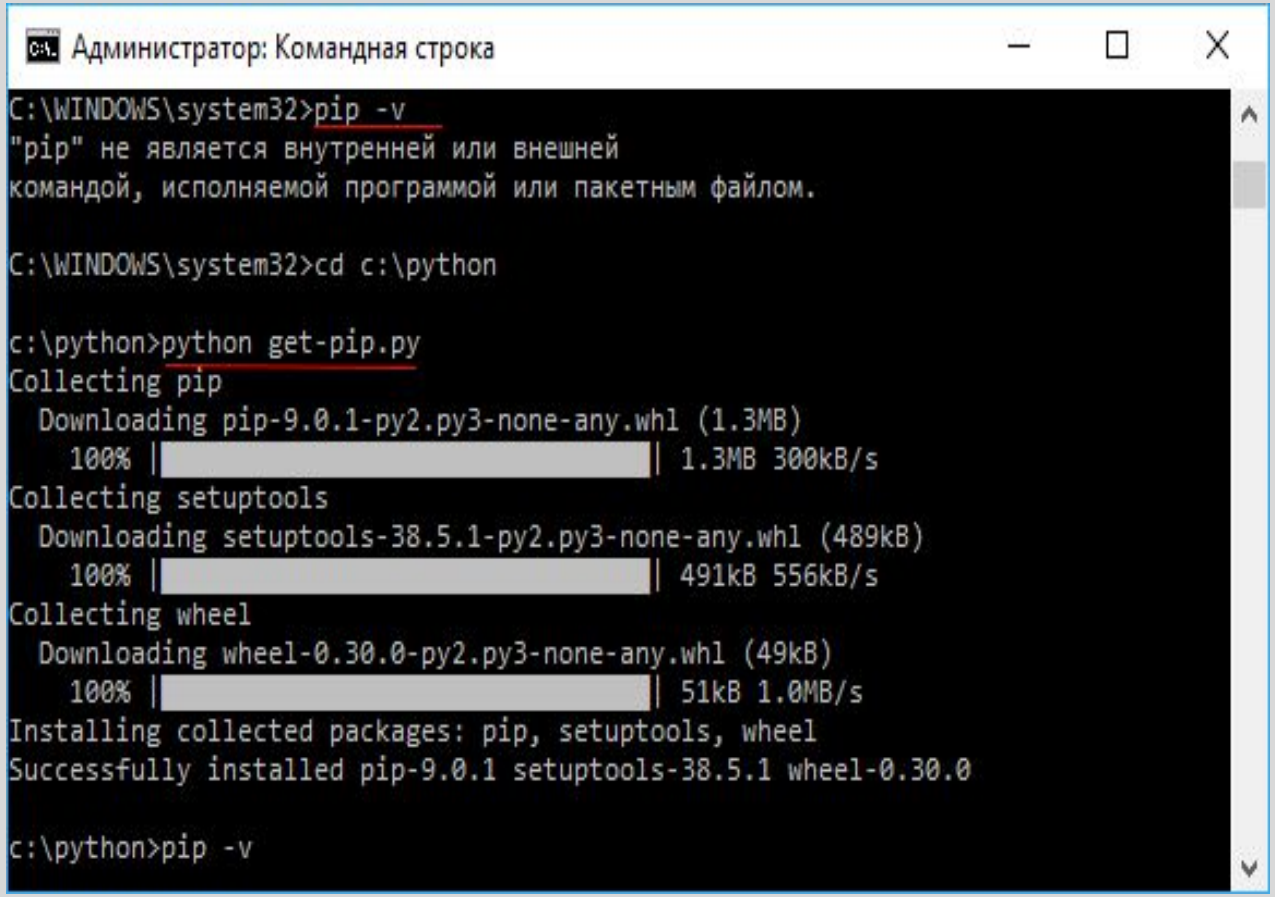


Установка Django

Для установки нам потребуется пакетный менеджер **pip**. Менеджер pip позволяет загружать пакеты и управлять ими. Нередко, при установке python также устанавливается и менеджер pip. В этом случае мы можем проверить версию менеджера, выполнив в командной строке/терминале следующую команду:

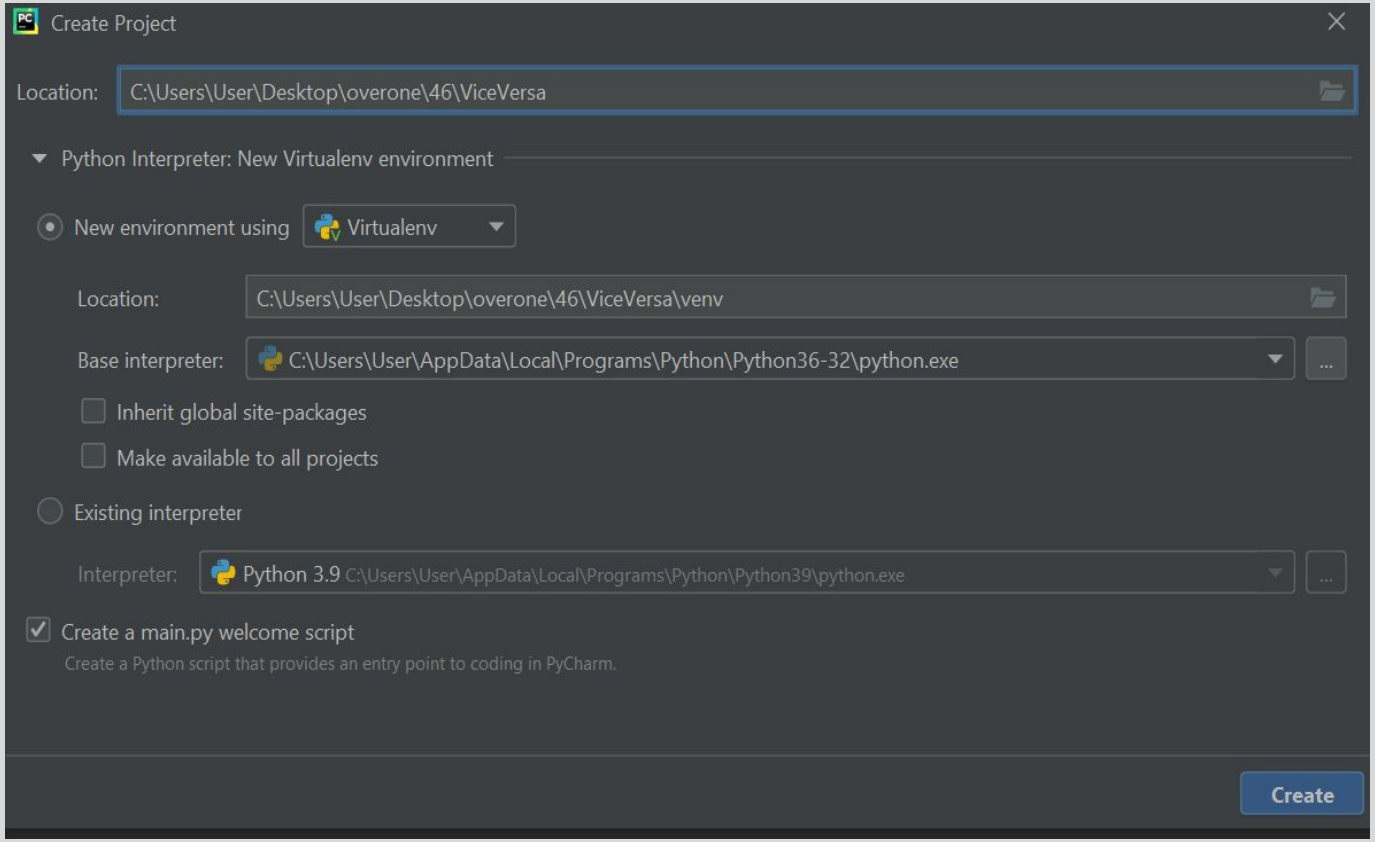


Но если pip не установлен, то мы увидим ошибку: "pip" не является внутренней или внешней командой, исполняемой программой или пакетным файлом. В этом случае нам надо установить pip.



Установка виртуальной среды

Виртуальная среда или **virtualenv** не является неотъемлимой частью разработки на Django. Однако ее рекомендуется использовать, так как она позволяет создать множество виртуальных сред Python на одной операционной системе. Благодаря виртуальной среде приложение может запускаться независимо от других приложений на Python.



Установите Django

Установите Django, **введя** следующую команду:

pip install django

```
(venv) C:\Users\User\Desktop\overone\46\ViceVersa>pip install django
Collecting django
  Downloading Django-3.2-py3-none-any.whl (7.9 MB)
     |████████████████████████████████| 7.9 MB 726 kB/s
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.1-py3-none-any.whl (42 kB)
Collecting pytz
  Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.3.4-py3-none-any.whl (22 kB)
Collecting typing-extensions
  Downloading typing_extensions-3.7.4.3-py3-none-any.whl (22 kB)
Installing collected packages: typing-extensions, sqlparse, pytz, asgiref, django
```

Создадим первый проект

Создайте новый проект в PyCharm. Назовём его ViceVersa. Затем переходим в терминал и продолжаем работу там:

django startproject viceversa

```
C:\Users\User\Desktop\overone\46\ViceVersa>py -m django startproject viceversa

C:\Users\User\Desktop\overone\46\ViceVersa>
```

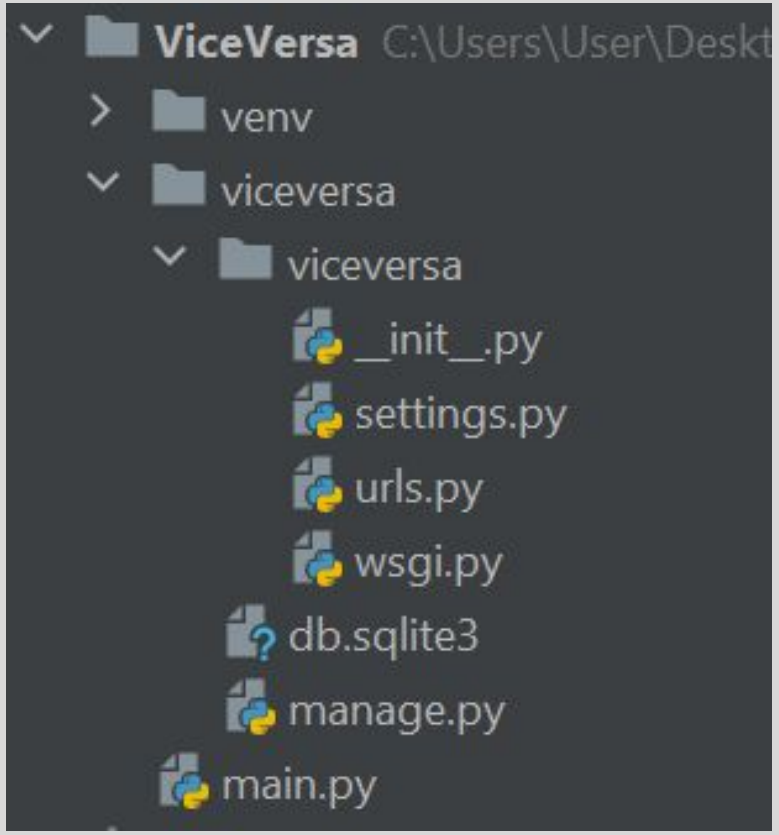
В данном пакете лежат файлы:

__init.py__ – пустой файл, сообщающий Python, что папка, в которой он находится, является полноценным пакетом.

settings.py – модуль с настройками самого проекта. Включает описание конфигурации базы данных проекта, пути ключевых папок, важные параметры, связанные с безопасностью.

urls.py – модуль с маршрутами уровня проекта.

wsgi.py – модуль, связывающий проект с веб-сервером. Используется при публикации готового сайта в Интернете.



Отладочный сервер Django

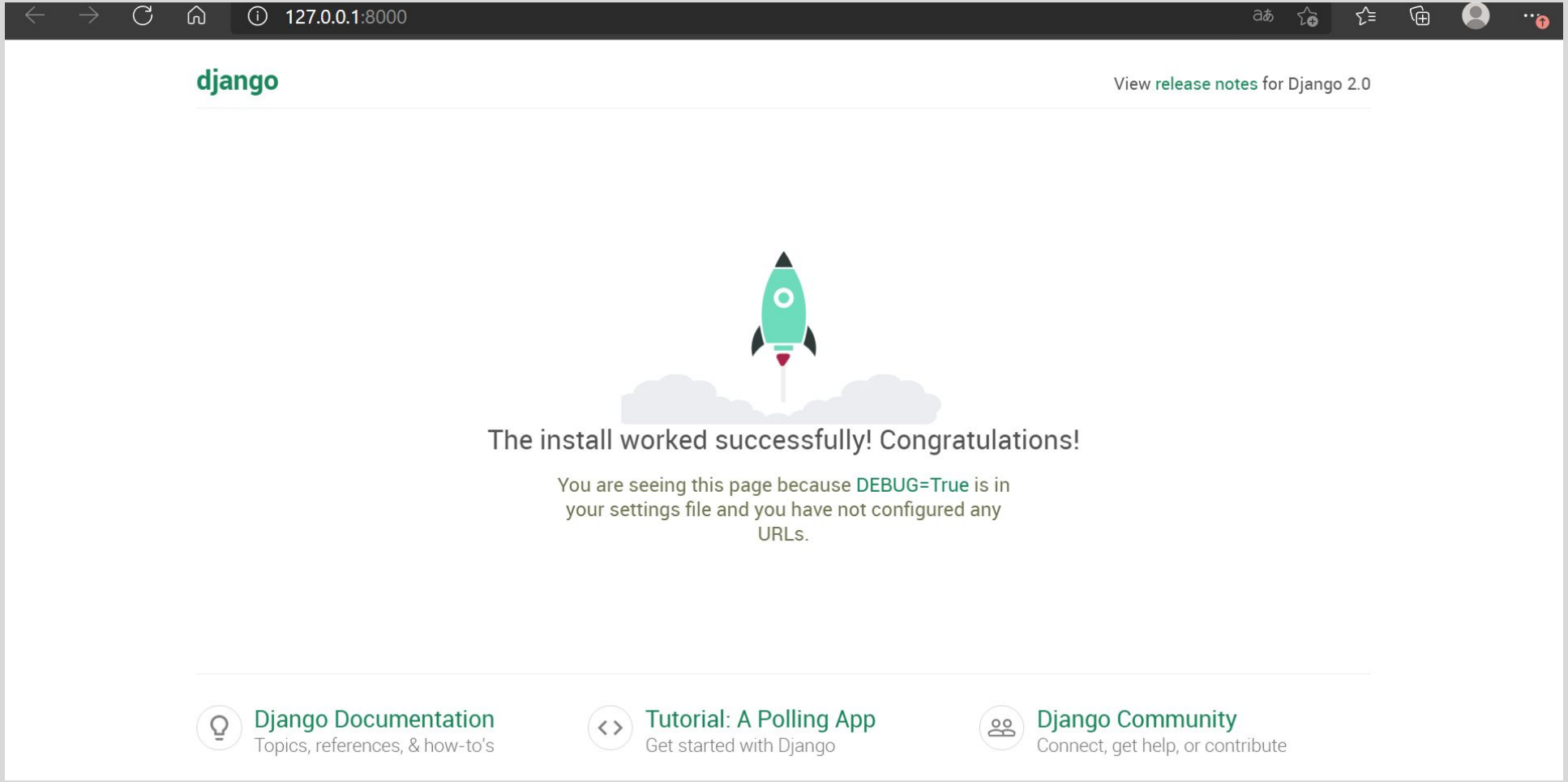
В состав Django входит отладочный веб-сервер, написанный на языке Python, не требующий сложной настройки и всегда готовый к работе.
Перейдём в папку с проектом.
Чтобы запустить веб-сервер необходимо выполнить команду:

cd имя_проекта
manage.py runserver

```
C:\Users\User\Desktop\overone\46\ViceVersa>cd viceversa

C:\Users\User\Desktop\overone\46\ViceVersa\viceversa>manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until
Run 'python manage.py migrate' to apply them.
April 09, 2021 - 20:05:32
Django version 2.0.5, using settings 'viceversa.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Примечание: чтобы остановить сервер используйте комбинацию ctrl + C

Поздравляю с успешным созданием проекта!!!



manage.py

Помогает делать административные команды верхнего уровня:

1. Добавляет пакет проекта в sys.path
2. Устанавливает переменную окружения DJANGO_SETTINGS_MODULE, чтобы она указывала на файл settings.py проекта

```
python manage.py help
```

Список команд, которыми мы можем управлять нашим сайтом при помощи manage.py

Нельзя редактировать, пока не станете экспертами в Джанго

db.sqlite3

Файл Базы Данных
Удобно открывать и просматривать с помощью sqlitestudio-2.1.3

settings.py

Показывает, где находится основная директория проекта:

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

Секретный ключ проекта(нельзя никому показывать):

```
23      SECRET_KEY = '2y4_87xk)unl^^-h-9$y1-4b@t(u7)3ne=n#w3u6_1z9%#wu+d'
24
```

Режим разработчика включен, т.е. позволяет видеть ошибки:

```
26      DEBUG = True
```

Более подробно этот файл будем разбирать дальше

urls.py

Каждый раз, когда кто-то будет посещать Ваш сайт он будет попадать в urlpatterns(list with path(лист с путями)), а этот список будет решать, куда запрос пользователя будет перенаправляться

```
16  from django.contrib import admin
17  from django.urls import path
18
19  urlpatterns = [
20      path('admin/', admin.site.urls),
21  ]
```

В urls.py добавим новый путь about. И дальше мы должны связать этот путь с каким-то python файлом (например, в котором мы сможем устанавливать какой-то текст на этой странице)

```
19  urlpatterns = [
20      path('admin/', admin.site.urls),
21      path('about/', admin.site.urls),
22  ]
```

Для этого создадим новый файл views.py и напишем его импорт в urls.py

```
16  from django.contrib import admin
17  from django.urls import path
18  from . import views
```

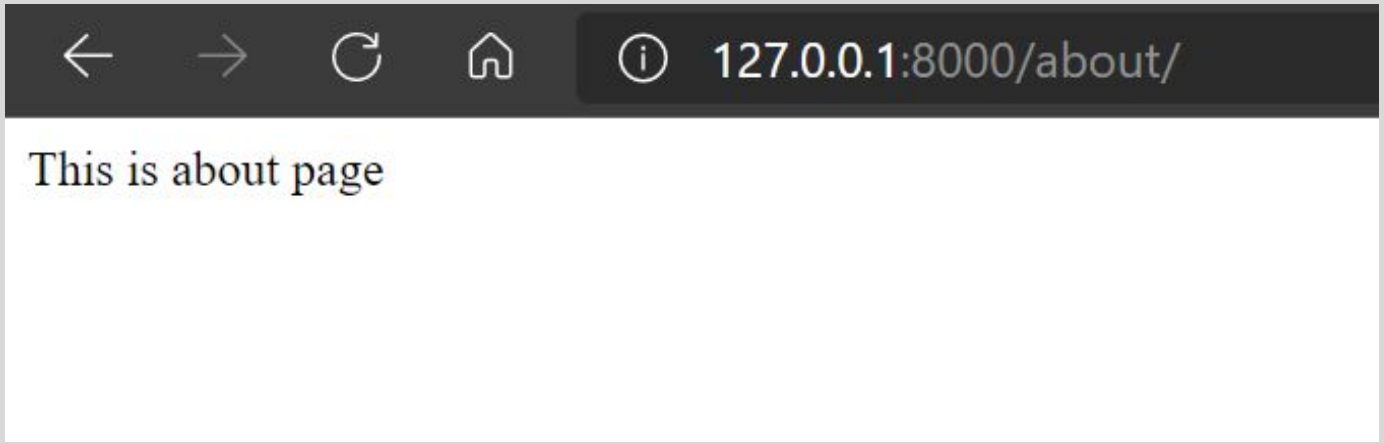
views.py

Создадим функцию about с параметром request (т.е. каждый раз, когда кто-то что-то делает на Вашем сайте, он отправляет request – запрос).
Функция представления принимает один аргумент request. Этот объект HttpRequestObject создается, когда страница загружается. Он содержит информацию о запросе, такую как метод, который может принимать несколько значений, включая GET и POST
Для того, что вернуть Http ответ импортируем его из соответствующей библиотеки.
Вернём ответ в виде строки

Перейдём в urls.py свяжем путь about с нашей функцией.

```
20  urlpatterns = [
21      path('admin/', admin.site.urls),
22      path('about/', views.about),
23  ]
```

Перейдём в браузере на страницу 127.0.0.1:8000/about/



Задание №1

Создайте путь home/ и привяжите к нему функцию, которая выведет: “My home”

Решение

urls.py

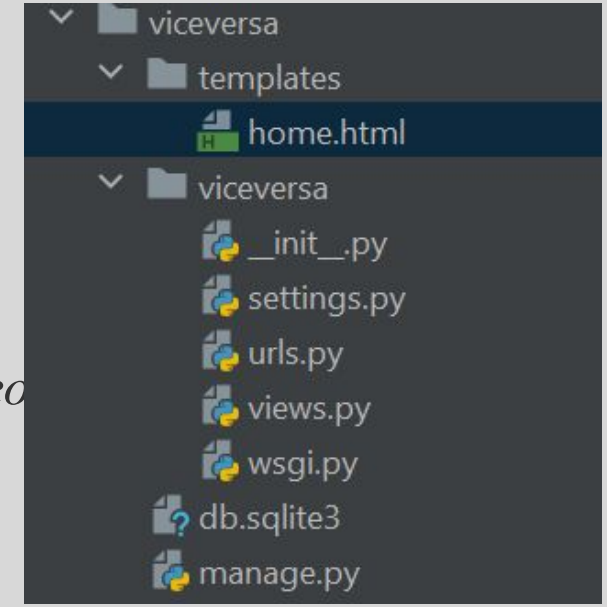
```
20  urlpatterns = [
21      path('admin/', admin.site.urls),
22      path('about/', views.about),
23      path('home/', views.home),
24  ]
```

views.py

```
1  from django.http import HttpResponse
2
3  def about(request):
4      return HttpResponse('This is about page')
5
6  def home(request):
7      return HttpResponse('My home')
```

Templates

В папке проекта создадим новую папку с именем: templates. В ней создадим новый файл с расширением html.
Эта папка хранит в себе шаблоны, которые отвечают за формирование внешнего вида приложения. Они представляют специальный синтаксис, который позволяет внедрять данные в код html



Справка
***HTML** (от английского HyperText **Markup Language**) — это язык гипертекстовой разметки страницы. Он используется для того браузеру понять, как нужно отображать загруженный сайт.*

settings.py

Перейдя в данный файл, находим раздел templates. Он содержит лист DIRS, в котором мы можем указывать все места, в которых Джанго должен искать шаблоны.

В нём мы указываем нашу созданную директорию templates в кавычках.

Это строка говорит Джанго, что он должен искать шаблоны внутри данной папки

views.py

Теперь изменим нашу функцию home, чтобы на странице текст выводился из файла home.html
Для этого импортируем функцию render. Затем в return вызываем эту функцию и передаём в неё 2 параметра: первый – request, второй – имя html файла

```python
from django.http import HttpResponse
from django.shortcuts import render


def about(request):
    return HttpResponse('This is about page')


def home(request):
    return render(request, 'home.html')

```

HTML

Основы HTML содержат основные правила языка HTML, описание структуры HTML-страницы, отношения в структуре HTML-документа между HTML-элементами.

HTML-документ — это обычный текстовый документ, может быть создан как в обычном текстовом редакторе (Блокнот), так и в специализированном, с подсветкой кода (Notepad++, Visual Studio Code и т.п.). HTML-документ имеет расширение .html.

HTML-документ состоит из дерева HTML-элементов и текста. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением).

Начальный тег показывает, где начинается элемент, конечный — где заканчивается. Закрывающий тег образуется путем добавления слэша / перед именем тега: <имя тега>…</имя тега>. Между начальным и закрывающим тегами находится содержимое элемента — контент.

Элементы, представленные одиночными тегами, не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, элемент <input type="button" value="Кнопка"> создаст кнопку с текстом Кнопка внутри.

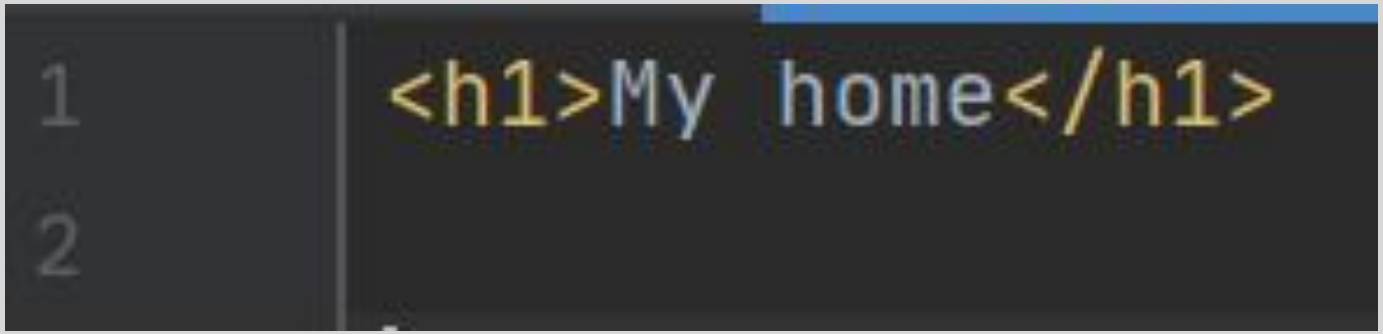
Ссылка на документацию: Основы HTML (html5book.ru)

Список базовых тэгов HTML

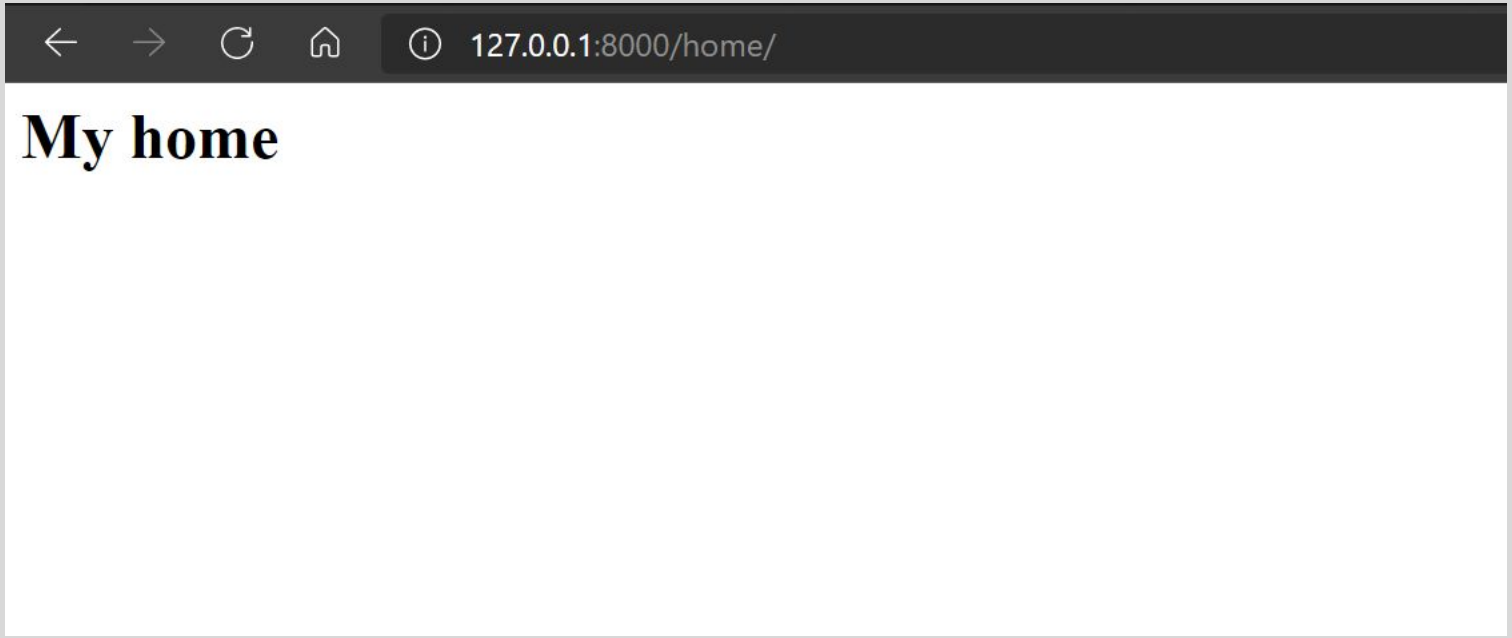
Стартовый	Завершающий	Описание
<HTML>	</HTML>	Обозначение HTML-документа
<HEAD>	</HEAD>	Заголовочная часть документа
<TITLE>	</TITLE>	Заголовок документа
<BODY>	</BODY>	Тело документа
<H1>	</H1>	Заголовок абзаца первого уровня
<H2>	</H2>	Заголовок абзаца второго уровня
<H3>	</H3>	Заголовок абзаца третьего уровня
<H4>	</H4>	Заголовок абзаца четвертого уровня
<H5>	</H5>	Заголовок абзаца пятого уровня
<H6>	</H6>	Заголовок абзаца шестого уровня
<P>	</P>	Абзац
<PRE>	</PRE>	Форматированный текст
 		Перевод строки без конца абзаца
<BLOCKQUOTE>	</BLOCKQUOTE>	Цитата

home.html

В данном случае мы используем простой тэг h1, в котором заключим наш текст.
В PyCharm предусмотрен вариант работы с HTML кодом.

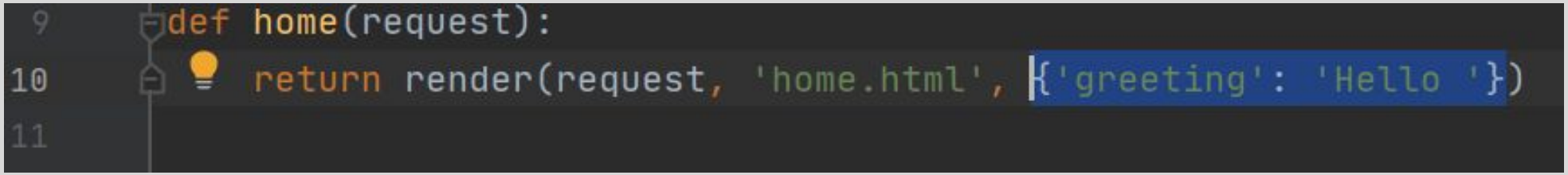


Результат



views.py

Изменим функцию render, добавив передачу в неё третьего параметра.
В качестве него может быть передан словарь, который заключается в {}.
В качестве ключа передадим greeting, в качестве значения для него Hello



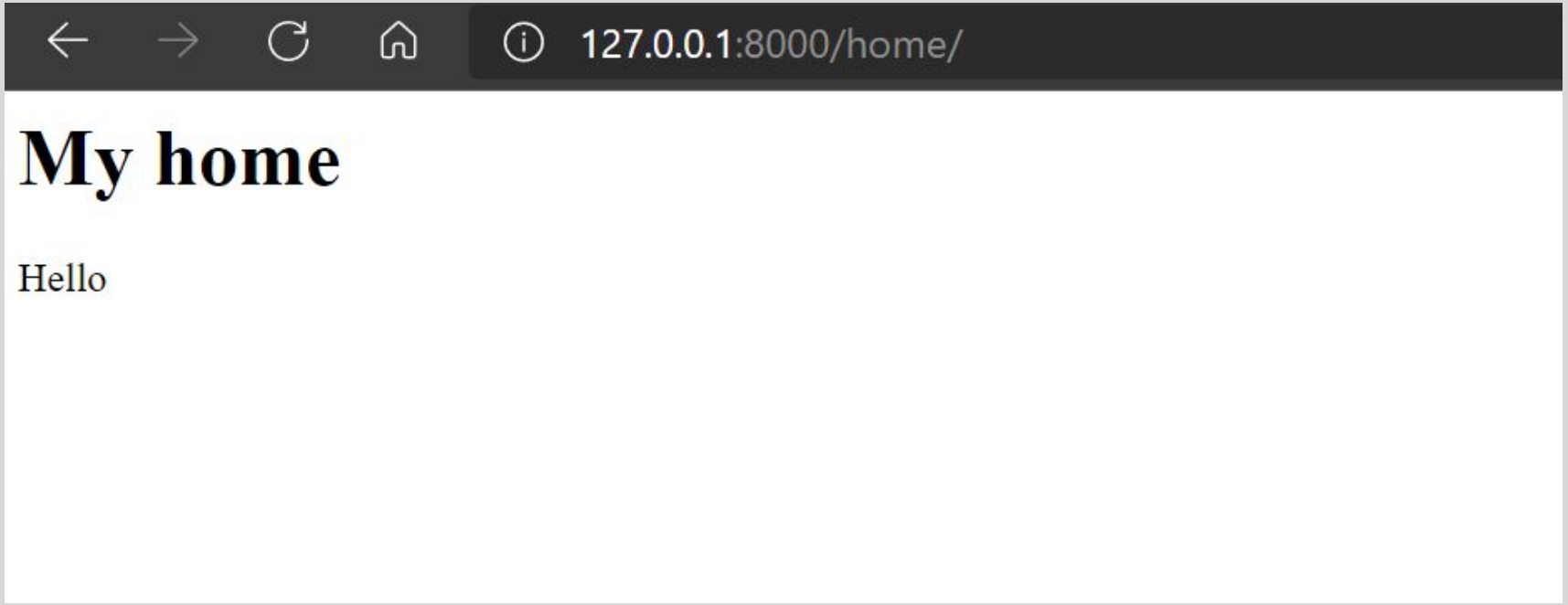
home.html

Укажем ключ в двух парах фигурных скобок {{}}

```
1    <h1>My home</h1>
2    {{greeting}}
3
```

Результат

Вывод
Получаем на нашей странице значение по данному ключу.
Т.е. мы можем использовать какой-то Python код, а не только html код.



urls.py

Для того, чтобы на главной странице у нас отображалась страница home.html. Изменим путь, оставив просто пустые кавычки

Домашнее задание

Создать PyCharm проект: Ваше Имя_2022
Создать новый Django проект: portfolio. В нём создать папку templates.
В сегодняшнем проекте изменить файл home.html, как показано на следующем слайде

Your_name – тэг **h1**
Для прямоугольника используйте тэг **textarea**
Для кнопки используйте либо тэг button, либо
<input type = “submit”…