

## Разработка программного обеспечения на языке Python

[Обзорная панель](#) ▶ [Мои курсы](#) ▶ [Разработка ПО на языке Python](#) ▶ [Веб-программирование на Python](#) ▶

[Лекция 2. Библиотека Django](#)

### Лекция 2. Библиотека Django

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

#### Создание проекта Django



#### Создание проекта

Команда

`django-admin startproject [имя проекта]`

00:00 / 04:53



В данной теме рассмотрим создание проекта веб-приложения с использованием фреймворка Django.

**Виртуальная среда** (ее иногда также называют виртуальное окружение) позволяет разграничить версии и набор библиотек python для каждого проекта. Для каждого проекта будет свой набор установленных библиотек, в том числе разных версий. Тогда в случае обновления какой-то библиотеки не потребуется вносить изменения в ранее созданные проекты.

Для работы с виртуальной средой в python применяется встроенный модуль `venv`. Он входит в стандартную библиотеку Python.

Создадим виртуальную среду. Вначале определим каталог, в котором будет создан проект. Перейдем в командной строке/терминале в этот каталог с помощью команды `cd`. Затем для создания виртуальной среды выполним следующую команду.

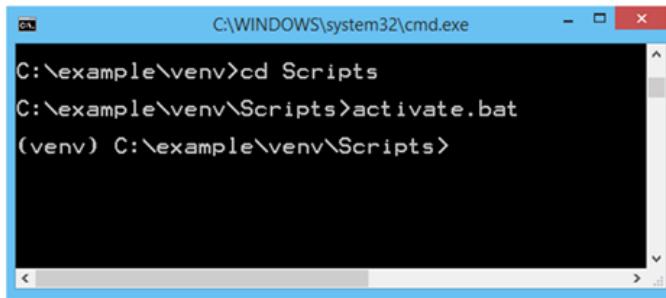
**`python -m venv venv`**

**`venv`** – название виртуальной среды (может быть произвольным)

Модулю `venv` передается название среды, которая в данном случае будет называться также `"venv"`. После этого в текущей папке будет создан подкаталог с названием виртуальной среды.

Для использования виртуальной среды ее нужно предварительно активировать с помощью исполнения файла `activate.bat`. Указанный файл находится в папке с виртуальной средой (в нашем случае название папки `venv`) в подкаталоге `Scripts`.

## venv/Scripts/activate.bat



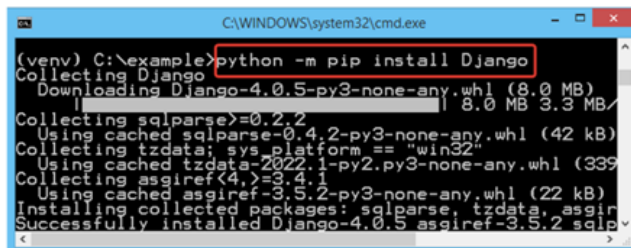
Это нужно делать каждый раз при работе с проектом на Django. После окончания работы виртуальную среду можно деактивировать с помощью файла `deactivate.bat`.

Далее перед началом работы нам требуется установить библиотеку Django. Для этого нам потребуется пакетный менеджер `pip`, который позволяет загружать различные библиотеки и управлять ими.

## pip -V – проверка версии

После активации виртуальной среды для установки Django выполним в консоли следующую команду: `python pip install Django`.

## python -m pip install Django

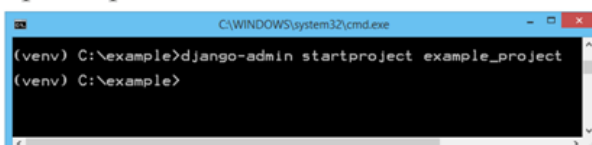


Перейдем в консоли к созданному для проекта каталогу с помощью команды `cd`.

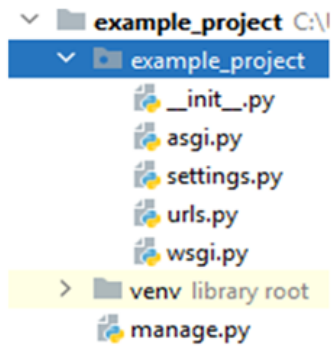
При установке Django в папке виртуальной среды устанавливается утилита **django-admin**. Она предоставляет команды для управления проектом Django. В частности, для создания проекта применяется команда **startproject**. Этой команде в качестве аргумента передается название проекта. Итак, вначале активируем ранее созданную виртуальную среду (например, среду `venv`, которая была создана ранее). И после этого выполним следующую команду: `django-admin startproject example_project`. В данном случае мы создаем проект с названием `example_project`.

## Команда django-admin startproject [имя проекта]

### Пример



И после выполнения этой команды в текущей папке будет создан подкаталог с таким же названием. Рассмотрим общую структуру проекта.



Проект веб-приложения будет состоять из следующих элементов:

Файла **manage.py**, который нужен для выполнения различных команд, например, запуск веб-приложения. А также папки, название которой совпадает с названием проекта. В нашем случае, *example\_project*, в которой содержатся следующие файлы.

**\_\_init\_\_.py**: - указывает, что папка, в которой он находится, будет рассматриваться как модуль Python.

**settings.py**: содержит настройки конфигурации проекта

**urls.py**: определяет систему маршрутизации проекта

**wsgi.py** и **asgi.py**: содержат свойства конфигурации при развертывании проекта.

Запустим проект на выполнение. Для этого с помощью команды `cd` перейдем в консоли к папке проекта. И затем для запуска сервера выполним следующую команду: `python manage.py runserver`.

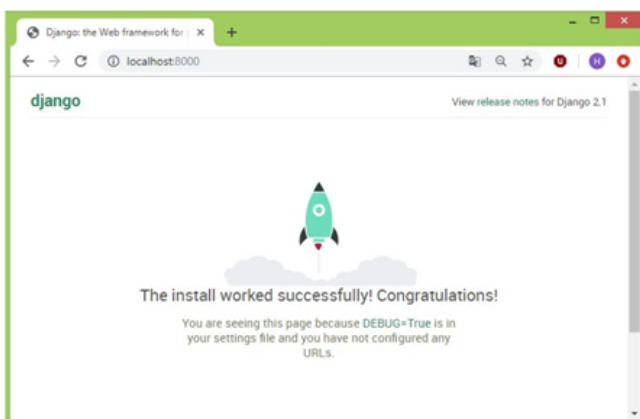
## python manage.py runserver

### Пример.

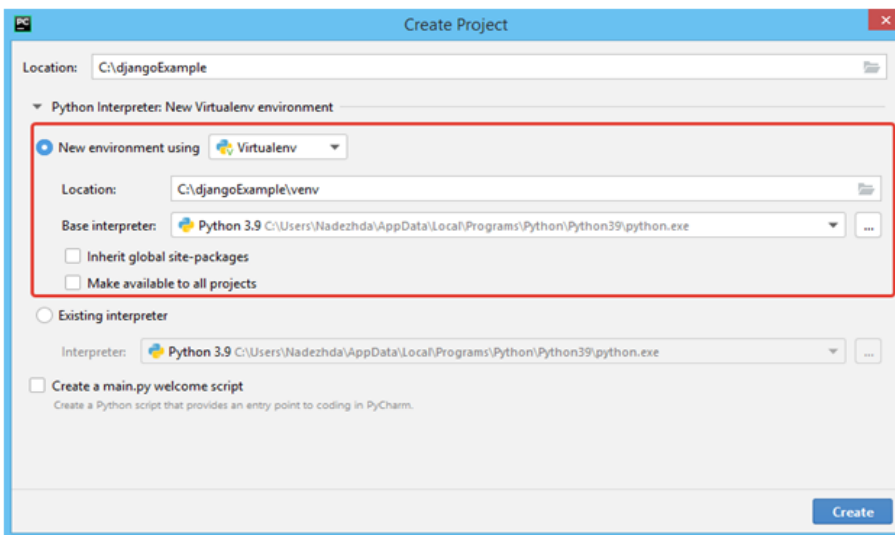
```
(venv) C:\example\example_project>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

После этого в консоли мы увидим адрес, по которому запущен проект. Как правило, это адрес **http://127.0.0.1:8000/**.

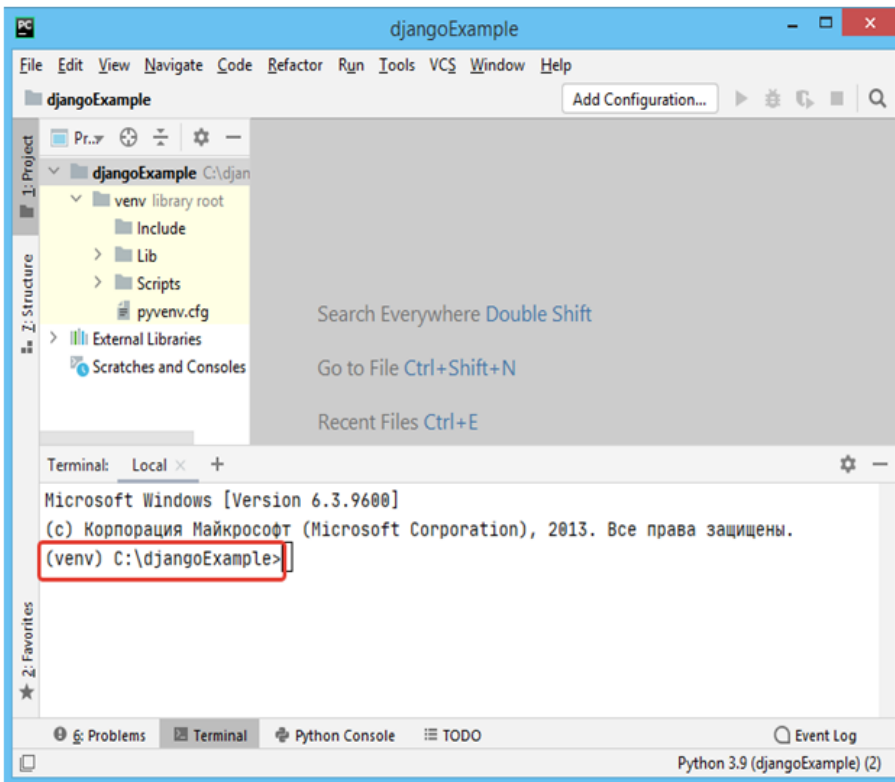
Откроем любой веб-браузер и введем данный адрес в адресную строку браузера. И нам откроется содержимое по умолчанию.



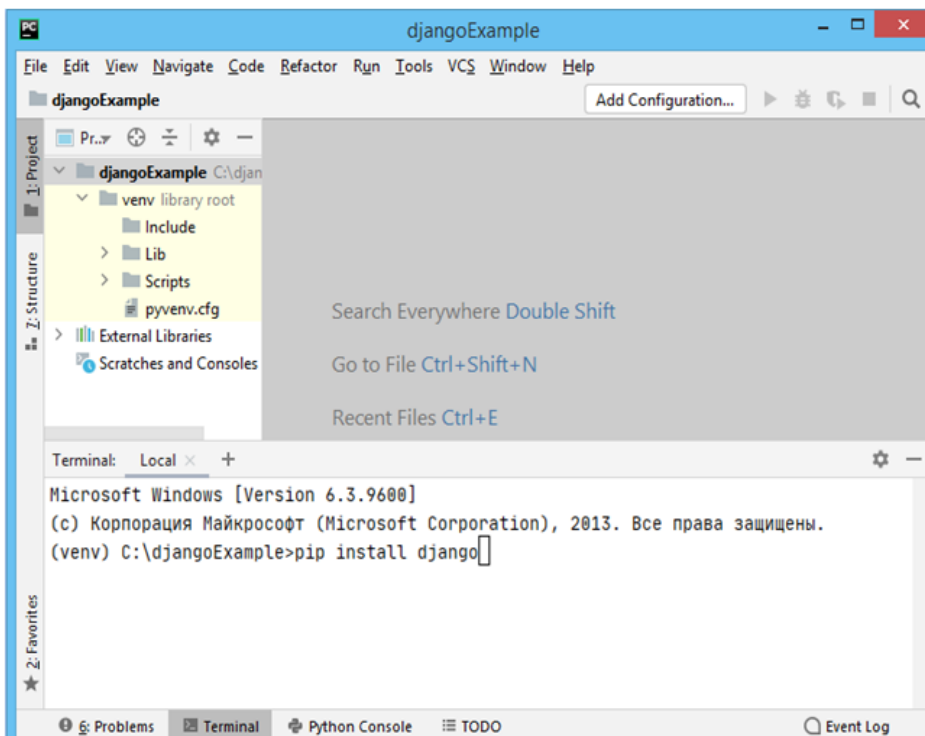
Если проект создается в среде разработки **pyCharm** - можно создать новый проект python сразу с виртуальной средой (командой **File->New->Project**)



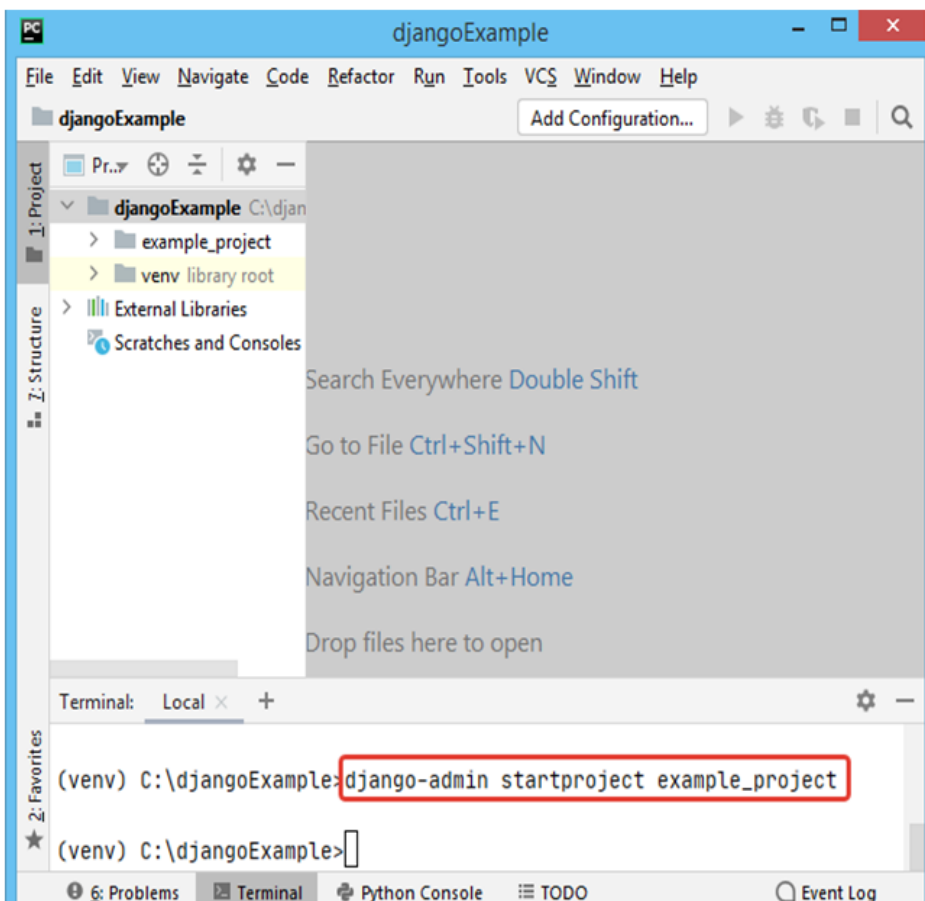
При этом в созданном проекте виртуальная среда уже активирована, что можно видеть на рисунке.



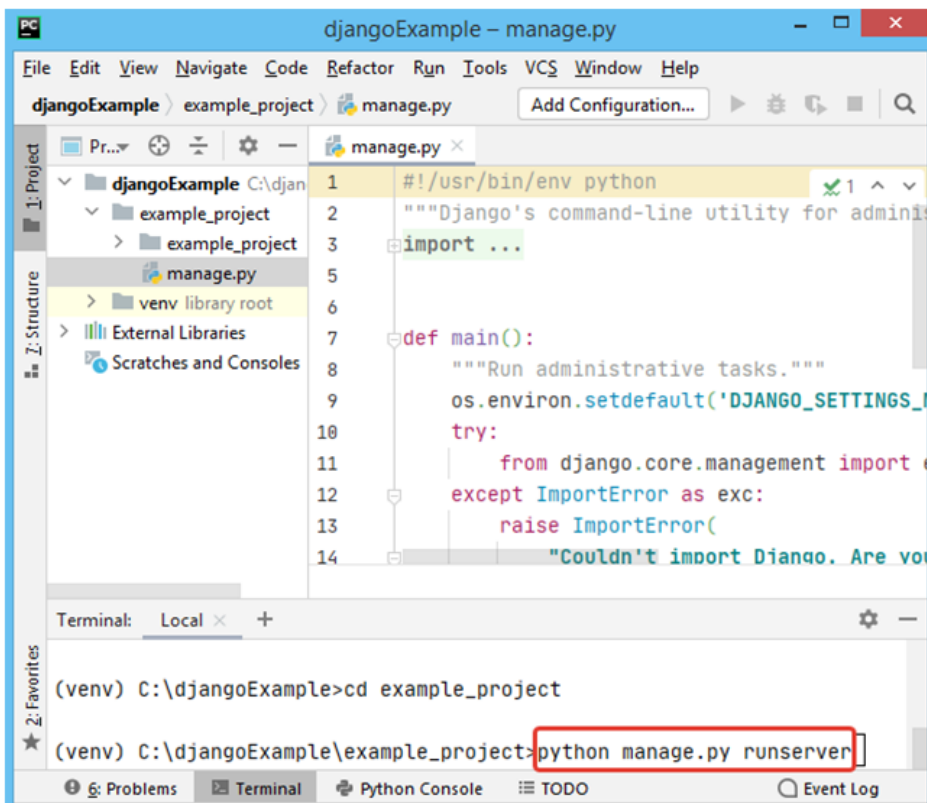
Далее переходим в консоль и используем те же команды для установки django



и создания проекта.



Запускать такой проект можно также в консоли.



Подведем итоги, в данной теме мы рассмотрели основные этапы создания проекта веб-приложения на django. Однако, для наполняемости требуется создать внутри проекта приложения, в которых будет содержаться наполнение нашего сайта.

Создание приложения Django

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◀ [Задание 2. Создание flask приложения](#)

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

[Задание 3. Создание своего первого сайта на Django](#) ▶

© 2010-2023 Центр обучающих систем  
Сибирского федерального университета, sfu-kras.ru

Разработано на платформе moodle  
Beta-version (3.9.1.5.w3)

[Политика конфиденциальности](#)

[Соглашение о Персональных данных](#)

[Политика допустимого использования](#)

Контакты +7(391) 206-27-05  
[info-ms@sfu-kras.ru](mailto:info-ms@sfu-kras.ru)

[Скачать мобильное приложение](#)

