

## Практика 11 - Создание веб сайтов на Python

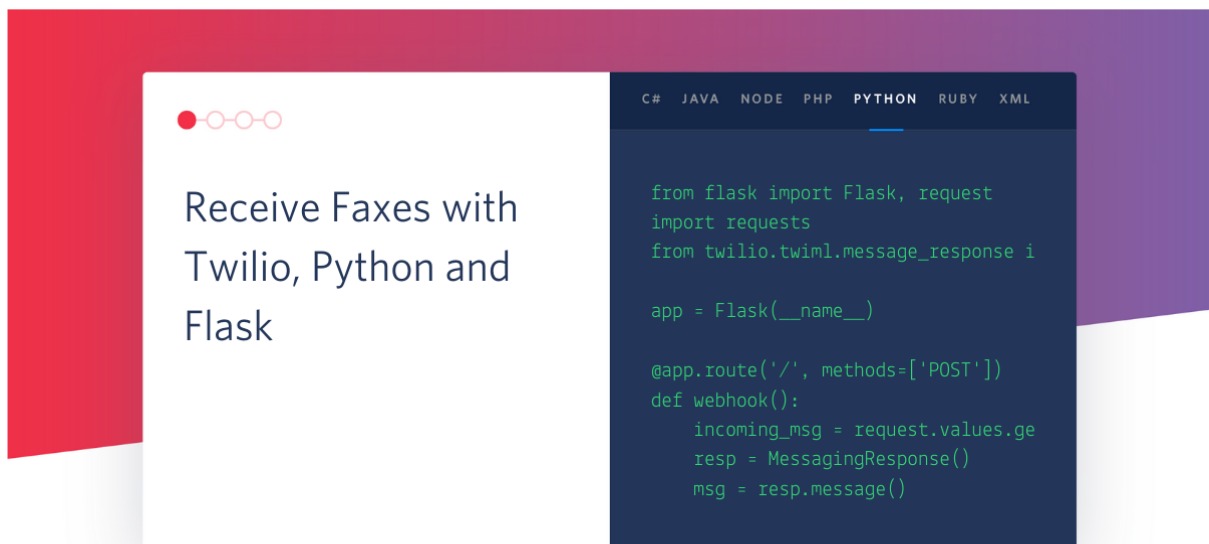
### Полезные ссылки:

- Курс по [Django](#);
- Курс по [языку Python](#);
- Программа [PyCharm](#).

## Информация про Flask

Фреймворк Flask относится к так званым микрофреймворкам - минималистичные каркасы, которые обеспечивают лишь базовые возможности для реализации той или иной задачи.

Что это означает простым языком? А то, что создать сайт на **Flask** это очень просто и на изучение фреймворка вы не потратите больше одной недели. Это крутой показатель, но как и в любом деле здесь есть свои плюсы и свои минусы.



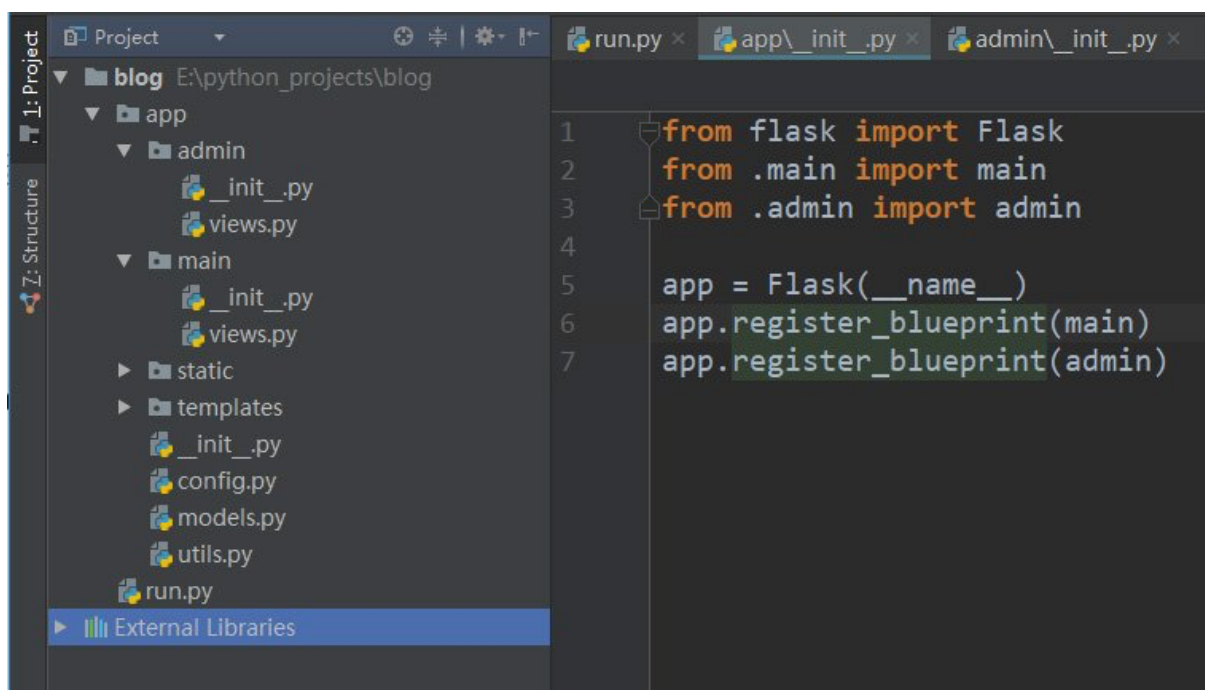
К плюсам относится быстрое изучение фреймворка, а к минусам относится его функциональность. За счёт Flask вы уже в ближайшие дни сможете создать свой сайт на языке Python, но при этом если вы захотите сделать сайт по типу Youtube, Google или сайта, который

требует взаимодействия с различного рода API, то данный фреймворк станет не идеальным выбором.

## Джанго или Flask?

И здесь мы подходим к ответу на один важный вопрос, который многие начинающие разработчики задают себе перед созданием сайтов на Питон. Какой фреймворк выбрать: Джанго или Flask?

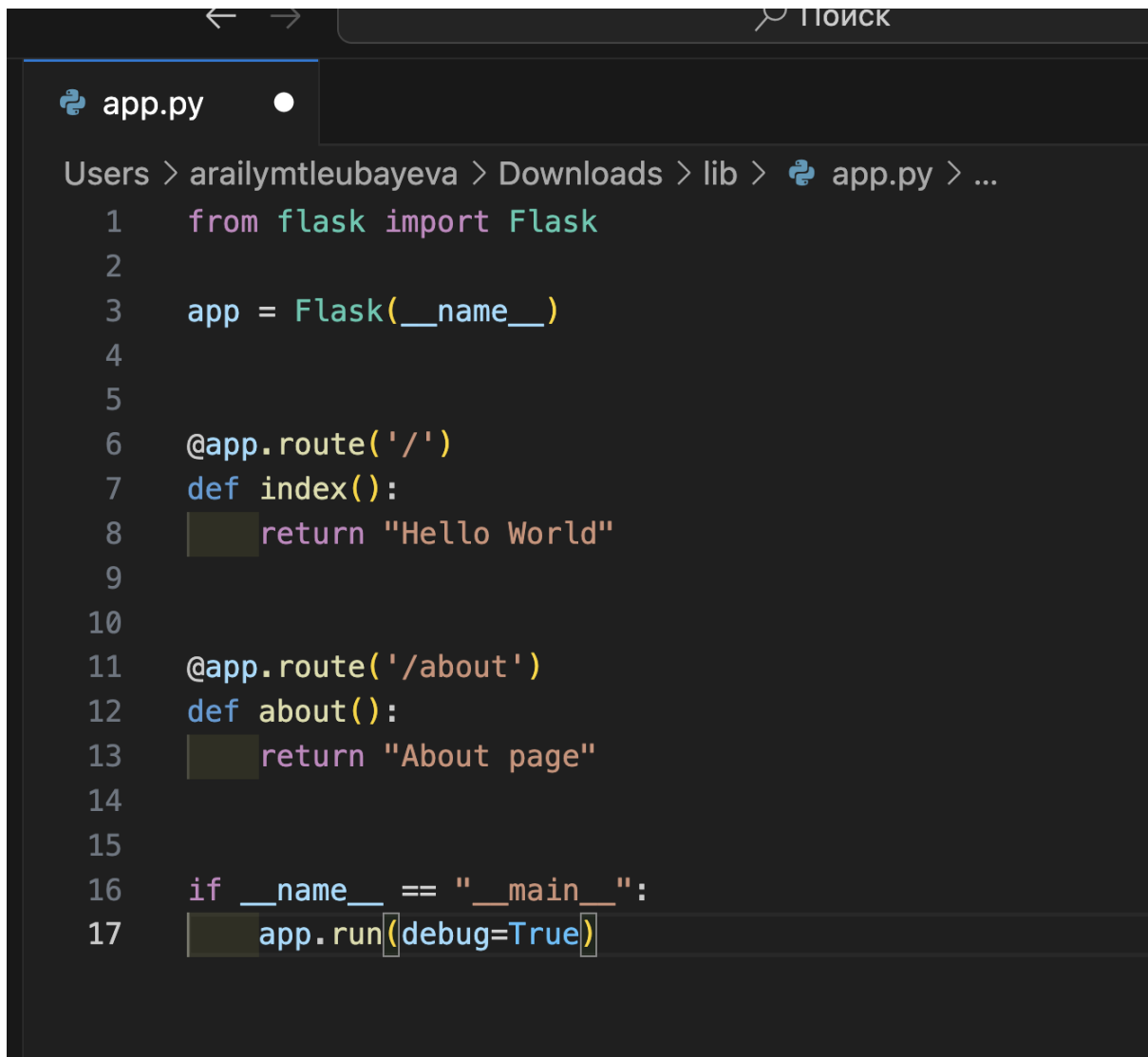
Ответить на этот вопрос достаточно просто. Если вас интересует самостоятельная разработка сайтов и вы быстро хотите вникнуть в сферу создания сайтов, то изучайте Flask. Если же времени у вас достаточно и вы не боитесь сложных фреймворков, то вашим выбором станет Джанго.



Сейчас мы не будем дискутировать что же круче и что же стоит выбрать. В любом случае, если вы хотите разрабатывать сайты на Питон, то данного курса вам вполне хватит, чтобы войти в эту сферу и начать заниматься делом.

## Исходный код

### Основной файл app.py



```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route('/')
7  def index():
8      return "Hello World"
9
10
11 @app.route('/about')
12 def about():
13     return "About page"
14
15
16 if __name__ == "__main__":
17     app.run(debug=True)
```

### Задание 1 -Установка Flask

- Скажите какая команда используется для установки Flask в проект?
- Также какая команда используется для установки библиотеки по работе с базой данных?

Для установки Flask используется команда:

`pip install flask`

Для установки библиотеки по работе с базой данных используется команда:

```
pip install flask-sqlalchemy
```

## **Задание 2 - Создание базового шаблона**

Создайте базовый шаблон HTML для вашего сайта, который будет использоваться как родительский шаблон для всех других страниц. Используйте Jinja2 шаблоны для наследования шаблона в дочерних страницах.

## **Задание 3 - Отслеживание URL**

Создайте простую страницу контактов для вашего сайта на Flask. Следующий код может быть использован в основном файле app.py для отслеживания URL /contacts:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/contacts')
```

```
def contacts():
```

```
    return 'Контакты'
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

## **Задание 4 - Интеграция с базой данных**

Создайте модель для пользователя в вашем Flask-приложении с использованием SQLAlchemy. Модель должна включать столбцы для ID, имени пользователя, электронной почты и пароля.

### **Задание 5 - Регистрация пользователей**

Реализуйте функционал регистрации пользователей на вашем сайте. Вам нужно будет создать HTML-форму и соответствующие маршруты в Flask для обработки данных формы и сохранения нового пользователя в базе данных.

### **Задание 6 - Аутентификация пользователей**

Добавьте функционал для аутентификации пользователей. Это должно включать создание маршрутов для входа и выхода, а также функционал для проверки паролей.

### **Задание 7 - Работа с формами**

Используйте Flask-WTF и Flask-Form для создания и обработки форм на вашем веб-сайте. Создайте форму для добавления новых записей в базу данных.

### **Задание 8 - Пользовательские сессии**

Научитесь работать с пользовательскими сессиями в Flask для отслеживания состояния входа пользователя в систему.

### **Задание 9 - Загрузка файлов**

Реализуйте функционал для загрузки файлов пользователями на ваш веб-сайт. Убедитесь, что обрабатываете загрузку безопасно.

## **Задание 10 - Создание API**

Создайте RESTful API с использованием Flask, который позволит выполнять CRUD операции с ресурсами вашего сайта.