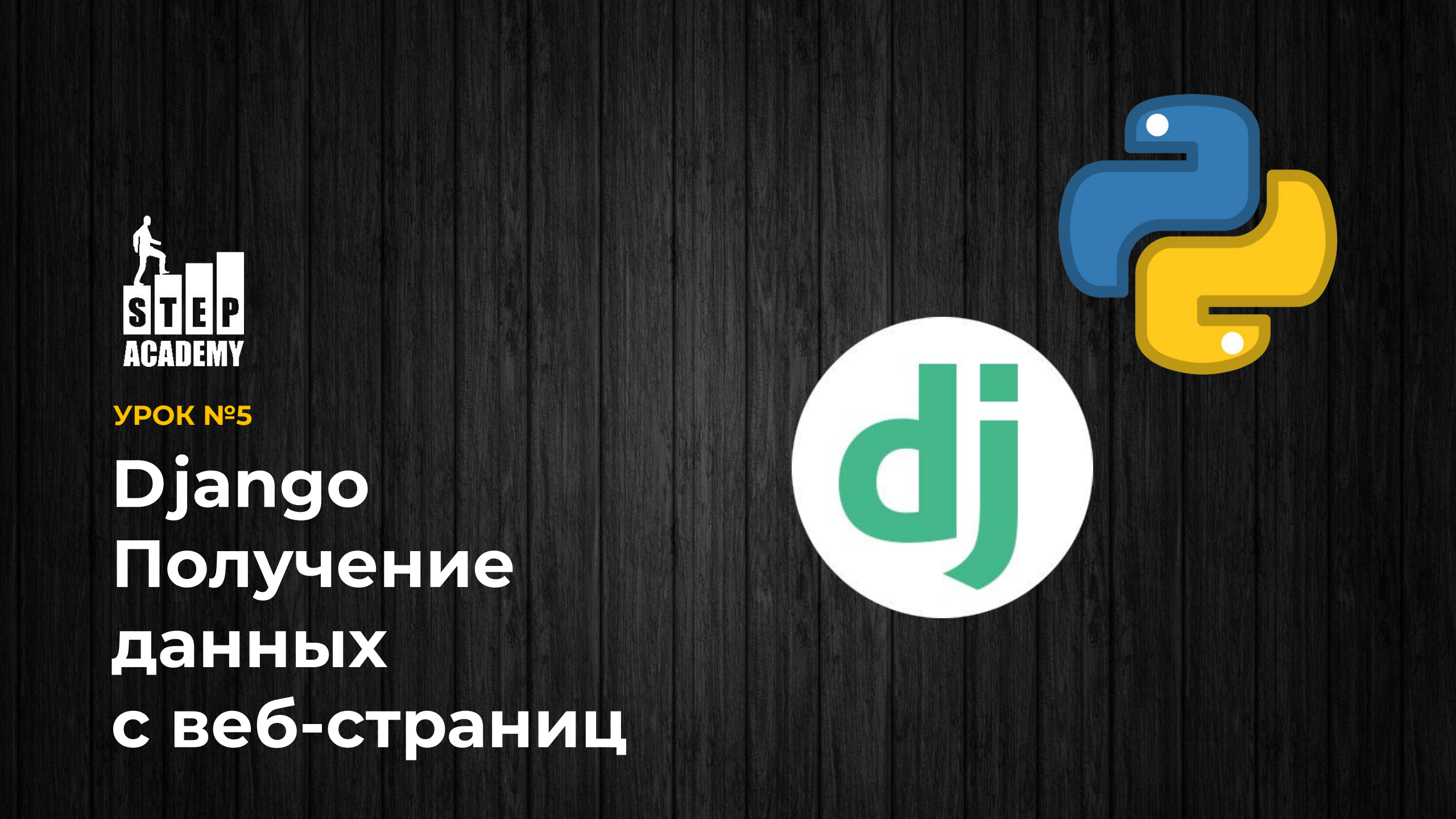


УРОК №5

Django Получение данных с веб-страниц

Ключевые темы

- GET и POST запросы
- Формы, как источник данных
- Bootstrap + Django
- CSRF-token

Понятийные сущности

MVT

Django Template Language

Миграция

Database

admin.py

models.py

QuerySet

Django ORM

Django-приложение

Django ORM

READ	CREATE	UPDATE	DELETE
all	create	?	?
filter		?	?
exclude	new obj		
get	save	update	delete

GET запросы

Ответы пользователя дописываются в URL в формате «**параметр**=**значение**», например «**email**=**name@yandex.ru**».

site.com/form?name=Max&email=name@yandex.ru

Пары «параметр=значение» разделяются знаком **&**. Вариант **method="get"** используется по умолчанию, но у него есть ограничение: URL не должен быть длиннее 3000 символов.

POST запросы

Данные из формы пакуются в тело формы и отправляются на сервер. Они передаются скрыто и не видны в адресной строке.

- POST-запросы более безопасны и подходят для передачи конфиденциальных данных, таких как **логин** и **пароль**;
- Ограничений по объёму пересылаемых данных нет;
- POST-запросы не кэшируются браузером, что может замедлять работу приложения.

GET и POST запросы

GET

- Получение данных без их изменения: если вы хотите получить данные с сервера, но не изменять их. Например, вы хотите получить информацию о продукте на странице товара.
- Кэширование: GET-запросы кэшируются браузером, что ускоряет загрузку страниц.
- Передача данных через URL: GET-запрос позволяет передавать данные через URL.

POST

- Изменение данных на сервере: если вы хотите обновить профиль пользователя.
- Отправка большого объема данных: вы хотите загрузить на сервер файл или отправить большой текстовый документ.
- Передача конфиденциальных данных: вы хотите отправить данные кредитной карты при оформлении заказа в интернет-магазине.

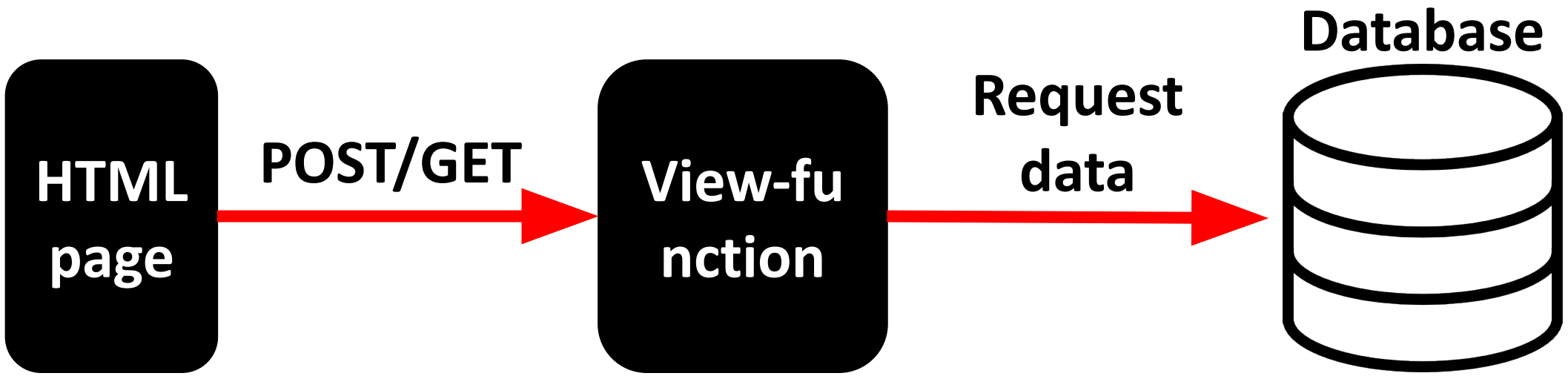
Получение данных

Способы получения данных для **backend** со стороны **frontend**:

- Гиперссылки (<**a**>): гиперссылка с определенным URL-адресом, который будет выполнять GET-запрос при нажатии на нее;
- Формы (<**form**>): данные с полей типа **input**, упакованные в запрос с помощью кнопки;
- Кнопки (<**button**>): Кнопка может быть использована для отправки GET- или POST-запроса с помощью JavaScript или явного указания атрибута formaction и formmethod.

Формы – как источник данных

В качестве примера будет использоваться следующая схема работы Django-приложения:



Создание шаблона с формой

Через **<link>** подключаются стили с **Bootstrap**. Подключение можно выполнить через URL-ссылку, либо через ссылку на папку с заранее скаченными с сайта стилями.

https://getbootstrap.com/docs/5.3/forms/overview/

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstra
    <title>My Django App</title>
</head>
```

Создание шаблона с формой

```
<form action="" method="post">
    {% csrf_token %}
    <div class="mb-3" style="margin-top: 50px;">
        <label for="book_title" class="form-label">Book title</label>
        <input name="book_title" type="text" class="form-control" id="book_title">
    </div>
    <div class="mb-3">
        <label for="book_author" class="form-label">Book author</label>
        <input name="book_author" type="text" class="form-control" id="book_author">
    </div>
     <div class="mb-3">
        <label for="page_count" class="form-label">Page count</label>
        <input name="page_count" type="number" class="form-control" id="page_count">
    </div>
    <button name="add_button" type="submit" class="btn btn-primary">Add book</button>
</form>
```

Комментарии к шаблону

Данный шаблон представляет из себя форму приёма данных о книге: название книги, автор книги и количество страниц в книге.

В шаблоне используются стили с Bootstrap.

Строка **{% csrf_token %}** – служит для защиты от хакерских атак (Cross-Site Request Forgery атака).

Для получения данных с input-полей, им надо прописать атрибут – **name=…**

«Пустая строка» в атрибуте **action** говорит о том, что перехода на другую страницу после отправки формы не будет.

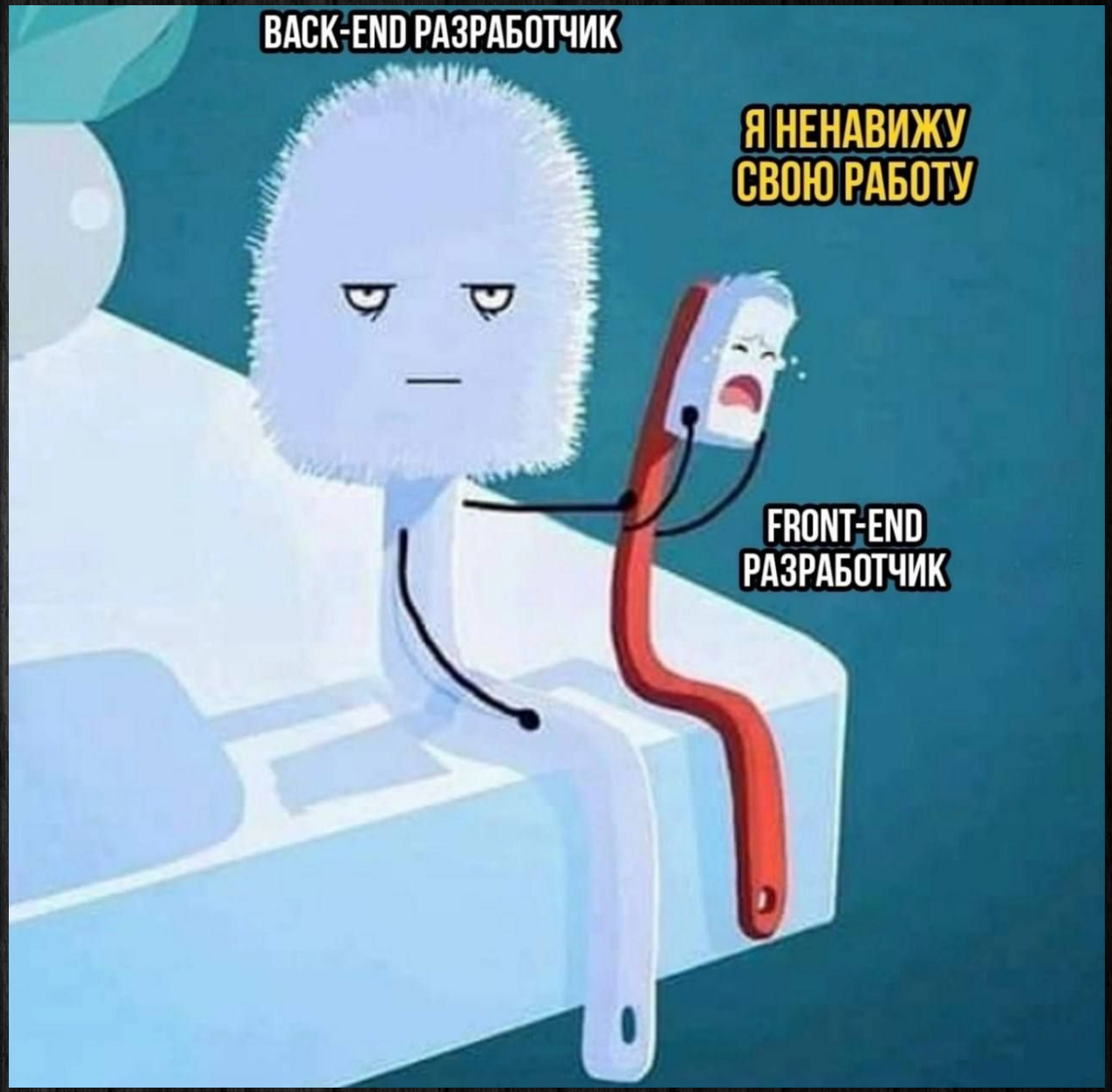
Данные от POST и GET

```
def get_new_book_by_post(request):
    book_title = request.POST.get("book_title")
    book_author = request.POST.get("book_author")
    page_count = request.POST.get("page_count")
```

```
def get_new_book_by_get(request):
    book_title = request.GET.get("book_title")
    book_author = request.GET.get("book_author")
    page_count = request.GET.get("page_count")
```

В скобках указаны названия элементов, которые прописываются в атрибуте **name** у html-тегов **input**.

Конец


BACK-END РАЗРАБОТЧИК
Я НЕНАВИЖУ СВОЮ РАБОТУ
FRONT-END РАЗРАБОТЧИК