

В нашем личном дневнике уже готовы HTML и CSS, а так же функции Flask, которые ты уже умеешь делать. Осталось научиться сохранять карточки с информацией (это как пост в блоге) — так, чтобы они не исчезали каждый раз при перезапуске.

Для этого мы создадим базу данных — это как Google Таблица, но для сайта, и подключим ее к HTML.

Но Flask сам по себе не умеет работать с базами. Нам поможет библиотека flask_sqlalchemy. Она сделает всё за нас! Для этого устанавливай:

```
pipenv install SQLAlchemy
```

Приступим к созданию класса и нужной нам таблицы. Для нашего проекта нам нужно собирать следующие данные:

1. id карточки
2. Заголовок карточки
3. Подзаголовок карточки
4. Текст карточки

Это значит, что в нашей таблице будет 4 строки для хранения информации о каждой карточке

Задача №1. Создание БД

1. Открой файл main.py и создай класс под комментарием "Задание №1":

```
class Card(db.Model):
```

db.Model — это основа, от которой мы создаём таблицу.

2. Создай 4 переменные (столбцы) - id, title, subtitle, text. Вот пример создания столбца:

```
title = db.Column(db.String(100), nullable=False) # Заголовок
```

На нем ты можешь увидеть, что функция `db.Column` создает столбец, `db.String(100)` указывает тип данных строка и ставит ограничение в 100 символов, а `nullable=False` означает, поле обязательно для заполнения..

Вот пара подсказок для других столбцов:

`id` - должен иметь тип данных `Integer`, а настройки `primary_key=True` (это как раз тот самый первичный ключ, который гарантированно сделает каждую карточку (строку) уникальной)

`title` и `subtitle` - должны быть строкой и иметь ограничение на количество символов, поле должно быть обязательным для заполнения.

`text` - должен быть типом данных `Text` и иметь такие же настройки как `title`, но не без ограничения в символах.

3. После создания всех нужных столбцов, нужно прописать функцию для вывода объектов по `id`. Эта функция нужна, чтобы если мы захотим вывести карточку в консоли (например, `print(card)`), она выглядела аккуратно, а не как странный набор символов:

```
def __repr__(self):  
    return f'<Card {self.id}>'
```

4. Отлично, наша БД прописана, теперь давай запустим код для ее создания как файла:

1. Открой терминал и введи `python`, для переключения на язык `python`

2. Теперь импортируй `app` и `db` из `main`: `>>>from main import app, db`

3. После того как в твоём проекте появилась папка `instance` введи в терминал: `>>>app.app_context().push()`. Эта команда говорит Flask, что мы сейчас работаем с нашим сайтом — как будто включаем “режим сайта” для базы данных.

4. И создай файл ДБ: `>>>db.create_all()`

Отлично! Теперь у тебя появилась своя БД, осталось соединить ее с нашим сайтом.

Но прежде чем переходить к следующему шагу, давай посмотрим, что у нас получилось?

Если представить базу данных в виде черного ящика, то **SQLite3 Editor** - это ключ от него. Давай установим это расширение к VS Code?

1. Переходи в VS Code и нажимай в меню Вид -> Расширения.
2. Вводи в поисковой строке SQLite3 Editor.
3. Выбирай его в списке и нажимай кнопку установить.
4. Готово!

Пришло время открыть черный ящик: все там же, в VS Code загляни в папку своего проекта - там есть папка instance, в ней то и лежит база данных, которую ты только что создал - **diary.db**. Ты видишь таблицу и столбцы которые создал в ней.

