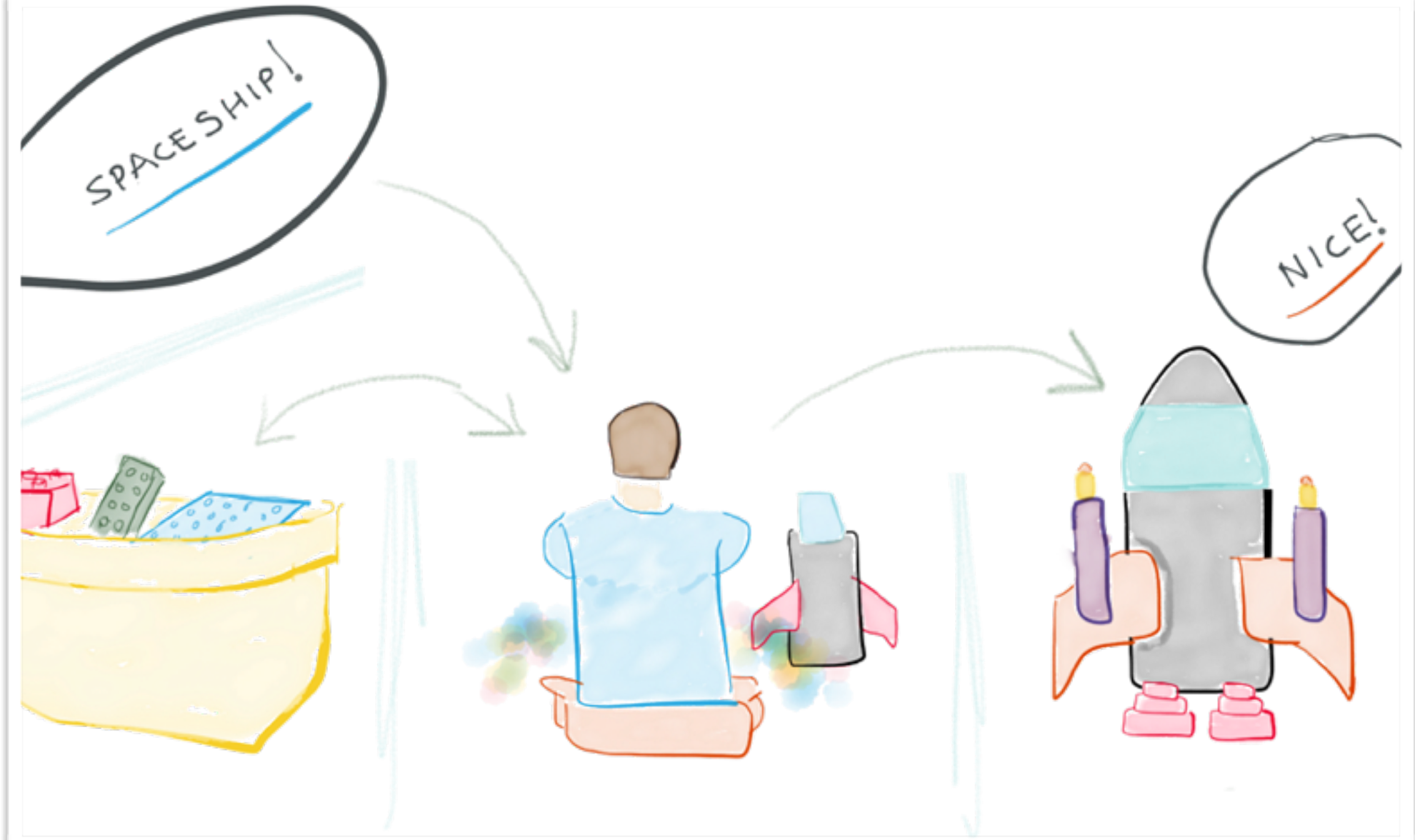
Python

Flask

Micro-Framework

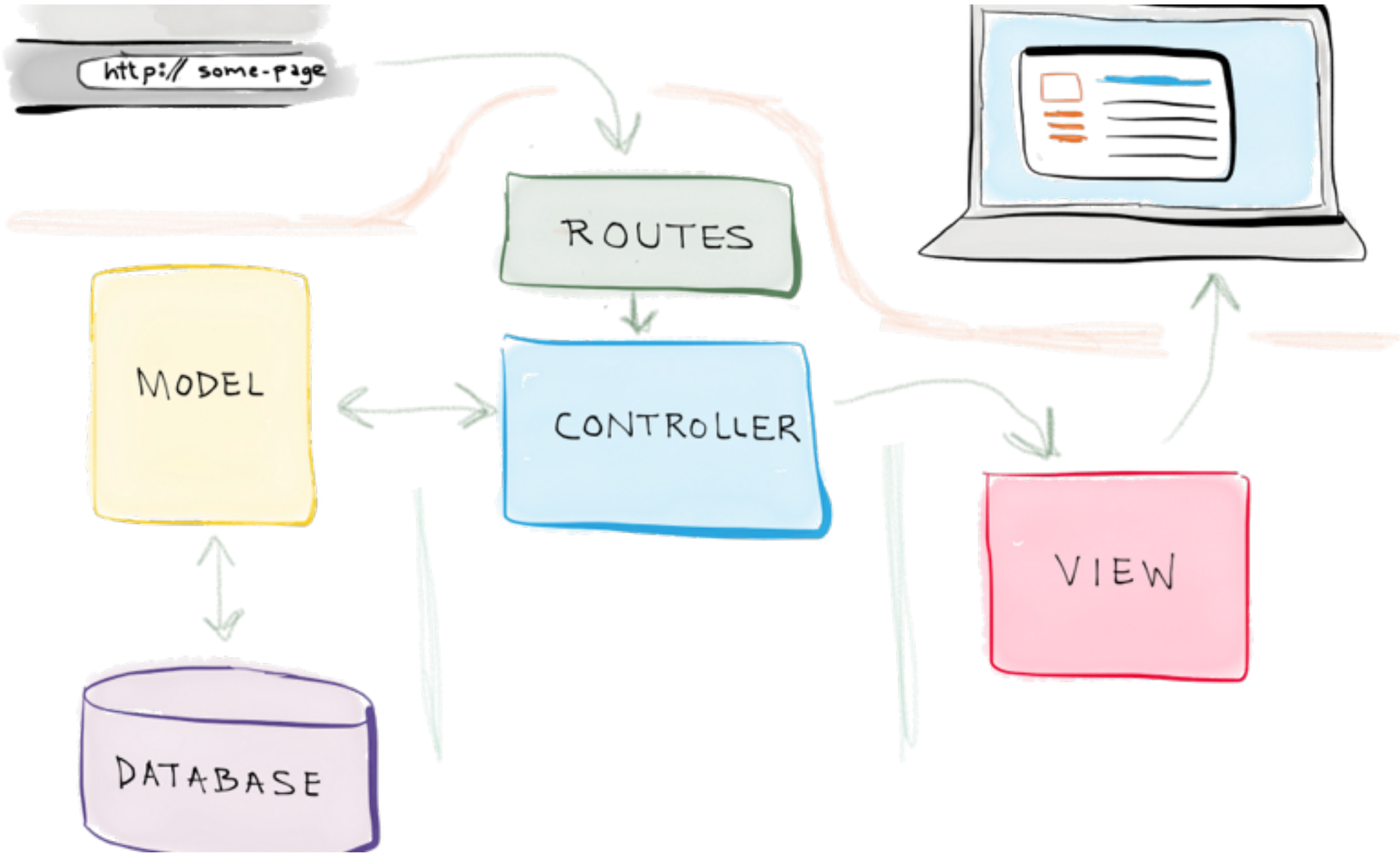
Минимальный рабочий сервер

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return 'hello world!'

if __name__ == '__main__':
    app.run()
```



MVC

Model - View - Controller


http:// some-page
ROUTES
MODEL
CONTROLLER
VIEW
DATABASE

Маршруты (Routes)

- Устанавливает соответствие между URL адресом и функцией вида (View)

Controller

- Получает запрос (Request) от пользователя
- Выбирает, какой маршрут (Route) ему соответствует
- Оперирует выборкой данных
- Вызывает нужную функцию вида (View)

View (Вид)

- Используется для отображения данных (Model)

Model (модель данных)

- Используется для представления данных
- Используется для изменения данных

Почему MVC?

- Представляет собой одну из реализаций парадигмы "разделение ответственности"
- От обратного: если смешать все в кучу - будет сложно вносить изменения
- Может использоваться только в приложениях, где существует UI

Web Server Gateway Interface (WSGI)

- Используется для общения Python-программы и сервера (Nginx, Apache)
- Стандартизирован в https://www.python.org/dev/peps/pep-3333/
- Используется всеми Web-фреймворками для Python

Werkzeug

- Библиотека, которая работает с WSGI

- Лежит в основе Flask

Практика

Файлы "werkzeug/*.py"

Практика

Файлы "flask/*.py"

MVT (Model - View - Template)

- Подвид MVC
- Во многих фреймворках не нужно модифицировать Controller, только поведение приложения в Routes

- Template (шаблон) - чаще всего специально-форматированный HTML файл

Jinja2 (шаблонизатор)

- Позволяет связывать данные Model и View
- Облегчает жизнь разработчику, наличием большого количества утилит
- Расширяем

Практика

Пакет "with_templates"