

빅데이터 분석 프로그래밍

Application

Week-06. Product Serving

Jungwon Seo, 2022-Fall

Product Serving

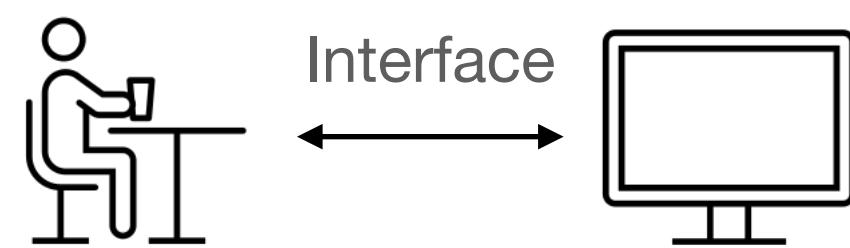
내가 만든 모델을 어떻게 사용자에게 제공 할 수 있을까?

- 주피터 노트북으로?
 - 개발자간에는 공유가 가능하나, 비개발자에게는 무의미한 파일
 - 완성된(훈련이 완료된) 부분만 테스트를 해보고 싶지만, 전처리-훈련 등과 같은 모든 과정을 실행해야함
- API와 GUI를 통해 제공
 - 이미 훈련된 모델이 대기 하면서 입력값이 주어진다면 그에 대응하는 출력값을 내보내도록

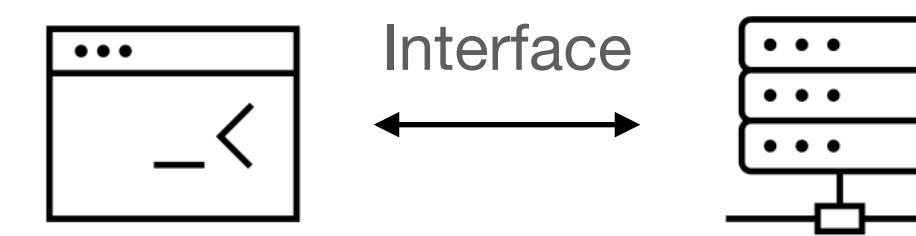
Interface

인터페이스란?

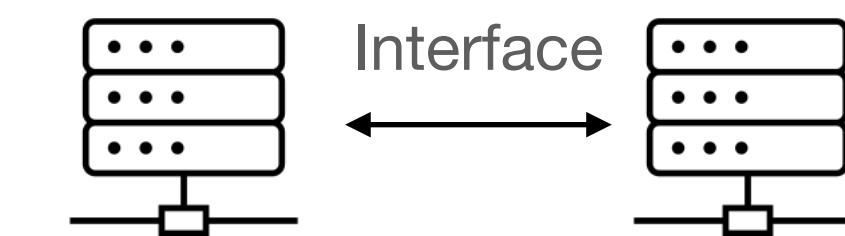
- 인터페이스(interface)는 서로 다른 두 개의 시스템, 장치 사이에서 정보나 신호를 주고받는 경우의 접점이나 경계면이다. 즉, 사용자가 기기를 쉽게 동작시키는데 도움을 주는 시스템을 의미한다.
- 컴퓨팅에서 컴퓨터 시스템끼리 정보를 교환하는 공유 경계이다. 이러한 교환은 소프트웨어, 컴퓨터 하드웨어, 주변기기, 사람 간에 이루어질 수 있으며, 서로 복합적으로 이루어질 수도 있다.
- 터치스크린과 같은 일부 컴퓨터 하드웨어 장치들은 인터페이스를 통해 데이터를 송수신할 수 있으며 마우스나 마이크로폰과 같은 장치들은 오직 시스템에 데이터를 전송만 하는 인터페이스를 제공한다.



사람과 컴퓨터



프로그램과 컴퓨터



컴퓨터와 컴퓨터

Interface

대표적인 Interface의 종류

- **CLI: Command Line Interface**
 - 명령어 기반으로 사용자와 프로그램 / 컴퓨터 간의 소통을 제공해 줄 수 있는 interface
 - GUI가 상용화 되기 전의 컴퓨터들, 서버 전용 컴퓨터
 - 예) gitbash
- **GUI: Graphical User Interface**
 - 시각적인 요소가 제공되어 사용자와 프로그램 간의 소통을 제공해 줄 수 있는 interface
 - 현대 개인용 컴퓨터기반의 프로그램들 (엑셀, 웹사이트 등등)
- **API: Application Programming Interface**
 - 규칙에 의해 미리 구현된 코드/명령들을 사용하여, 프로그램과 컴퓨터간의 소통을 제공해 줄 수 있는 interface



애플2



매킨토시

API란

Application Programming Interface

- 두개 이상의 응용프로그램들이 서로 통신을 할 수 있도록 하는 방법
- 한쪽에서 미리 구현된 기능 또는 정의를 다른 쪽에서 사용할 수 있도록 하는 것
- 요청하는 쪽과 응답하는 쪽
 - 요청자: "내가 이미지 파일을 보내줄 테니, 강아지인지 고양이인지 확인한 결과만 보내줘!"
 - 그 안에서 어떻게 동작하는지는 관심 없어! 결과만 잘 보내줘!
 - 응답자: "오케이! 난 계속 준비하고 있으니까, 입력값이나 제대로 보내줘!"
 - 입력값 이상하면 응답 안하거나, 에러 낼꺼야!
- 함수와는 뭐가 다른가?
 - 함수는 API를 구현하기 위한 한가지 방법!

모델 서빙

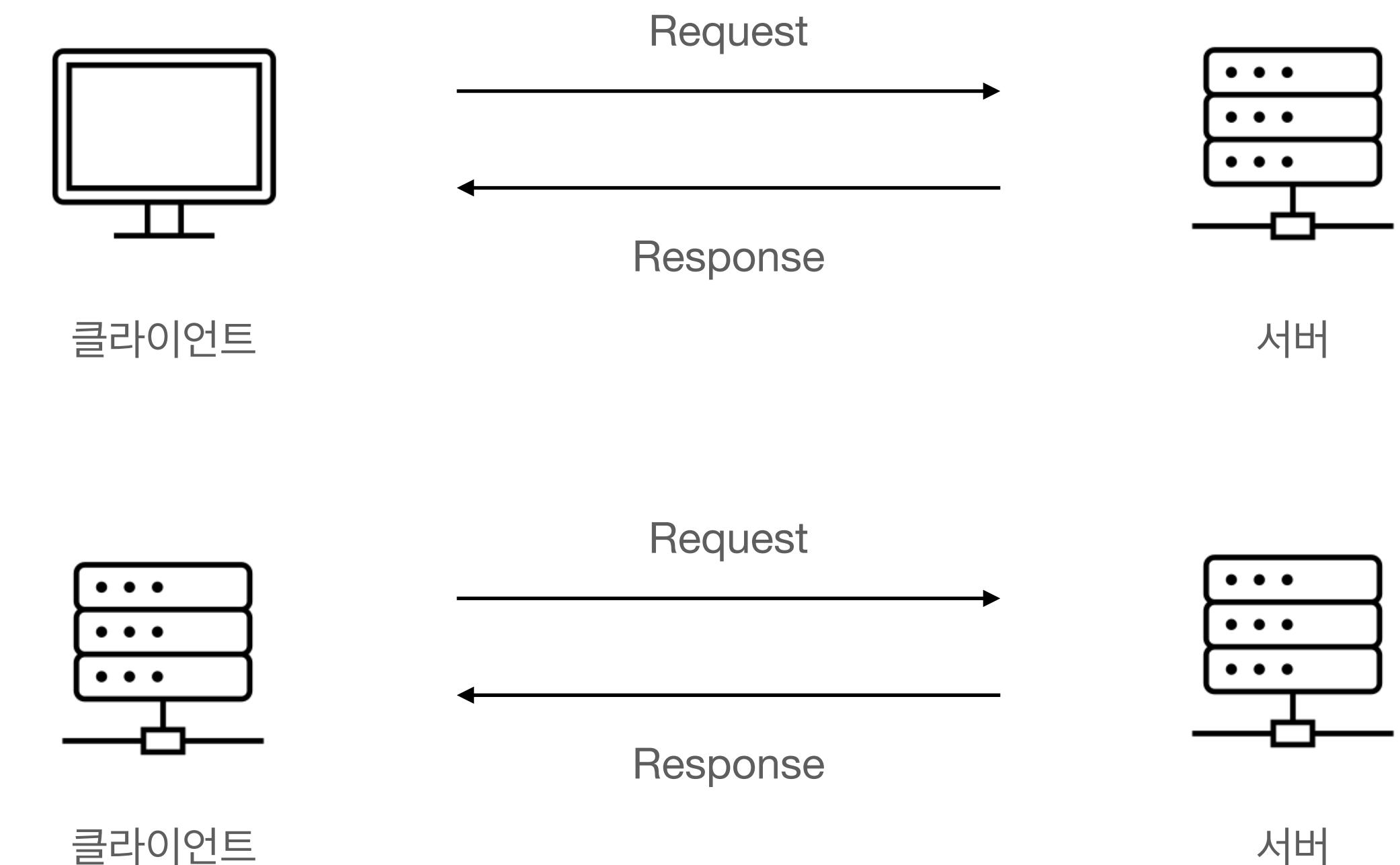
훈련된 모델을 어떻게 제공할까?

- Web API 기반
 - HTTP 요청으로 데이터를 보내면 그에 대한 응답을 반환한다.
 - 예) 강아지 사진을 보내면, 강아지 여부를 응답해준다.
- Web GUI 기반
 - 웹사이트 형태로 사용자가 웹사이트를 이용하면서 원하는 입력과 출력을 할 수 있는 환경을 제공한다.
 - 예) 강아지 사진을 업로드하면, 강아지 여부를 응답해준다.
 - API와의 차이는 Programmatic하게 interaction을 하는게 아니라, 사람이 직접 행동을 해서 interaction을 함

모델 서빙

Web API 기반

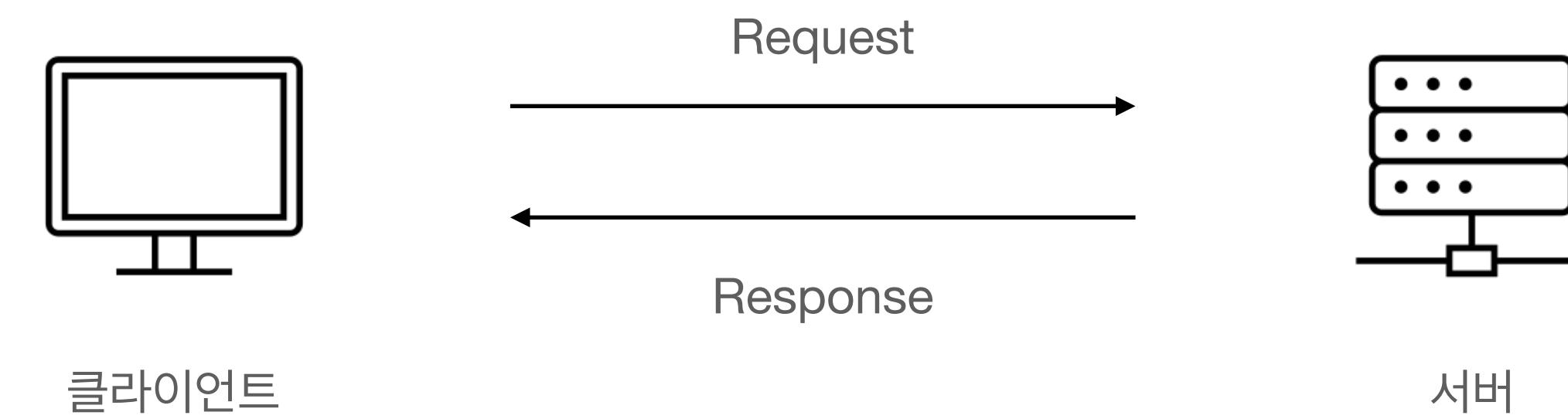
- **클라이언트**: 요청을 보내는 쪽
- **서버**: 요청에 대응하는 응답을 보내는 쪽
- Web API기반의 관점에서는 서버와 클라이언트는 정해져 있다기 보다, 당시의 역할에 따라서 결정
 - 누가 초기 요청을 보내고, 누가 그에 대한 응답을 해주나!



모델 서빙 - 서버

서버를 구성하기 위한 조건

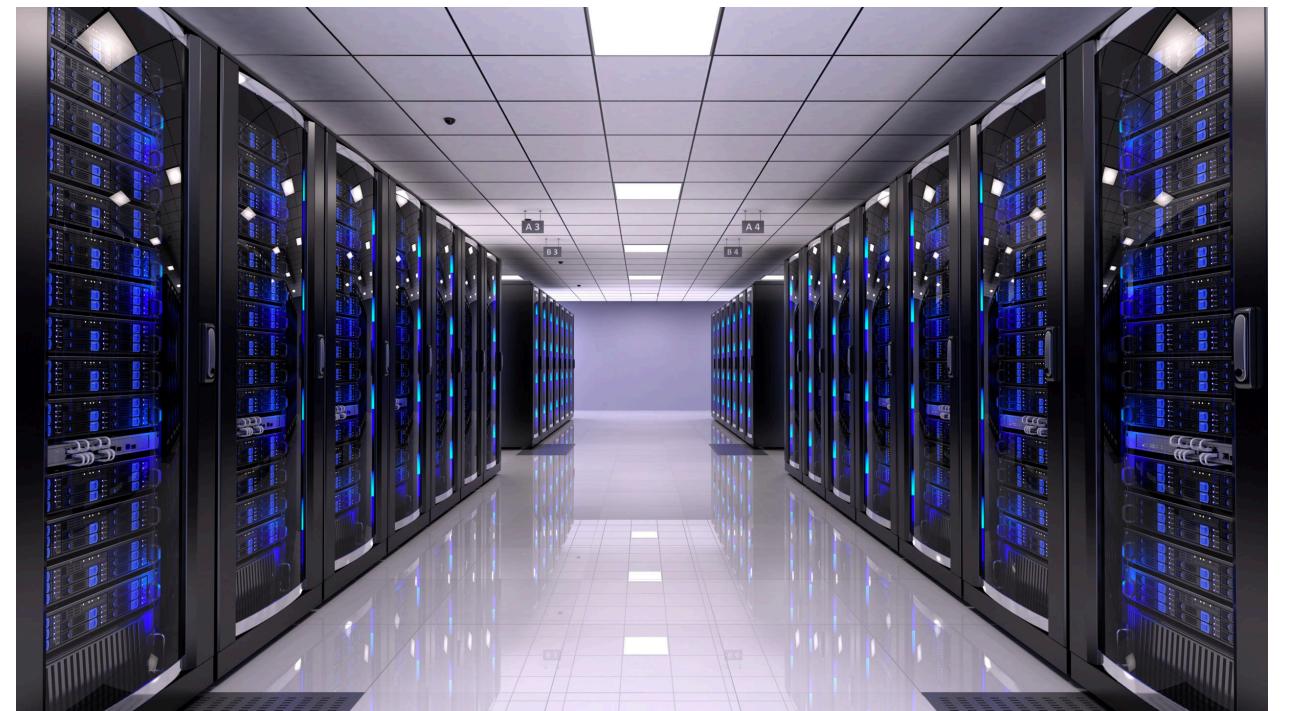
- 항상 요청에 응답할 준비가 되어 있어야 한다.
- 요청에 담겨진 데이터를 읽을 수 있어야 한다
- 요청에 맞는 응답을 보내줄 수 있어야 한다.
- 설정에 따라 특정/불특정 위치에서 접근 가능 해야 한다.



모델 서빙 - 서버

서버를 구성하기 위한 조건

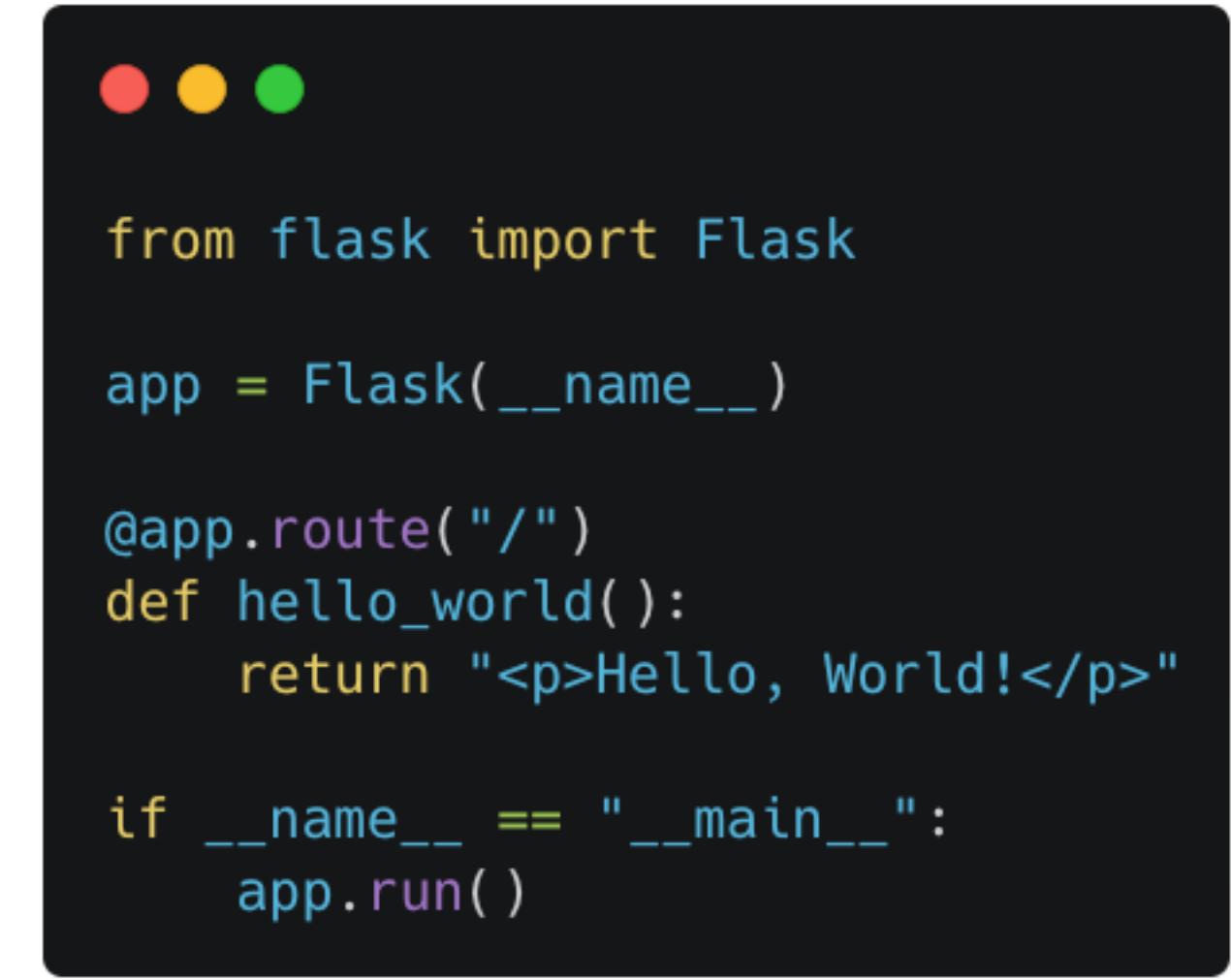
- 항상 요청에 응답할 준비가 되어 있어야 한다.
 - 24시간 돌고 있는 서버(Hardware/Software)
 - On-premise, Cloud
 - 각종 서버 관련 프로그램들
- 요청에 담겨진 데이터를 읽을 수 있어야 한다.
 - API에 의해 사전 정의된 payload 규칙
- 요청에 맞는 응답을 보내줄 수 있어야 한다.
 - 서버 사이드 프로그램에서 요청에 맞는 응답을 제공해 줄 수 있는 코드
 - 예: 강아지의 사진을 판별할 수 있는 모델
- 설정에 따라 특정/불특정 위치에서 접근 가능해야 한다.
 - IP 주소 및 방화벽



서버 Framework

Flask

- 서버 프로그래밍을 위한 Python 기반의 Framework
- Framework와 Library의 차이?
 - 누가 통제권을 갖고 있나!?
 - Library: 내 코드에 남이 짜놓은 코드를 import 해서 쓰는 경우
 - Framework: 남이 짜놓은 틀에 내 코드를 삽입하는 경우
 - 머신러닝에서 Framework의 예: PyTorch, Tensorflow
- HTTP 통신과 같은 요소를 직접 코딩하지 않고, 이미 구현이 되어있는 것들을 사용
 - Flask 외에도, Django, FastAPI 등 Python 기반의 서버 프레임워크 존재
- 단, 웹 개발적인 지식이 많이 필요함



```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

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

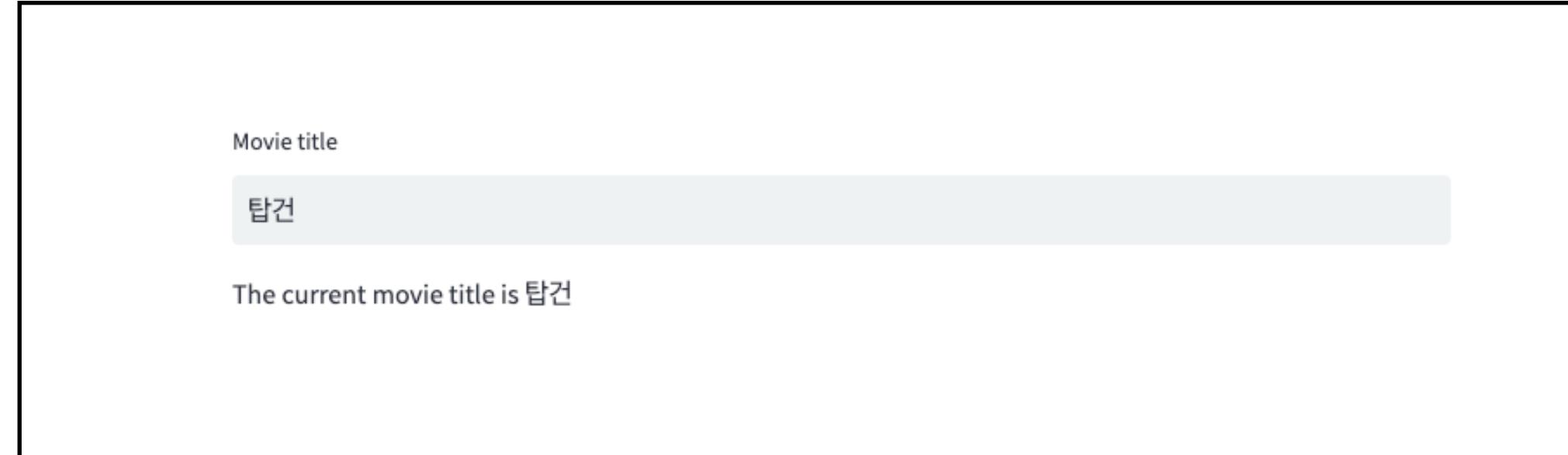
Python 기반의 웹 GUI

Streamlit

- HTML, CSS, Javascript와 같은 코딩 없이 Python만으로 웹사이트를 구축할 수 있는 framework
- 2018년도에 창업, 오픈소스로 진행중
- 웹 개발 경험이 없이도, streamlit에서 제공하는 API들을 통해 다양한 기능들을 개발 가능!



```
import streamlit as st  
  
title = st.text_input('Movie title', '탑건')  
st.write('The current movie title is', title)
```

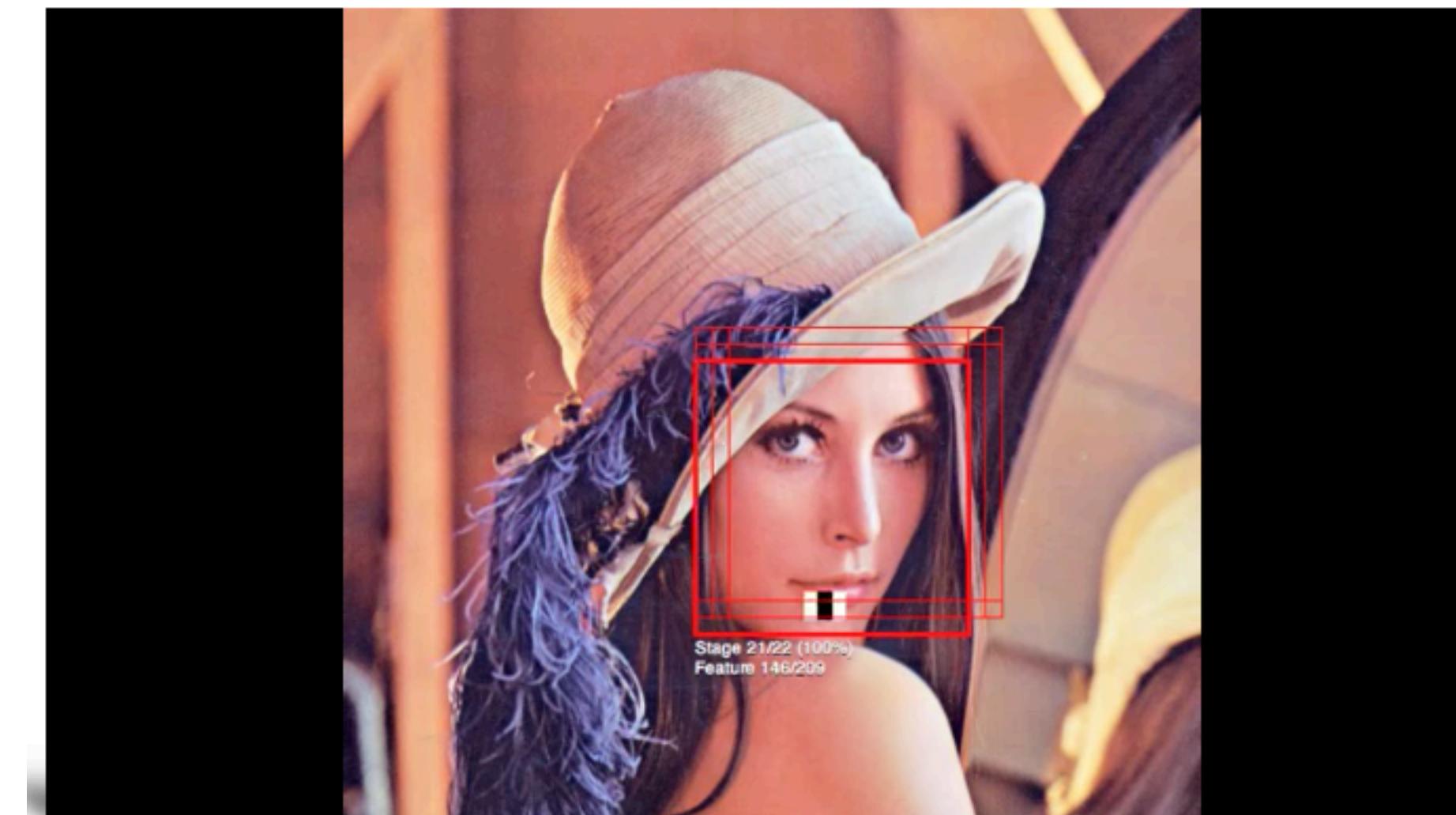


오늘의 최종 결과물

얼굴 인식

Haar feature-based cascade classifiers

- Haar feature-based cascade classifiers를 사용한 객체 감지
 - Paul Viola와 Michael Jones가 2001년 논문 "Rapid Object Detection using a Boosted Cascade of Simple Features"에서 제안한 효과적인 객체 감지 방법



Haar Cascade Visualization
<https://youtu.be/hPCTwxFOqf4>

얼굴 인식

Haar feature-based cascade classifiers

- 미리 트레이닝 된 XML파일을 사용

 haarcascade_eye.xml	some attempts to tune the performance	7 years ago
 haarcascade_eye_tree_eyeglasses.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalcatface.xml	fix files permissions	5 months ago
 haarcascade_frontalcatface_extended.xml	fix files permissions	5 months ago
 haarcascade_frontalface_alt.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalface_alt2.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalface_alt_tree.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalface_default.xml	some attempts to tune the performance	7 years ago
 haarcascade_fullbody.xml	Some mist. typo fixes	3 years ago
 haarcascade_lefteye_2splits.xml	some attempts to tune the performance	7 years ago
 haarcascade_licence_plate_rus_16stages.xml	Added Haar cascade for russian cars licence plate detection, 16 stage...	6 years ago
 haarcascade_lowerbody.xml	Some mist. typo fixes	3 years ago
 haarcascade_profileface.xml	some attempts to tune the performance	7 years ago
 haarcascade_righteye_2splits.xml	some attempts to tune the performance	7 years ago
 haarcascade_russian_plate_number.xml	Create haarcascade_russian_plate_number.xml	6 years ago
 haarcascade_smile.xml	fixing models to resolve XML violation issue	3 years ago
 haarcascade_upperbody.xml	Some mist. typo fixes	3 years ago

E.O.D