

# 웹 어플리케이션

유튜브 주소 : <https://youtu.be/8UX0mhgSfZ8>

웹 어플리케이션 개요 및 핵심 기술

웹 프레임워크, 클라우드 서버

웹 어플리케이션 예시 및 시연

# 웹 어플리케이션 개요

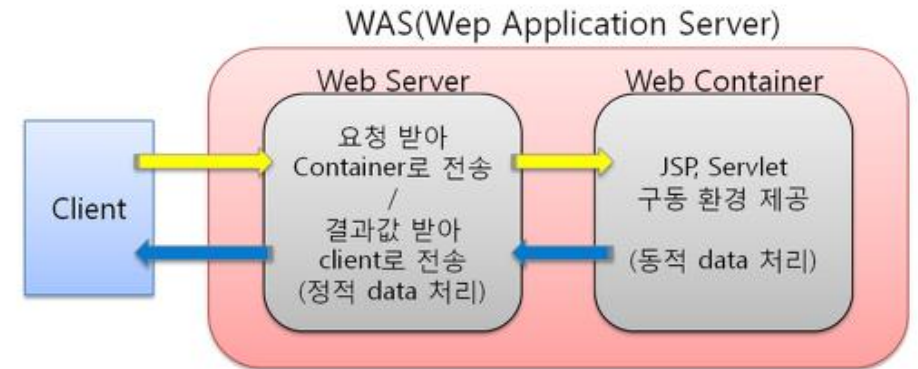
## 웹 어플리케이션 구성 요소

- 웹 브라우저(클라이언트)
  - Request / Response 전송, HTML, CSS 출력
  - E.g. 파이어폭스, 크롬, 마이크로소프트 엣지, 오페라, 웨일 등
- 웹 서버
  - 동적인 정보를 처리해 정적인 정보를 제공
  - E.g. 아파치 톰캣, JBoss, WebLogic, JEUS 등
- 웹 애플리케이션 서버(WAS)
  - 클라이언트로부터 들어온 요청을 처리
  - 응답을 HTML에 담아 브라우저로 전송
- 데이터베이스
  - 데이터 구조 정의, 데이터 CRUD, 대부분 WAS와 직접 통신
  - E.g. 오라클, MySql, MongoDB, Redis, MariaDB 등

# 웹 어플리케이션 개요

## 웹 서버와 WAS의 차이점

- 웹 서버 – 네트워크망에 종속X
  - 작성된 html페이지 등을 웹서비스 해주는 역할
  - 이미지, 단순 html파일 등 정적인 데이터를 처리
- WAS – 웹 서버 + 웹 컨테이너
  - Asp, php, jsp 등 개발 언어를 읽고 처리
  - 프로그램 실행 환경과 DB 접속 기능 제공
  - 동적인 데이터를 처리하는 서버
  - 웹 서비스 플랫폼으로서의 역할



# 웹 어플리케이션 핵심 기술

## **API** - Application Programming Interface

- 응용 프로그램에서 기능을 사용하거나 데이터를 주고 받기 위한 기능
- API 구성 요소
  - 요청정보(요청 URL, 요청 방식(GET/POST...))
  - 서버가 제공할 기능(회원 데이터 조회, 게시판 생성 등)
  - 응답데이터(어떤 key로 데이터를 줄지?)
    - E.g. `return render_template('main.html', data=results)`
- E.g. 은행에서 돈을 인출한다고 할 때
  - 고객 - 클라이언트, 은행 - 서버, 창구 - API
  - 고객은 정해진 창구를 통해 은행에 인출을 요청함

# 웹 어플리케이션 핵심 기술

## 클라이언트 - 서버 간 통신 방식

### • GET 방식

- 통상적으로 데이터 조회(Read)를 요청할 때 사용
- 데이터 전달 형식 - URL 뒤에 물음표를 붙여 key=value로 전달
  - E.g. google.com?pass=한성대
- 데이터를 헤더에 포함하여 전송 - URL에 데이터가 노출되어 보안에 취약

### • POST 방식

- 통상적으로 데이터 생성, 변경, 삭제 요청할 때 사용
  - E.g. 회원가입, 회원탈퇴, 비밀번호 수정
- 데이터 전달 형식 - HTML body(보이지 않음)에 key:value 형태로 담아서 전달
- URL에 변수(데이터)를 노출하지 않고 요청
- GET 방식보단 기본 보안이 갖춰져있음

# 웹 어플리케이션 핵심 기술

## Ajax – Asynchronous Javascript And Xml(비동기식 JS와 xml)

- JS 라이브러리 중 하나임
- JS를 이용해 클라이언트-서버간 XML 데이터(혹은 JSON)를 주고 받는 기술
- Ajax로 주로 하는 작업
  - 웹 페이지 새로고침 없이 서버에 요청(Request) -> 비동기 서버 통신 방식
    - 웹 페이지의 속도 향상
  - 서버로부터 데이터를 받고 작업을 수행
- импорт – JQuery 이용
  - JQuery – JS를 편하게 쓸 수 있게 해주는 일종의 라이브러리
  - `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>`

# 웹 어플리케이션 핵심 기술

## • GET 방식 API 예시

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <!--ajax 임포트-->

  <link rel="stylesheet" href="static/bgColor.css">

  <title>title</title>
```

```
from flask import Flask, render_template, jsonify, request
```

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

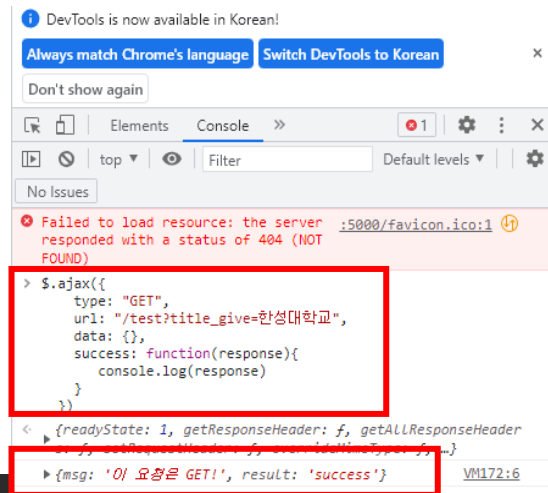
```
# HTML을 주는 부분
```

```
@app.route('/')
def home():
    return render_template('index.html')
```

```
# API 역할을 할 부분
```

```
@app.route('/test', methods=['GET'])
def test_get():
    title_receive = request.args.get('title_give')
    print(title_receive)
    return jsonify({'result': 'success', 'msg': '이 요청은 GET!'})
```

```
if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

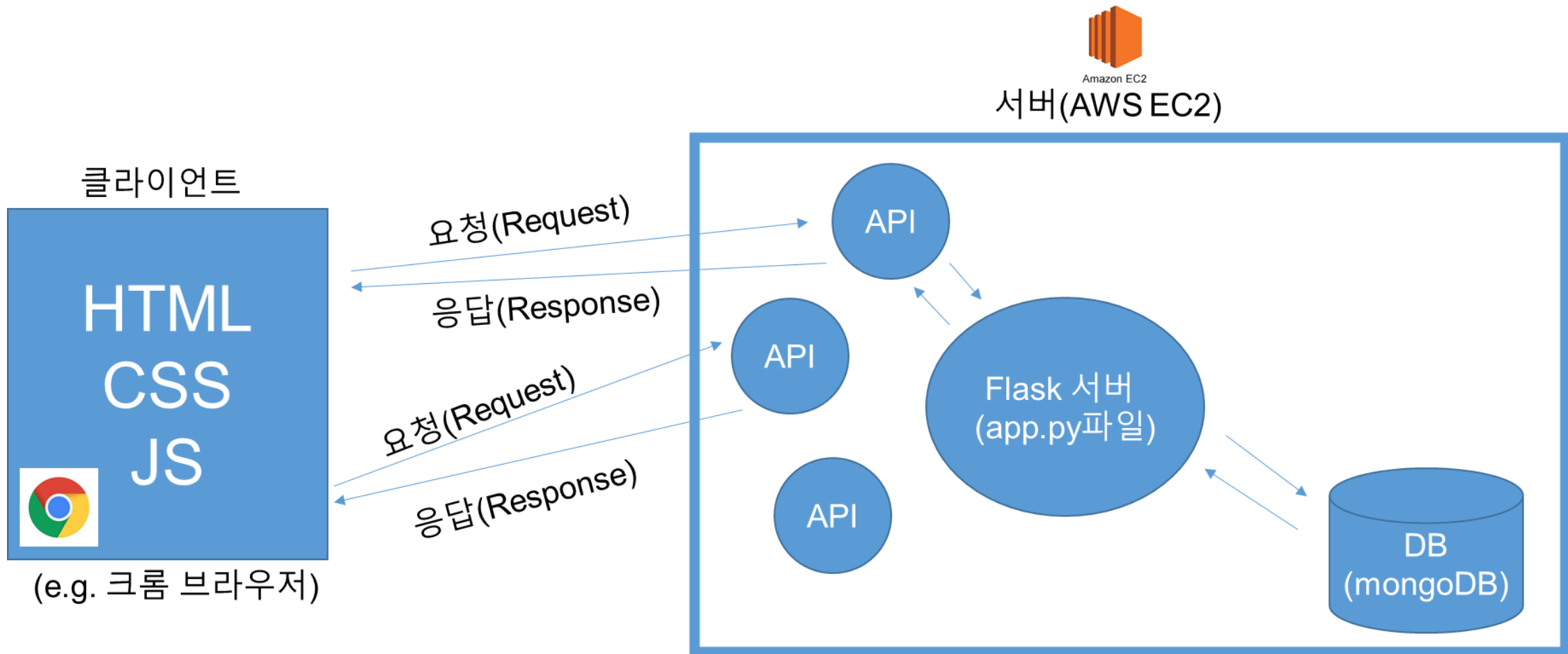


```
* Running on all addresses (0.0.0.0)
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:5000
* Running on http://121.129.220.159:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 578-193-160
127.0.0.1 - - [06/Jun/2022 08:18:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2022 08:18:46] "GET /static/bgColor.css HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2022 08:18:46] "GET /static/sangsangbugi.PNG HTTP/1.1" 200 -
127.0.0.1 - - [06/Jun/2022 08:18:46] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Jun/2022 08:18:48] "GET /static/bgColor.css HTTP/1.1" 304 -
127.0.0.1 - - [06/Jun/2022 08:19:56] "GET /test?title_give=한성대학교 HTTP/1.1" 200 -
한성대학교
```



# 웹 어플리케이션 개요

- 웹 어플리케이션 구성도



# 웹 프레임워크

- 웹 프레임워크

- 어플리케이션 개발에 필요한 스크립트를 정리한 스크립트 모음
- 개발자가 반복적으로 해야하는 공통부분을 최소화 할 수 있도록 설계

- 장점

- 체계적인 코드 관리로 유지보수가 용이
- 기본 설계 및 기능 라이브러리를 제공해 개발 생산성 향상
- 코드의 재사용성이 높음
- 추상화된 코드 제공을 통해 확장성이 좋음

- 단점

- 학습 난이도가 높음
- 기본적으로 설계된 구조때문에 자유로운 개발에 한계가 있음
- 사용하지 않는 기능에 대한 라이브러리가 포함될 수 있음
- 제공되는 기능만큼 프로젝트의 용량이 증가함

# 웹 프레임워크

- 프레임워크 종류

구분	종류
자바 프레임워크	Spring, Struts, 전자정부 프레임워크
JS 프레임워크	AngularJS, React, Polymer, Ember
프론트엔드 프레임워크	Bootstrap, Foundation, MDL
파이썬 프레임워크	Flask, Django, Pyramid, CherryPy

# 클라우드 웹 서비스

- 웹 어플리케이션을 공개 IP로 접근할 수 있게 하기 위해 필요
  - 서버가 요청에 언제나 응답할 수 있게 항상 켜져있어야 함
  - 웹 서비스가 항상 실행되어 있어야 함
- EC2 – AWS에서 제공하는 클라우드 컴퓨팅 서비스
  - 아마존이 구축한 데이터 센터의 서버용 컴퓨터들의 자원을 임대 가능
  - AWS가 제공하는 URL(public DNS)를 통해 이 컴퓨터에 접근할 수 있음
- 장점
  - 용량을 늘리거나 줄일 수 있음(탄력성)
  - 사용한 만큼 지불하므로 합리적인 가격에 이용 가능
  - 사용자가 인스턴스를 완전히 제어할 수 있음
  - 보안 및 네트워크 구성, 스토리지 관리가 효과적임

Q & A