

# Lab02: Introduction to Python web framework

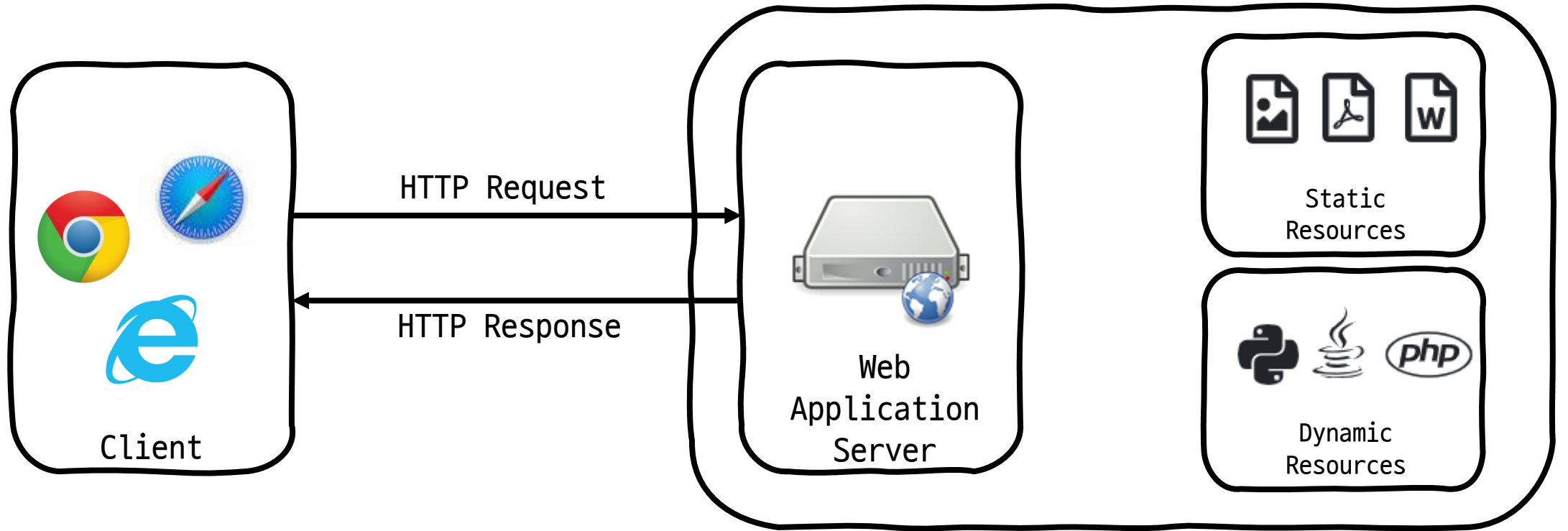
ESM2014-41

객체지향프로그래밍 및 실습

SKKU 시스템경영공학과

조영일

# What is Web (Application) Server?



# What is Web (Application) Server?

- 클라이언트의 HTTP Request를 해석해서, 해당 Request에서 요구하는 Resource를 찾거나 생성하여 다시 클라이언트에게 HTTP Response로 전송하는 일련의 소프트웨어 집합
- 일반적으로 웹 서버라고 하면 서버 역할 수행을 위한 하드웨어와 소프트웨어 모두를 의미하지만, 사실상 거의 모든 컴퓨터가 웹 서버 하드웨어 역할을 할 수 있으므로 우리는 웹 서버 소프트웨어에 대해서만 다룬다.
- 웹 서버가 클라이언트에게 제공하는 Resource는 Resource의 생성 시점에 따라 Static(정적) Resource와 Dynamic(동적) Resource로 나뉘어진다.

# What is Web (Application) Server?

- 웹 서버가 클라이언트에게 제공하는 Resource는 Resource의 생성 시점에 따라 Static(정적) Resource와 Dynamic(동적) Resource로 나뉘어진다.
- Static Resource : 클라이언트 혹은 요청에 따라 변하지 않는 리소스. 서버에 미리 저장된 파일(Image, CSS, JavaScript, HTML 등). 서버에 저장된 파일 자체가 변경되지 않는 한 항상 같은 리소스가 전달 된다.
- Dynamic Resource : 클라이언트 혹은 요청에 따라 변하는 리소스. 서버에 미리 저장되어 있지 않으며, 요청과 동시에 Server-Side Script(Server Application/Program)에 의해 적절한 응답을 생성하여 클라이언트로 전송한다.

# What is Web (Application) Server?



- 쇼핑몰 'MyPage' 예시

- Static Resource : 쇼핑몰 전체 디자인 레이아웃을 구성하는 파일(css), 쇼핑몰 로고, 상품 이미지 등 상황에 따라 변하지 않는 것들

- Dynamic Resource : 상황, 사용자, 시간에 따라 동적으로 달라져야만 하는 콘텐츠 들 (개인별 주문내역 등)

<https://www.shopintake.com/>

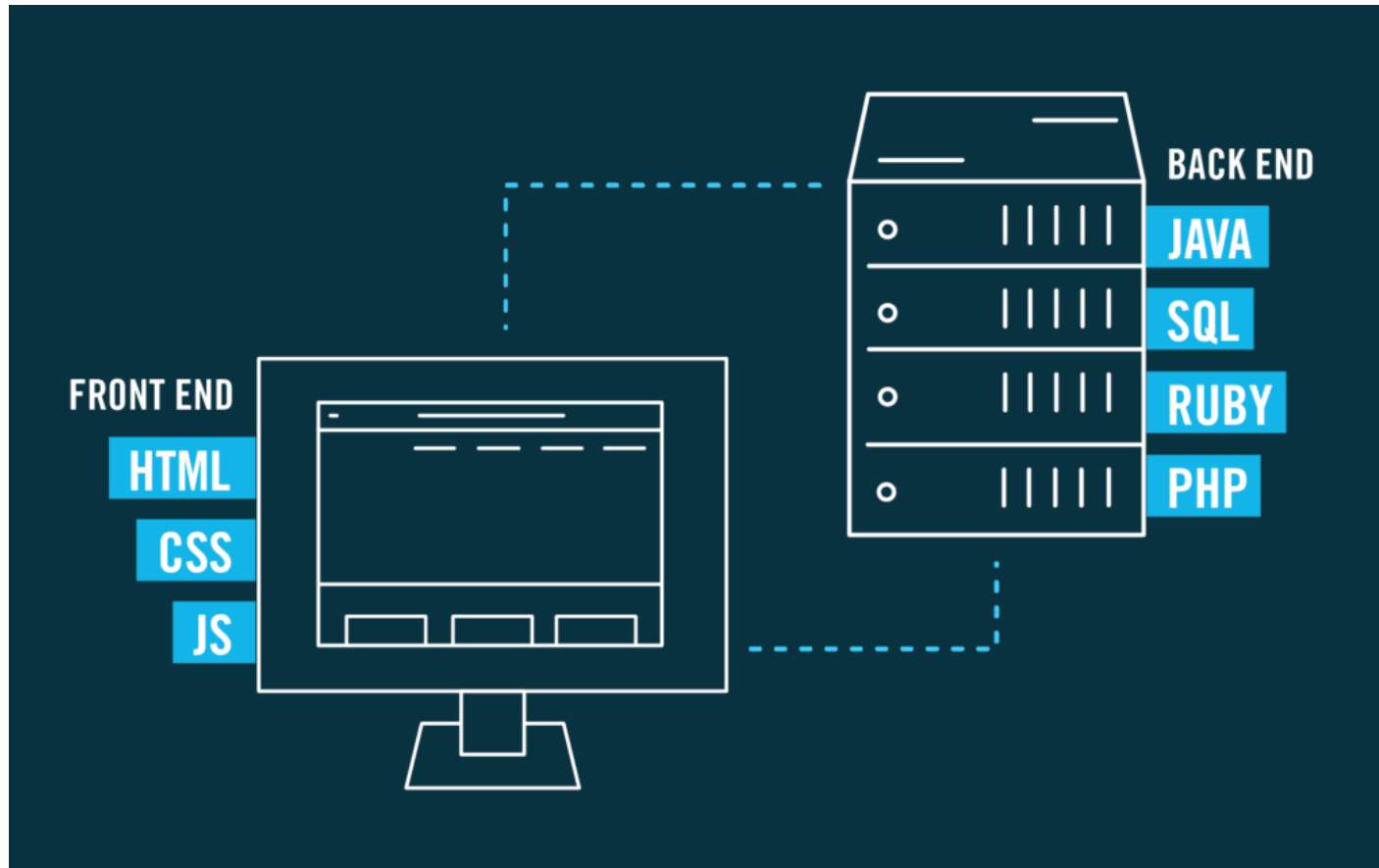
# Server-Side Languages

- 요청에 따른 Dynamic Resource 를 만들고 적절한 응답을 생성하여 반환하는 것이 결국 웹 서버에서 가장 중요한 역할
- 컴퓨터 프로그램의 정의 :  $\text{Input Data(Request)} + \text{Logic(Web Server)} = \text{Output Data(Response)}$
- Web Server는 어떤 프로그래밍 언어로 만들까?
  - 대부분의 범용적 프로그래밍 언어는 대부분 Web Server 구현에 사용 가능 : C/C++, Java, Javascript, Python
  - 웹 서비스 구축만을 주된 목적으로 하는 언어 : PHP

# Back-end vs Front-end

## Front-End:

Back-End에서 생성된 결과물이 클라이언트(웹 브라우저)에 시각적으로 보여지는 방식(Presentation)에 관한 내용을 다룬다.



## Back-End:

Dynamic Resource의 생성과 관계된 데이터 가공, 처리, 수정 로직을 다룬다.

# Front-end

- HTML : Hyper Text Mark-up Language의 약자. 하이퍼텍스트(HyperText)로 이루어진 웹 페이지 모습을 기술하기 위한 규약. 프로그래밍 언어가 아니라 마크업 언어이다.
- CSS : Cascading Style Sheet의 약자. HTML로 작성된 문서가 실제로 웹사이트에서 어떻게 보여질지를 결정하는 언어. 주로 웹페이지의 디자인과 연관되어 있다. 역시 프로그래밍 언어가 아니다.
- JavaScript : 웹 브라우저 내에서 동작하는 Interpreter Language. 웹 페이지 내에서 사용자와 상호작용하는 UI를 구성하는 등 웹 페이지 내의 여러 요소들의 동작을 담당한다.

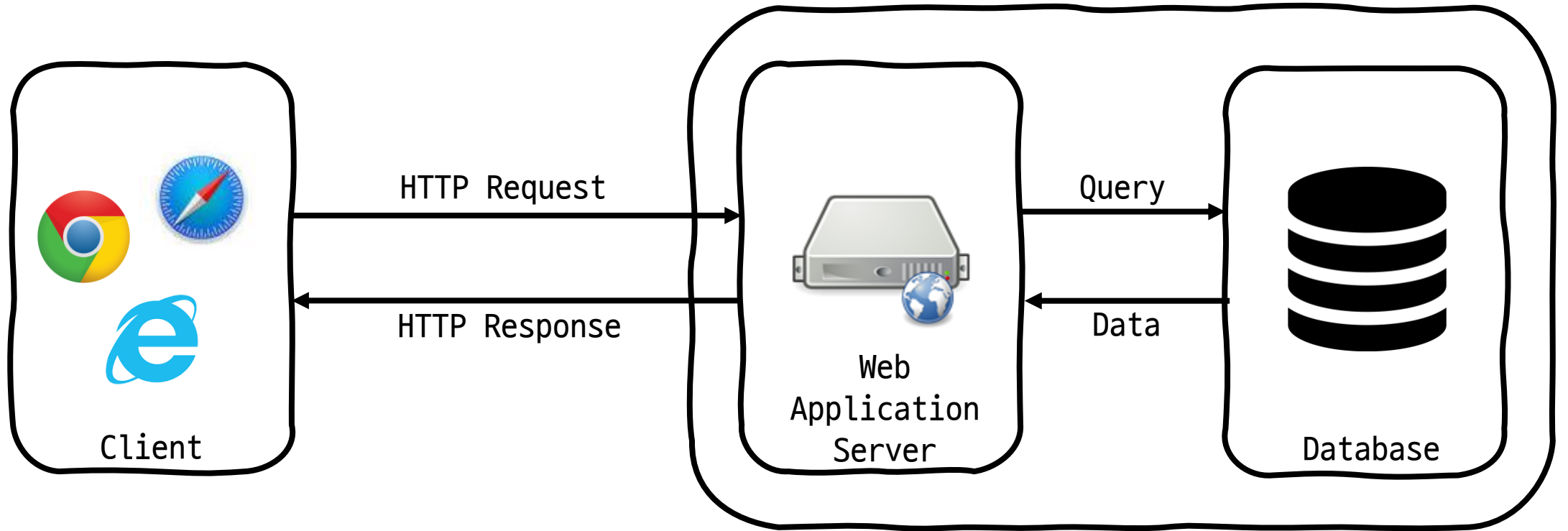


# Front-end : HTML

```
<!-- 모든 HTML 문서는 <html> 로 시작해서 </html>로 끝난다 -->
<html>
<head>
  <!-- 문서의 정보를 나타내기 위한 영역 head -->
</head>
<body>
  <!-- 문서의 본문을 나타내기 위한 영역 body -->
  <p>Hello World!</p>
  <p>It works!</p>
</body>
</html>
```

- 모든 HTML 문서는 <html>로 시작해서 </html>로 끝나야함
- 문서의 요소를 나타내기 위한 조각 “Tag”
- 나타내고자 하는 요소(문단, 표, 머리말, 이미지, 리스트 등)에 따라 이미 지정된 HTML 태그가 존재한다.
- HTML Examples : [https://www.w3schools.com/html/html\\_examples.asp](https://www.w3schools.com/html/html_examples.asp) (숙지 필요)

# Web Service Architecture



# Database



- 웹 서비스에서 가장 중요한 것은 Data!
- 일반적인 프로그래밍 언어의 경우 대량의 데이터를 영속적이고, 효율적으로 보관하는 것에 한계가 있음

# Database

- DBMS(Database Management System) : Software system that enables users to define, create, maintain and control access to the database.
- SQL(Structured Query Language) : DBMS상에서 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어
- 사용되는 대부분의 DBMS는 관계형 데이터베이스(RDB) 형태로 데이터들을 컬럼(Column)과 열(Row)로 이루어진 테이블로 저장하고 테이블간 관계(Relation)로서 데이터 사이의 연결을 표현한다.
- DBMS의 종류 : MySQL, MariaDB, MSSQL, Aurora, Oracle 등.

# Web Framework

- HTTP Protocol, Front-end(HTML, Javascript, CSS), DBMS까지.. 이거 다 직접 구현하면 웹 서비스는 도대체 언제 만들 수 있나요?
- 웹 프레임워크 : 웹 서비스를 만들기 위해 필요한 공통적인 요소들을 미리 구현하여 Package/Library 형태로 제공하는 것
- 프로그래머는 HTTP Protocol 등 상대적으로 낮은 레벨의 구현에 대해 세세하게 신경 쓰지 않고 웹 서비스 설계 자체에만 집중 할 수 있도록 해준다.
- 웹 서비스를 개발하기 위해 빈번하게 반복적으로 만들어야만 하는 기능들(회원가입, 로그인, 로그아웃 등)을 미리 제공함으로써 웹 서비스의 개발 속도를 높인다.

# Web Framework

- 대부분의 프로그래밍 언어에서는 이러한 웹 프레임워크가 오픈소스로 제공되고 있음
- Java : Spring
- Python : Django, Flask
- Ruby : Ruby on Rails
- JavaScript : Express

# Django

- 한글 발음 “장고” => 어디가서 “드장고”라고 발음하지 마세요.
- 신문사의 인턴 웹 프로그래머였던 에이드리안 홀로바티(Adrian Holovaty)와 사이먼 윌리슨(Simon Willison)이 공동개발
- 2005년 최초 오픈소스 라이선스로 공개
- 현재는 DSF(Django Software Foundation)에 의해 운영되고 있으며, 전 세계 많은 개발자들이 Django 개발에 참여하고 있음
- Python에서 가장 유명한 웹 프레임워크이며, 상용 서비스 개발에도 많이 사용됨.

# Dj ango



The web framework for perfectionists with deadlines.

## Core concepts of Django

- Ridiculously fast : 낮은 러닝 커브와 빠른 개발 속도
- Fully loaded : 일반적인 웹 개발에 필요한 대부분의 기능  
기본 탑재(Batteries Included)
- Incredibly versatile : 다양한 용도를 위해 사용 가능



# Django in Production



# Features of Django

- ORM(Object Relational Mapping) : Django는 데이터베이스를 다룰 때 별도의 SQL을 알지 못해도, 데이터베이스 상의 테이블과 관계를 모두 클래스와 객체(Object)로 다룰 수 있도록 하는 ORM을 기본 채택하고 있다.
- MVT(Model-View-Template) : 데이터 구조를 저장하는 Model, 데이터를 가져오고 수정하는 View, 사용자에게 보여질 화면을 구성하는 Template을 분리하여 구현 할 수 있도록 하여 체계적인 웹서비스 개발을 가능하도록 한다.
- Batteries-included : 각종 HTTP 요청/응답 처리, 사용자 인증, 관리자 페이지, Form 등 웹 서비스 구현에 필요한 대부분의 기능을 이미 포함하고 있음

# Study Topic

- Code Academy HTML : <https://www.codecademy.com/learn/learn-html> ★★