

Web Programming 개요

송실대학교 AI융합학부
School of AI Convergence

Contents

1. 웹의 개념
2. 웹의 활용 분야
3. 웹 표준과 HTML5 ☆
4. 웹 브라우저와 HTML 편집기
5. 웹 서버의 개념
6. Web Application Architecture
7. Full Stack JavaScript Web Development
8. Front-end Web Application 개발 Framework
9. 강의 내용

웹의 개념 - 인터넷

■ 인터넷

- 우리가 사용하는 컴퓨터를 전 세계 규모로 연결한 것
- TCP/IP protocol 사용



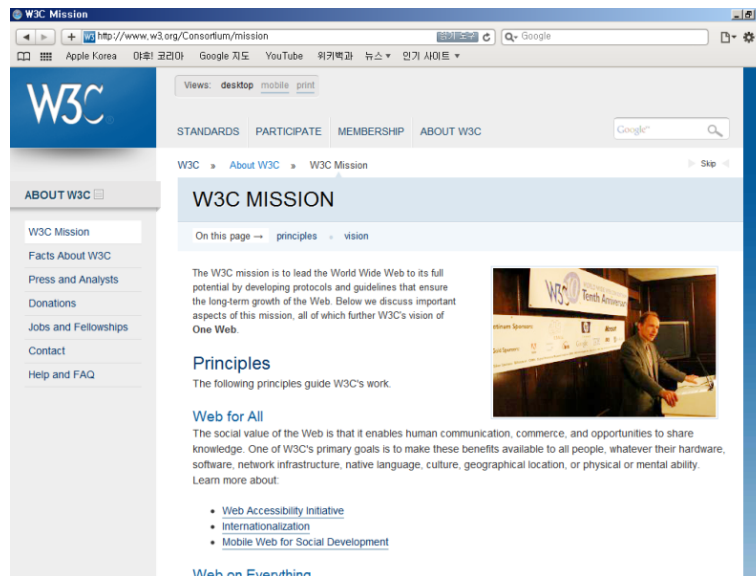
웹의 개념 - 등장

- Web (WWW: World Wide Web)
 - 인터넷 표준 문서 시스템
 - WWW가 나오기 전에는 ftp, email등으로 문서 공유
 - 인터넷 상에서 온라인으로 문서를 공유하는 것이 목적



웹의 개념 - 등장

- WWW 창시자 : 팀 버너스 리(Tim Berners-Lee)
 - 영국 옥스퍼드 대학 졸업
 - 1989년도에 인터넷 기반의 hyper media의 초기 모델을 설계
 - 1990년에 HTTP, HTML의 스펙을 제안한 것이 현재의 웹 기술로 널리 퍼짐
 - 현재는 교수로 W3C의 디렉터 역할



웹의 개념 - 발전

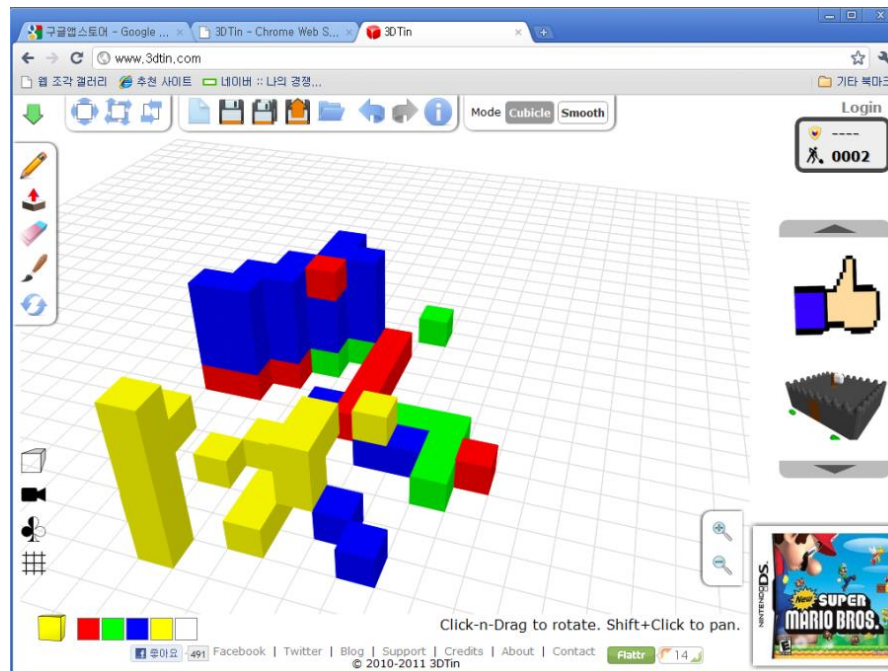
- Multimedia 문서 공유 *WWW 문서: ms word text 한 문서*
 - 최초 HTML로 작성된 문서 연결 목적이었으나, Text, graphic, audio, video, program file 등 multimedia 문서 제공
- 반응형 웹 (Responsive Web) *썩썩*
 - 웹 서버에서 제공되는 정보가 Desktop, Notepad, 모바일 폰 등 다양한 기기에 맞춰 제공되는 웹 환경



웹의 개념 - 발전

■ HTML5

- HTML5 = HTML + CSS + 자바스크립트 API
- HTML5 = 최신 웹 개발을 위한 다음 세대 웹표준

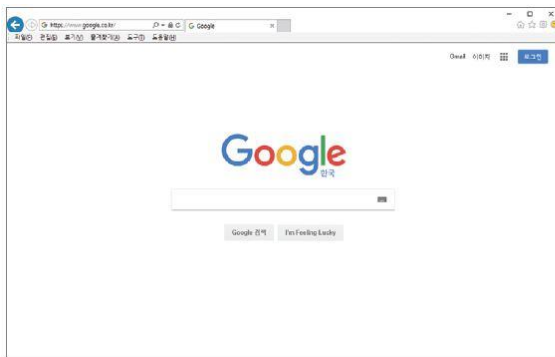


크롬 브라우저에서 게임 애플 실행한 화면

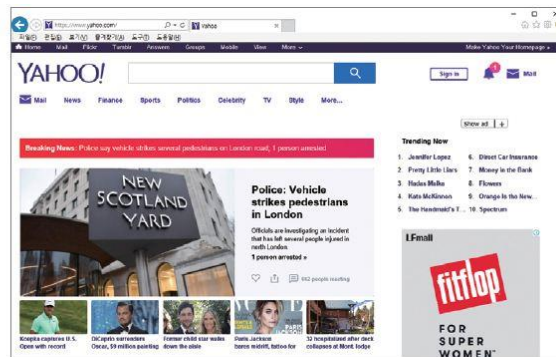
웹의 활용 분야 - 정보 검색

■ 정보 검색

- 구글(Google), 야후(Yahoo), 다음(Daum), 네이버(Naver) 등



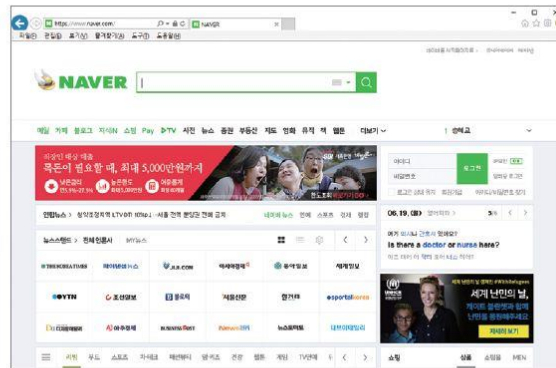
구글



야후



다음



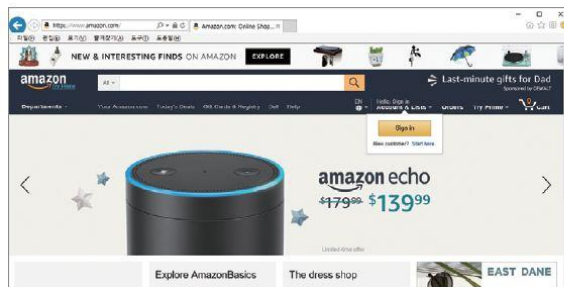
네이버

그림 1-4 정보 검색 플랫폼

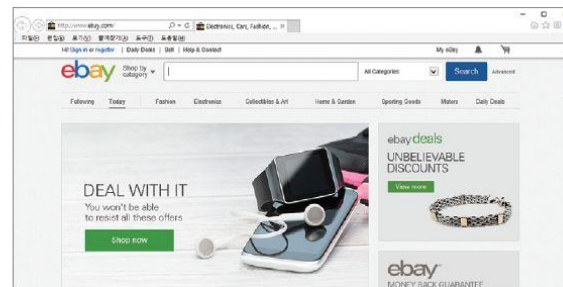
웹의 활용 분야 - 온라인 쇼핑

■ 온라인 쇼핑

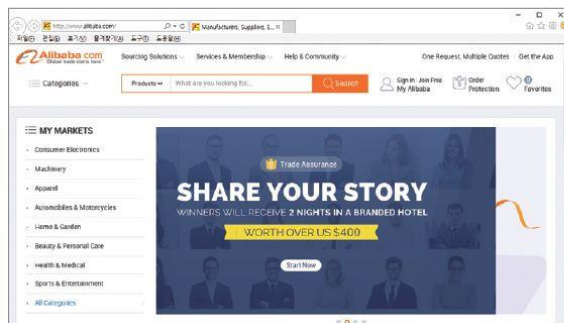
- 상품 정보와 이용 후기를 공유하고 상품을 직접 구매할 수 있는 플랫폼을 제공하여 수익을 창출
 - Aamazon, ebay, alibaba, 옥션(auction) 등



아마존



이베이



알리바바



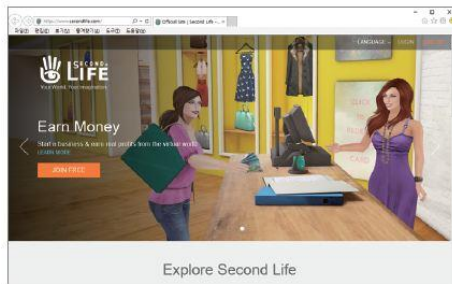
옥션

그림 1-5 온라인 쇼핑물 사례

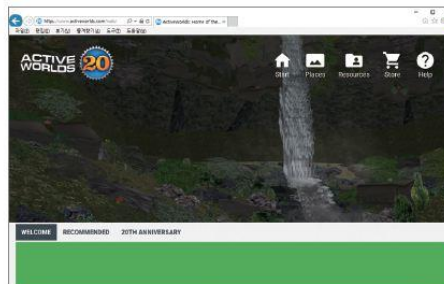
웹의 활용 분야 - 가상공간 서비스

■ 가상공간

- 현실 세계를 웹에 가상으로 구축해놓은 것
- 세컨드 라이프(secondlife), 액티브 월드(activeworld), 오픈 시뮬레이터(opensimulator) 등



세컨드 라이프



액티브 월드



오픈 시뮬레이터

그림 1-6 가상공간 플랫폼 사례

웹의 활용 분야 - 이러닝

■ 이러닝

- 정보통신기술을 활용하여 언제(anytime), 어디서(anywhere), 누구나(anyone) 수준별 맞춤형 학습을 할 수 있는 웹 서비스
 - 무들(Moodle), 프론터(Fronter) 등



무들



프론터

그림 1-7 이러닝 플랫폼 사례

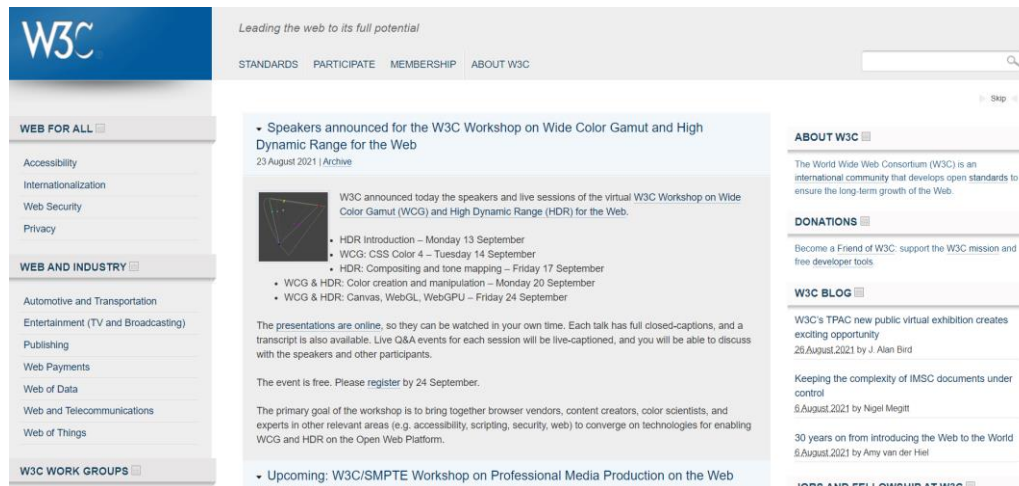
웹 표준과 HTML5

웹 표준 필요성

- 개발자 간 일종의 약속으로, 누가 개발하든 정해진 규칙을 준수하면 모두 호환되어 편리하게 사용

W3C (World Wide Web Consortium)

- 웹에 관련된 기술과 웹 브라우저 사용을 위한 표준안 제정
- 웹 개발자나 사용자 간의 정보 공유 및 신기술 개발 등에 기여



<http://www.w3.org>

■ 웹 표준 기술 문서의 발전 단계

- 초안 : 아직 완전하지 않은 워킹 그룹의 아이디어
- 후보 권고안 : 심사한 작업의 최종안(제안 권고안이 되기 위한 후보 문서)
- 제안 권고안 : 4주 이내에 검토해서 최종적으로 권고안이 되는 문서
- 권고안 : 최종적으로 표준화한 규격을 정의한 문서

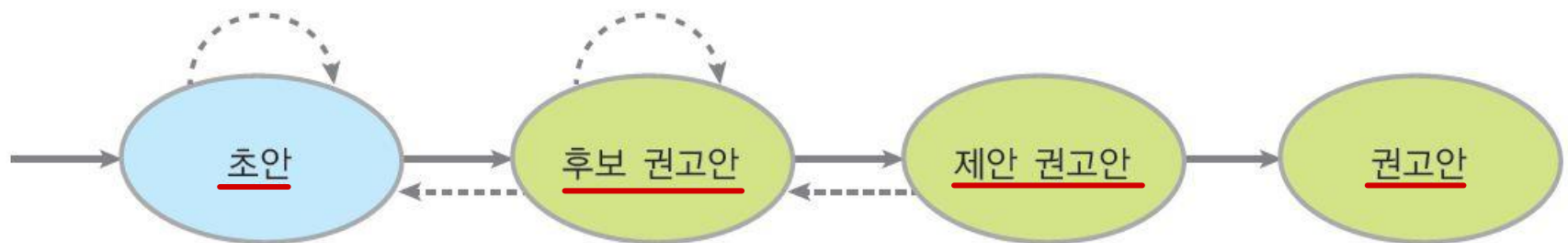


그림 1-9 웹 표준 기술 문서의 발전 단계

■ HTML5

- W3C와 WHATWG의 두 단체가 협동하여 제안한 새로운 HTML 규격

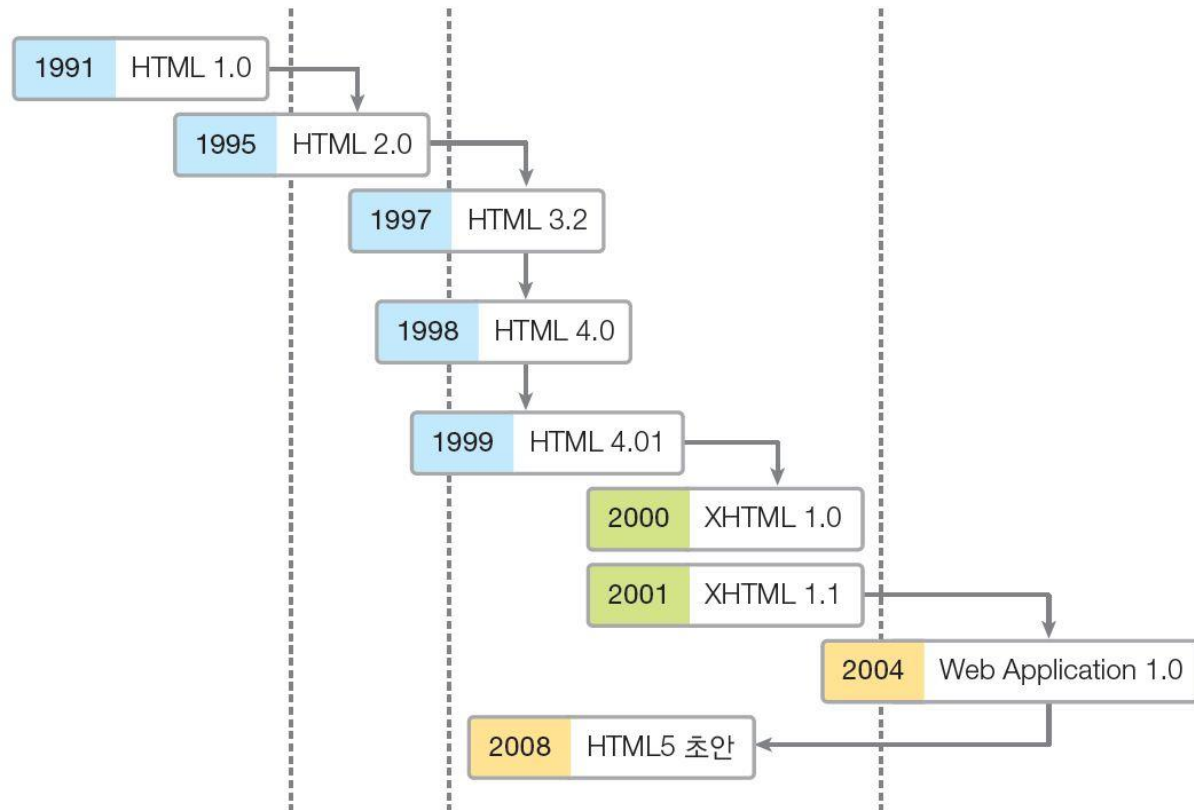
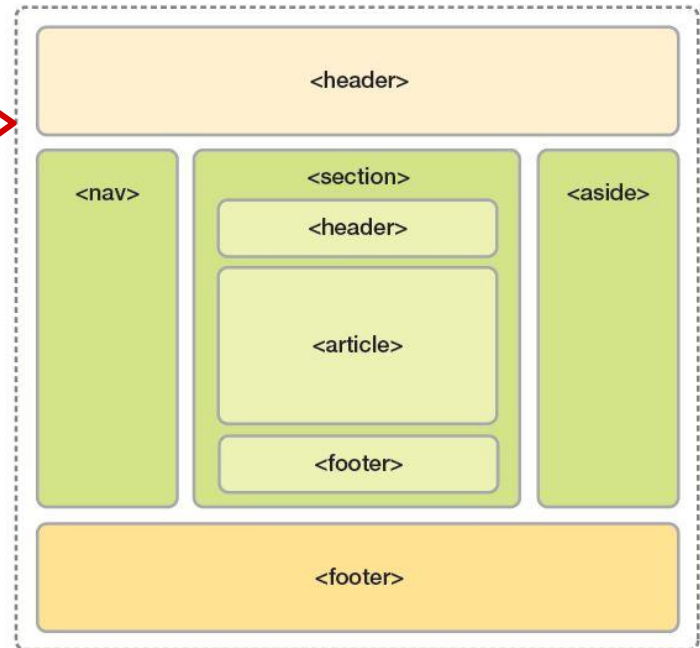


그림 1-10 HTML의 진화 과정

■ HTML5의 특징

- 구조적 설계 지원
- 그래픽 및 멀티미디어 기능 강화
- CSS3 지원
- 자바스크립트 지원
- 다양한 API 제공
- 모바일 웹 지원



〈header〉: 머리글처럼 페이지의 상단에 표시하는 부분

〈footer〉: 꼬리말처럼 페이지의 하단에 표시하는 부분

〈article〉: 본문 내용을 표시하는 부분

〈section〉: 문장의 그룹처럼 세부 내용을 그룹화할 때 사용하는 부분

〈aside〉: 본문에 관련된 주석처럼 별도의 내용을 분리하여 보여줄 때 사용하는 부분

〈nav〉: 내비게이션을 표시하는 부분

그림 1-11 HTML5의 구조적 설계

웹 브라우저와 HTML 편집기

- 웹 브라우저 (Web Browser)
 - HTML 문서를 읽고 웹 페이지에 정보를 표시하는 소프트웨어
 - HTML, CSS, JavaScript interpreter 내장되어 있음
 - URL(Uniform Resource Locator)을 통해 접근
- 웹 브라우저 기능
 - 웹 페이지 탐색
 - 접속하는 프로그램의 주소 관리
 - 웹 페이지의 저장, 인쇄, 소스 파일 보기
 - 웹 페이지 보안에 관련된 각종 필터 도구 제공

웹 브라우저와 HTML 편집기

■ 웹 브라우저 종류



▲ 구글 크롬



▲ 마이크로소프트 인터넷 익스플로러



▲ 모질라 파이어폭스



▲ 애플 사파리



▲ 오페라

웹 브라우저와 HTML 편집기

■ HTML 편집기 종류



▲ 메모장



▲ 노트패드++



▲ 울트라에디트



▲ 에디트플러스



▲ 아크로에디트



▲ 비주얼 스튜디오



▲ 어도비 브라켓

웹 서버의 개념

■ Web Server/Client 환경



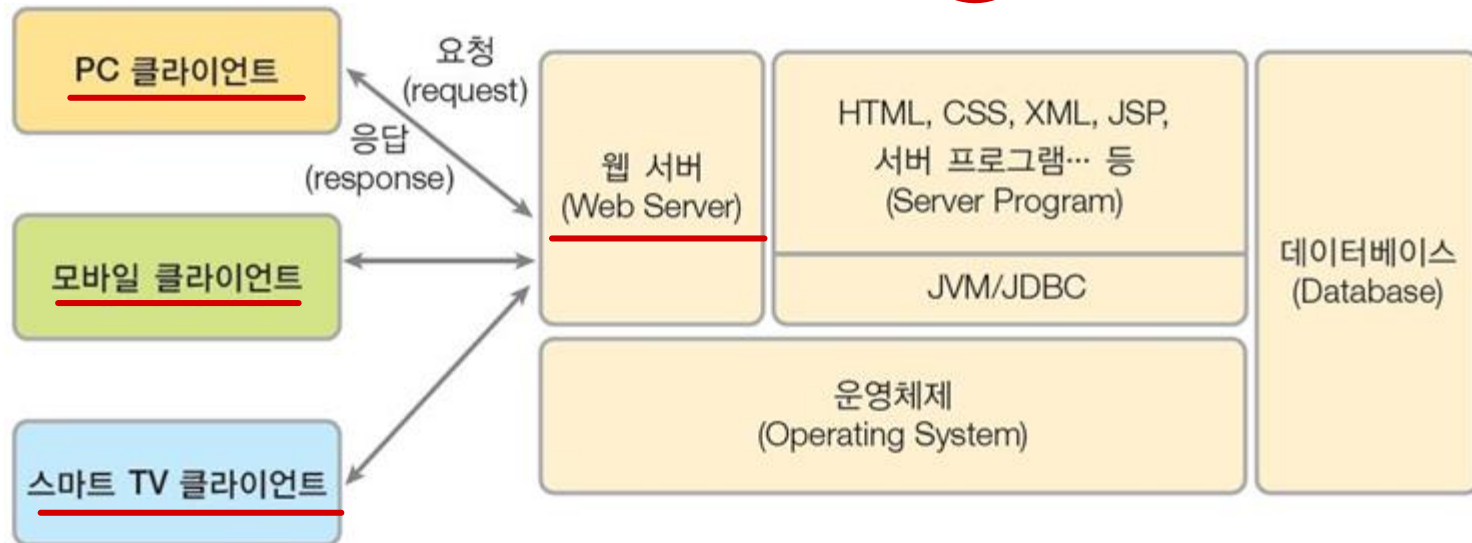
웹 서버의 개념

■ 웹 서버란?

- 클라이언트 요청에 따라 서버에서 HTML 문서 또는 정보 제공
 - 제공 문서 종류 : html, css, javascript, image 등 (정보 : json data)

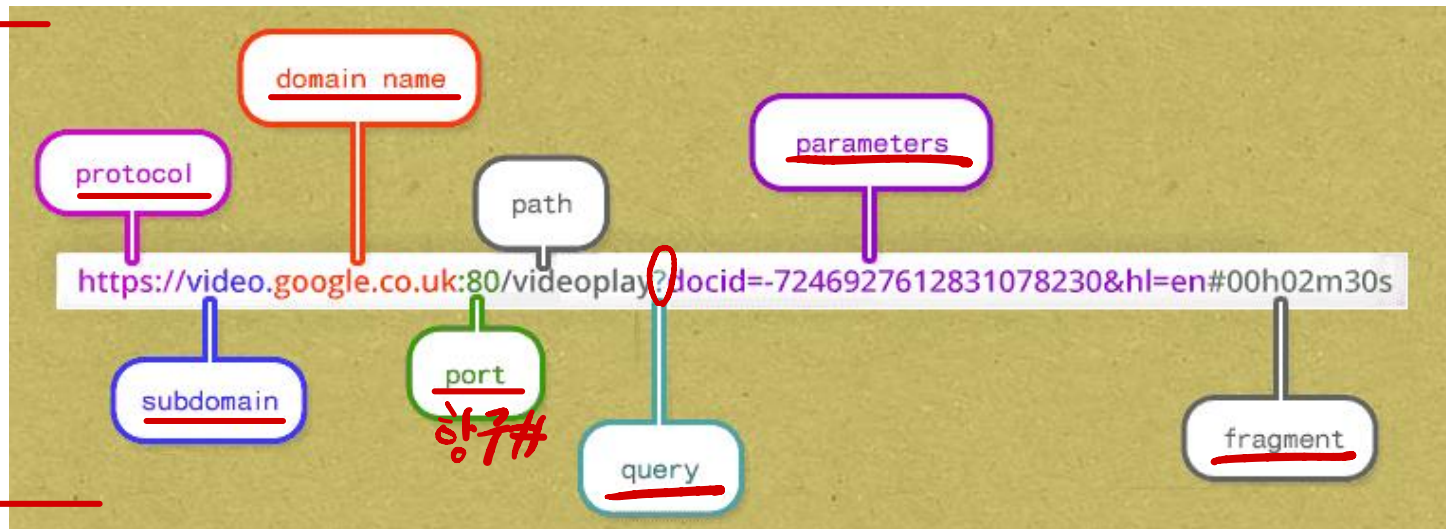
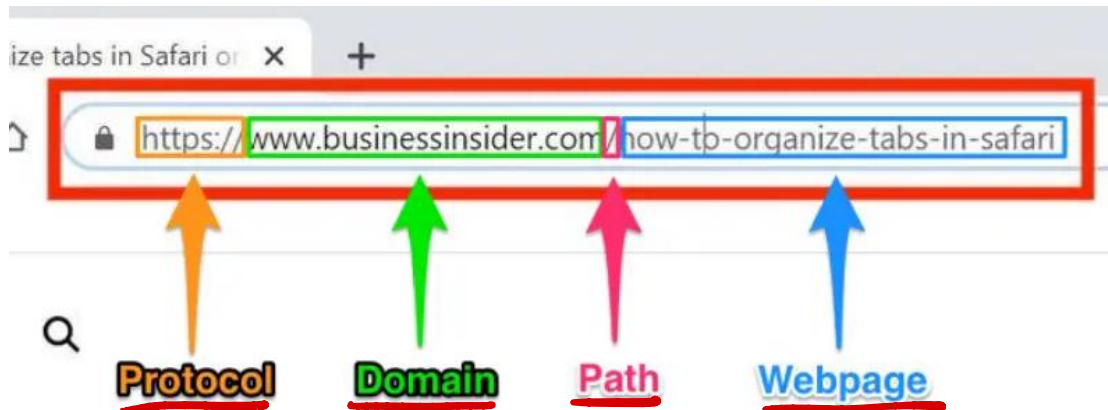
■ 웹 서버의 기능

- 리스너 기능 : client로부터 접속이 있는지 항상 체크하고 대기
- 답변 기능 : 요청한 사항을 처리한 후 결과를 클라이언트에 보냄



웹 서버의 개념

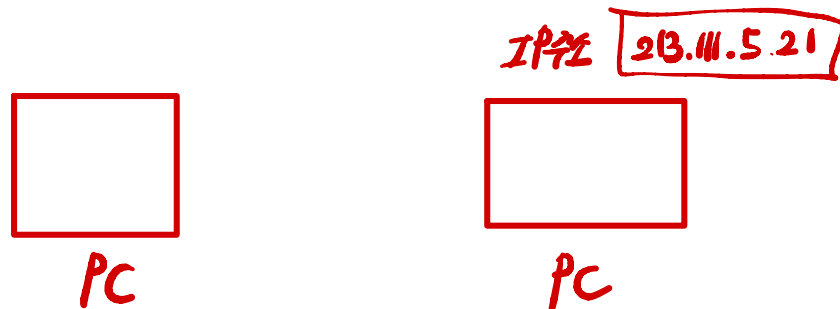
■ URL(Uniform Resource Locator)



웹 서버의 개념

- 프로토콜 *http* ——— *통신, Network*
 - 네트워크에서 제공하는 일종의 데이터 전송 규약
 - 웹 서버는 HTTP(Hyper Text Transfer Protocol)를 통해 클라이언트와 대화
- 포트
 - 웹 서버에서 일종의 문
 - Application program마다 고유의 port number 존재
 - 아파치 톰캣(Apache Tomcat)의 경우 8080번 포트를 사용

ex) 아파치 (포트 값이 8080)



웹 서버의 개념

■ 포트 번호

- 0~65,535번 중에서 사용 가능 *16bits(2¹⁶)*
- 미리 예약된 포트 번호 : 0~1,023번
- 사용할 수 있는 포트 번호 : 1,024~65,535번

표 2-1 미리 예약된 프로토콜과 포트 번호

프로토콜	포트 번호
HTTP(Hypertext Transfer Protocol)	80
<u>HTTPS</u> (Hypertext Transfer Protocol(Secure))	443
FTP(File Transfer Protocol(Data))	20
FTP(File Transfer Protocol(Control))	21
Telnet(Telnet Protocol)	23
SMTP(Simple Mail Transfer Protocol)	25
POP3(Post Office Protocol-Version 3)	110
IMAP(Internet Message Access Protocol)	143
SSH(Secure Shell Remote Login Protocol)	22



웹 서버의 개념

■ 웹 서버 제품 종류 (예시)

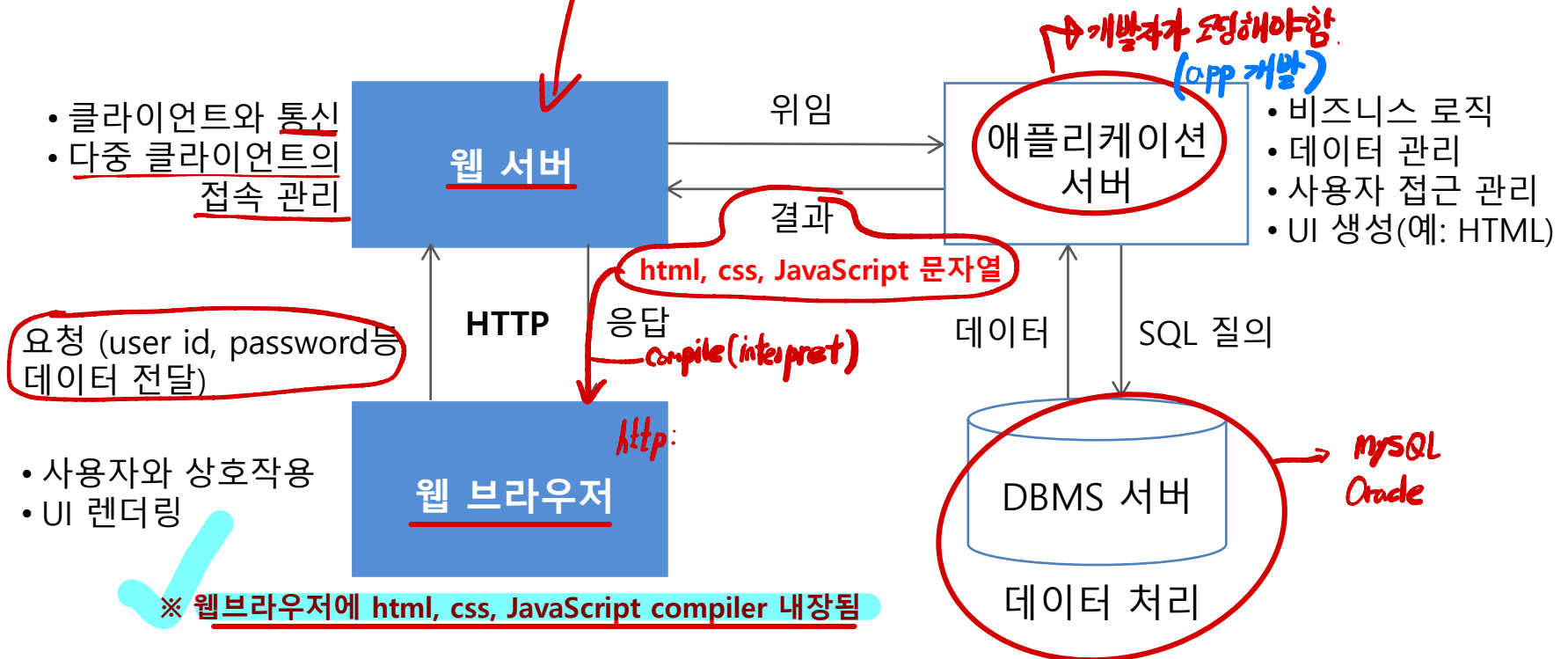
참고

이름	제작사	라이선스	참고 사이트
<u>아파치 HTTP 서버</u>	아파치 재단	오픈소스	http://httpd.apache.org
<u>IIS 웹 서버</u>	<u>마이크로소프트</u>	상용	http://www.microsoft.com (윈도우 서버 OS에 포함됨)
엔진 X	엔진 X, Inc	오픈소스	http://<u>nginx.org</u>
라이티	버클리 대학(BSD, Berkeley Software Distribution)	오픈소스	http://www.<u>lighhttpd.net</u>
아파치 톰캣 서버	아파치 재단	오픈소스	http://<u>tomcat.apache.org</u>

Web Application Architecture

■ 웹 애플리케이션 서버 구조

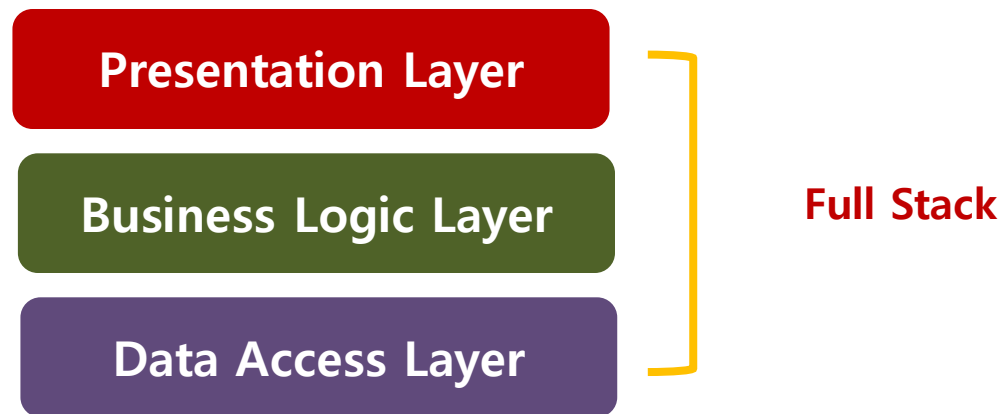
- 클라이언트와의 통신은 웹 서버가 전담
- 애플리케이션 서버는 애플리케이션 실행 및 관리에 집중
 - 네트워크 및 multi-thread 프로그래밍으로부터 탈출



Full Stack JavaScript Web Development

■ Full Stack 정의

- ① Presentation Layer : UI 관련 작업
- ② Business Logic Layer : dynamic data 처리, data validation
- ③ Data Access Layer : data access, data persistence



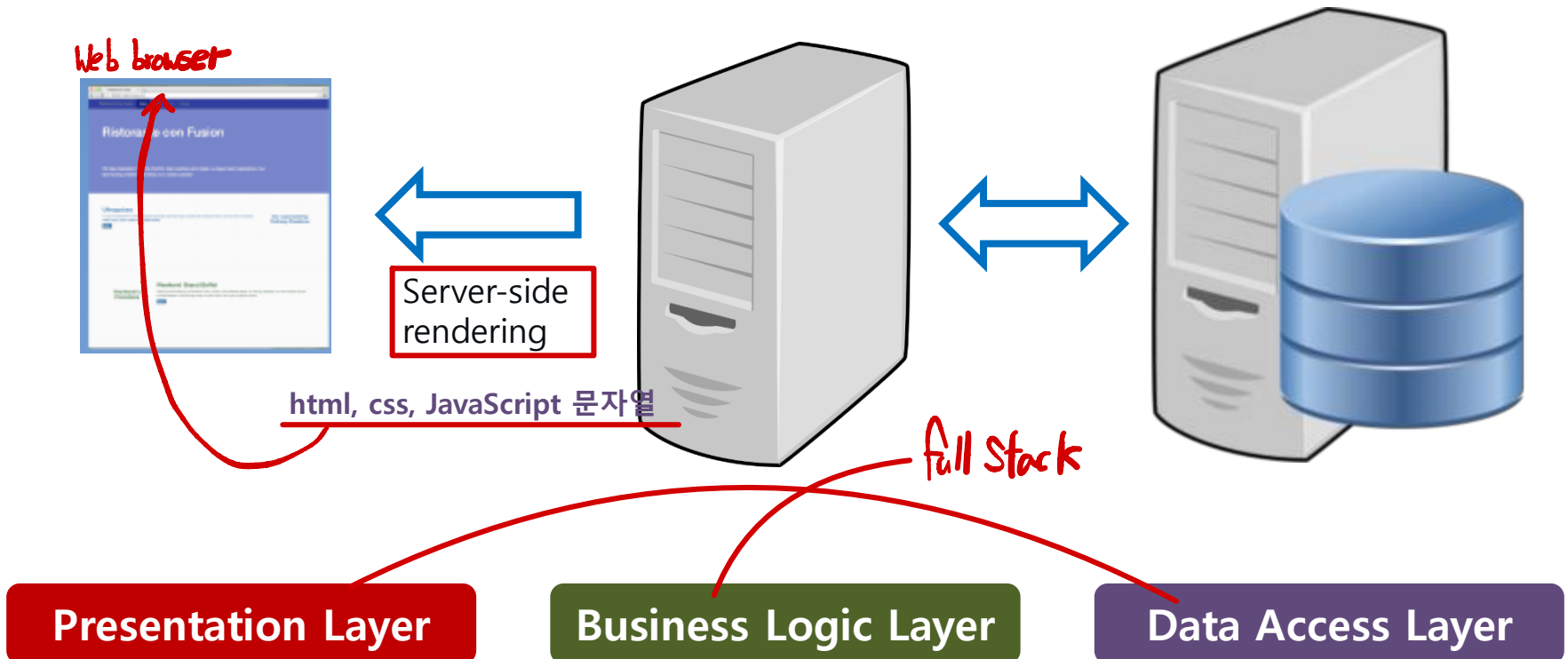
Full Stack JavaScript Web Development

■ Traditional Web Development

HTML, CSS, JavaScript

Ruby, Python, Java, C++, PHP

DBMS (Oracle, MySQL)



Full Stack JavaScript Web Development

Full Stack JavaScript Development

Front end

Backend

Single page Apps
using JavaScript Frameworks/libraries
like Angular or React

Bootstrap

Rest API
serving JSON

html, css, JavaScript code

NodeJS and
NodeJS Modules

MongoDB
JSON documents

Presentation Layer

Business Logic Layer

Data Access Layer

Full Stack JavaScript Web Development

■ JavaScript기반 Web Design and Development

- UI Design

- User Interface(UI) / User Experience(UX) Design
- Visual Design
- Prototyping
- Colors, Graphics and Animation
- HTML + CSS + JavaScript
- Front-end (Client-side)

↓
UI Designer
(UX)

- Development, Building and Deployment
 - Web UI Frameworks: Bootstrap 4
 - JavaScript Framework/Libraries: Angular/React
 - Hybrid Mobile Frameworks: Ionic, Cordova, NativeScript and React Native
 - Server-side Deployment: Node + Express + MongoDB

Full Stack Web Developer

Front-end Web Application 개발 Framework

- Front-end Web App 개발 환경

- HTML

- CSS

- Responsive web

- JavaScript (ES6:ECMA2015)

- TypeScript

- JavaScript Framework

- Angular

- React (flux, redux)

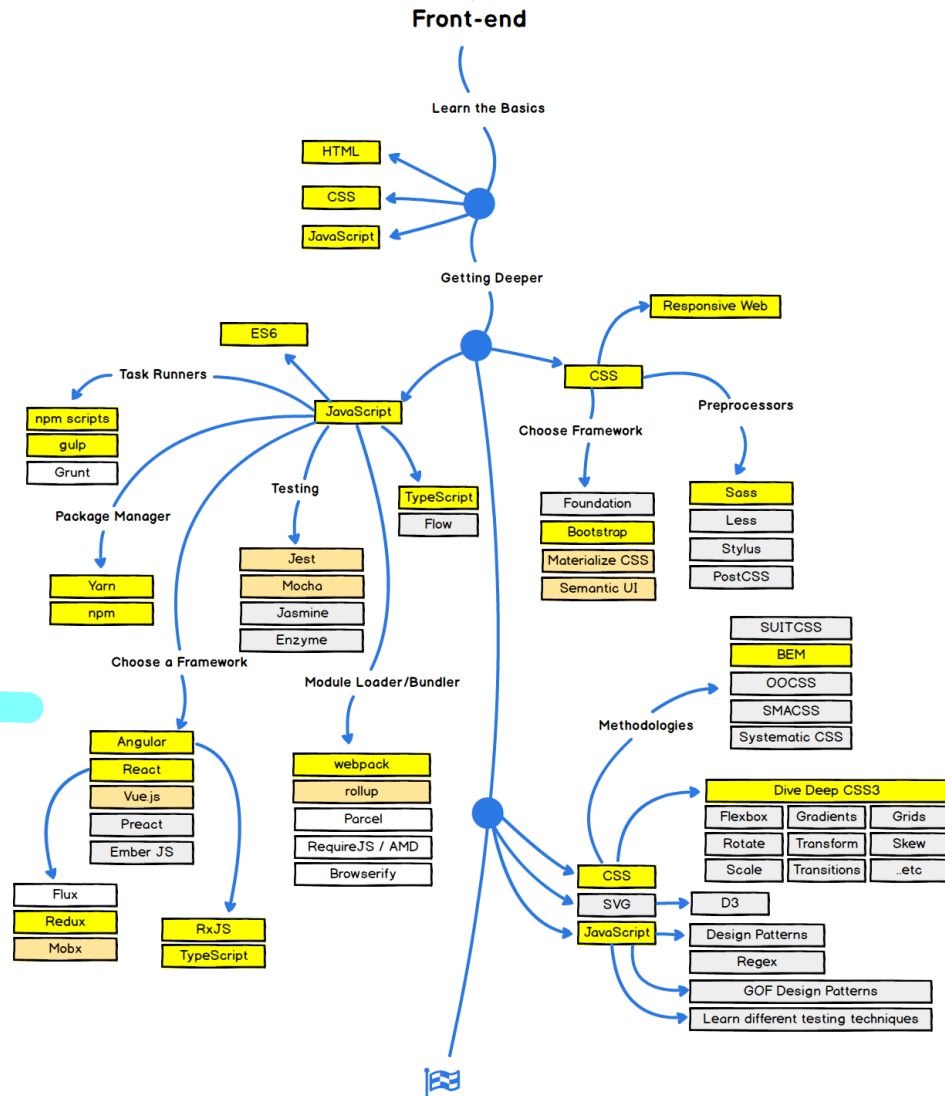
- Vue.js

- Package Manager (yarn, npm)

- Module Loader/Bundler

- Webpack, Browserify

- Testing (Jest, Mocha)



■ 강의 주요 내용

- HTML, CSS, JavaScript, Node.js, Rest API 소개
- Bootstrap and/or jQuery 소개

■ 수업 실습 환경

- Visual Studio Code tool
 - HTML, CSS, JavaScript editor
 - Node.js 실습에서도 사용