

## - 웹프로그래밍 언어

: Web Browser에서 동작하는(실행) 언어

# 1. Web FrontEnd

## 1) HTML - Hyper Text Markup Language

- 화면구성 = GUI

ex) 로그인 화면, 가입폼, 게시판폼, 게시판목록, 방명록, 장바구니화면

주로 사용자 입력 값을 받는 폼구성, 사용자가 요청한 결과를 화면폼에 출력

: version - HTML4 --> HTML5

현재는 HTML5를 사용한다 - HTML5는 다양한 API를 제공

(Web Storage, Drag&Drop, Audio, Video, WebSocket...)

: 대소문자를 구분하지 않는다. 오타를 내면 그냥 적용 안되고 실행은 된다!!

## 2) CSS - Cascading Style sheet

: HTML에 디자인을 추가한다(화면의 레이아웃- template)

: CSS를 이용하면 시각적으로 풍부하게 만들 수 있다.

과거에는 플래쉬나 포토샵을 이용해서 시각적으로 풍부하게 만들었으나 요즘에는 CSS3로 대부분 가능하다

: 대소문자를 가린다, 다만 오타를 내면 실행은 되나 css가 적용이 안된다!!

## 3) JavaScript - HTML에 기능(동작=Event)을 추가

: 대소문자를 완벽하게 구분한다.

: 자바의 개념과 비슷하다(객체기반의 언어로 객체의개념, 메소드 + 속성)

: 인터프리터 언어이다 .- 소스코드를 한줄씩 해석하면서 오류를 순차적으로 출력해준다)

웹브라우저에서 오직 싱글스레드로 동작한다. - 동기화작업(블로킹-bloking)

## \* 결론

: Web Browser (IE, 크롬, 파이어폭스, 사파리, 오페라....) 는 기본적으로 3가지 언어만 지원한다  
브라우저가 해석할 수 있는 언어는 3가지

1) HTML로 구조

2) CSS로 표현

### 3) JavaScript로 동작

: Web Front언어의 단점은 소스가 100%로 공개가 된다.

CS(Client Side)언어이다. 클라이언트가 가지고 있는 웹 브라우저에서 실행되는 언어로 브라우저의 영향을 받는다. - 단점

같은 소스라 하더라도 브라우저마다 해석하는 방법이 달라서 실행결과가 달라질 수 있다 - 단점

\* 이러한 단점을 해결하기 위해서 요즘은 다양한 JavaScript기반의 LIB를 제공한다.

그 중의 대표적인 lib jQuery가 있다.

jQuery는 모든 브라우저에서 동일하게 동작할 수 있도록 만들어 놓은 경량의 JavaScript기반의 lib이다.

: DB연동의 기능이 없다. - 영속성문제 있다.

: HTML로 사용자 입력한 값을 Back단으로 전송할 수 있지만 전송된 다음 페이지에서 그 전송된 데이터를 받을 수 있는 능력이 없다.

: 웹은 많은 페이지를 이동하는데 이동할 때 마다 새로운 요청(request, response) 생성되는 것으로 상태정보를 유지할 수 없다.

\* 위 단점 때문에 결국 Web BackdEnd 기술의 언어가 필요하다.

- Servlet&JSP or ASP or PHP

- **Static Web Application** : HTML, CSS, JavaScript로 이루어진 프로그램

- **Dynamic Web Application** : HTML, CSS, JavaScript + BackEnd 언어 함께 만들어진 프로그램

### \* MEAN Stack 개발자

1) Mongo DB : NoSQL(비 관계형 데이터베이스)

2) Express JS : Angular 와 Node 를 중간에 연결해주는 역할 - Controller 의 개념

3) Angular JS : View

4) Node JS : BackEnd 기술

\* JavaScript 기반의 언어가 새롭게 등장

js 기반으로 만들어 놓은 LIB 형태 = FrameWork( Model + View + ViewModel) : MVVM

### 1) Angular.js 1 , Angular.js 2 : 구글 + 개별 커뮤니티에서 만든 프레임워크

- 폼처리, 상태관리, 라우팅 등 완전한 기능 세트를 제공
- 데이터 바인딩, DOM 조작 쉽게 처리
- TypeScript 기반으로 하여 정적 타입 검사를 지원하므로 코드의 안정성
- 학습이 어렵고 복잡한 프레임워크 구조
- 다른 프레임워크에 비해 크기가 크고 초기 로딩시간이 오래 걸림.
- 업데이트 주기가 빨라 업데이트에 따른 코드 변경이 잦다.

### 2) React.js : 페이스북에서 개발된 UI 구축을 위한 자바스크립트 라이브러리

- 가상 DOM(Virtual DOM) : 빠른 UI 업데이트를 위해 가상 DOM 을 활용하여 실제 DOM 조작을 최소화
- 컴포넌트 기반 : 재사용 가능한 컴포넌트를 통해 코드의 가독성과 유지보수성을 향상
- 활발한 커뮤니티 : 큰 생태계와 다양한 지원을 받을 수 있다.(커뮤니티 크기의 강점)
- 학습곡선이 가파르고 복잡
- React 는 뷰 렌더링에 초점을 맞춘 라이브러리로 라우팅, 상태관리와 같은 추가 기능은 별도의 라이브러리 필요.

### 3) Vue.js : EvanYou 라는 사람이 개인적으로 만듦

- 가벼운 자바스크립트 프레임워크로 개발자 친화적인 문법과 유연한 구조
- 가벼운 크기와 빠른 속도
- React 나 Angular 에 비해 생태계가 상대적으로 작아 지원되는 플러그인 라이브러리 수가 작다.

## \* 기타

**Next.js** - React 를 위한 프레임워크

**Nuxt.js** - Vue 를 위한 프레임워크

**Node.js** - 자바스크립트를 서버에서도 사용 할 있도록 만든 프로그램

V8 이라는 JavaScript 엔진 위에서 동작하는 자바스크립트 런타임(환경)이다.

서버사이트 스크립트 언어가 아니다. 프로그램(환경)이다.

**Vanilla JS** - 프레임워크 또는 라이브러리가 적용되지 않은 순수한 자바스크립트

**TypeScript** - JavaScript 에 타입을 부여한 언어로 JavaScript 확장된 언어

Microsoft 에서 개발 및 관리되고 있는 오픈소스

**Babel** - 바벨은 최신 버전의 자바스크립트가 실행되지 않는 구 버전의 브라우저에서도 정상적으로 실행되도록 변환 해주는 NodeJS 패키지

ex) ES6 코딩 --> ES5 으로 변환

각 브라우저마다 JavaScript 엔진이 다르지만 모든 브라우저에서 동작하도록 호환성을 지켜준다.

**Webpack** - 모듈 번들러로 웹 애플리케이션을 구성하는 자원(HTML, CSS, JavaScript, Image 등)을 각각의 모듈로 보고 이를 조합해서 하나의 결과물을 만드는 도구

html 파일에 들어가는 자바스크립트 파일들을 하나의 자바스크립트 파일로 만들어주는 방식으로 필요한 다수의 자바스크립트 파일을 하나의 자바 스크립트 파일로 만들어 주는 것을 Webpack 이라함.

**ESLint** - ES(Ecma Script)와 Lint 를 합친 것.

ES 는 Ecma 라는 기구에서 만든 Script, 즉, 표준 Javascript 를 의미

Lint 는 에러가 있는 코드에 표시를 달아놓는 것을 의미

즉, ESLint 는 자바스크립트 문법에서 에러를 표시해주는 도구.

## 2) Web BackEnd 기술( Servlet&JSP 또는 ASP 또는 PHP)

: 우리는 자바 기반인 Servlet&JSP 학습

### \* Java 영역

- Java SE : 응용프로그램 개발
- Java EE : 웹프로그래밍 개발
- Java ME : 모바일환경 프로그램 개발

### : Servlet & JSP (Dynamic Web Application)

- JSP 문서에서 HTML, CSS, JavaScript 작성 : VIEW 역할
- Servlet : Controller 역할
- Java SE : Mdel 역할 (Service, DAO, DTO=VO,....)

\* 결론 : 웹프로젝트를 진행할 때 MVC 구조로 웹사이트 구축한다.

DB 연동을 해서 데이터 영속성을 유지 할 수 있다. - JDBC 를 사용한다.

: 웹브라우저는 HTML, css, JavaScript 만 해석가능하다.

그렇다면 Servlet&JSP 는 누가 해석해주나?? --> 브라우저가 해석못한다.

그래서, BackEnd 언어를 해석해서 브라우저가 해석할 수 있는 HTML 언어로 변환해주는 Servlet Container=WAS 가 필요하다.

Apache tomcat = WebServer +Servlet Container(Web Application Server)로 구성되어 있으며  
<https://tomcat.apache.org/> 에서 Apache Tomcat 을 무료로 이용한다.

**\* Ajax 기술 - 비동기화 통신 = ASynchronous JavaScript And XML**

: 화면의 새로고침없이 서버와 통신(XMLHttpRequest)을 해서 그결과를 받아 화면의 일부분을 갱신(update)해주는 기술이다.

: JavaScript 언어를 이용해서 서버와 통신을 하고 그결과를 xml 형태로 받아서 화면을 갱신한다.

요즘은 JavaScript 는 자바스크립트 내장함수 fetch 함수 또는 외부라이브러리 jQuery , axios 사용

xml 은 JSON 의 형태로 많이 사용한다.

**\* HTML(Hyper Text Markup Language) :** 이미 DTD 에 정의되어 있는 태그만 사용할 수 있다.

: Markup 은 태그로 ( < > ) 이루어진 언어를 뜻한다.

**\* XML (Extensible Markup Language) -** 필요한 태그를 직접 만들어서 사용할 수 있다.

: 서로 다른 언어사이에서 데이터를 주고 받을때 데이터 포맷 형태로 많이 사용한다.

ex)

<과일>배</과일>

<운송수단>배</운송수단>

ex) 게시판의 목록을 뷰쪽으로 전달

<boards>

<board>

<no>1</no>

<subject>제목</subject>

<content>내용</content>

</board>

<board>

<no>2</no>

<subject>제목</subject>

<content>내용</content>

</board>

<board>

```

    <no>3</no>
    <subject>제목</subject>
    <content>내용</content>
  </board>
</boards>

```

\* xml 은 문법이 엄격하고 무겁고 결과를 가져와서 파싱(해석)하는 과정이 복잡하다 - 단점  
요즘은 xml 형태를 JSON 형태로 많이 사용한다.

JSON 의 구조 -->

```
[{no:1 , subject: "제목", content : "내용"} ,{no:2 , subject: "제목", content : "내용"} ,{} ,..... ]
```

\* 요즘 xml 은 주로

- 설정문서 (configuration)
- ex) web.xml , server.xml .....

## 웹프로그래밍을 시작하기 위한 준비

1) 웹브라우저 필요( IE, 크롬, 오페라....) : 크롬을 많이 사용한다.

2) 어디서 코딩할까? : 메모장에서 작성을 하지만 코딩을 손쉽게 할수 있도록 도와주는  
다양한 tool(Visual Studio Code , Aptana Studio, eclipse, IntelliJ)이 제공된다.  
우리는 eclipse 에서 작성!

3) 코딩한 코드를 어떻게 실행할까?

: FrontEnd 언어는 브라우저를 선택하고 브라우저에서 그냥 실행하면된다.

~.html 문서를 더블클릭하면 브라우저에 의해 실행된다.

이러한 동작은 local pc 에서 그냥 실행하는 것으로 웹서버에서 웹서비스가 실행되는것이 아니다.

우리는 파일이 실행될때 local 아니라 웹서버(http)에서 동작할수 있도록 Apache tomcat 가 필요하다.

http://도메인:port/파일이름 실행을 해야만 Apache tomcat 에서 동작한다.

\* tomcat.apache.org 사이트에 가서 다운로드 받는다. - tomcat 10

-다운로드 받을 때

- 1) install version : 운영모드에서 권장
- 2) 압축 version : 개발모드에서 권장 - 우리는 압축버전

주의 : tomcat 의 port 번호가 기본 8080 이다.

oracle 에서 이미 8080 port 를 사용하고 있다.

충돌이 나지 않도록 tomcat 의 port 를 변경해서 사용한다.

server.xml 수정

## SPA(Single Page Application)

: 웹 사이트의 전체 페이지를 하나의 페이지에 담아 동적으로 화면을 변경해 표시하는 기술  
브라우저는 최초에 한번 서버에 요청하여 페이지 전체를 로드하고 이후부터는 특정 변경 사항이 일어나는 부분만 Ajax 와 같은 기술을 통해 데이터를 바인딩하는 방식으로 동작(사용자에게 바른 인터렉션 제공)

## SSR(Servier Side Rendering ) vs CSR(Client Side Rendering) 개념

1) SSR : 서버 사이드 렌더링이라는 뜻으로 구동 방식은 서버에서 렌더링하여 완성된 HTML 파일을 로드해 준다.

클라이언트에서 요청할 때마다 각 상황에 맞는 HTML 파일을 넘겨주기 때문에 페이지가 여러 개 이므로 MPA(Multi Page Application) 에 적합한 환경

2) CSR : 클라이언트 사이드 렌더링이라는 뜻으로 구동 방식은 초기 로드 시 빈 HTML 과 모든 로직이 담겨 있는 Javascript 다운로드를 하고 그 후 빈 HTML 에 Javascript 를 이용하여 Dom 을 동적으로 생성하여 그린다. - SPA 에 적합한 환경

\* SSR 는 검색엔진 최적화가 가능하지만 CSR 방식은 검색엔진 최적화가 어렵다.

: 포털사이트 검색엔진 크롤러가 데이터를 수집하지 못하는 경우가 많기 때문

최근에는 구글 검색엔진에 자바스크립트 엔진이 내장되어 있어 크롤링이 된다는 이야기도 있음(구글이 공개하지 않음)

네이버나 다음 같은 경우 CSR 방식으로 만든 페이지의 데이터를 검색하지 못하는 이슈가 있어 별도의 보완작업이 필요.

SSR

- 네트워크가 느릴 때

CSR

- 네트워크 빠를 때

- 검색엔진 최적화가 필요할 때
- 검색엔진 최적화가 필요 없을 때
- 페이지 상호작용이 별로 없을 때
- 페이지 상호작용이 많이 필요할 때

\* React 는 CSR 방식이라 SEO 에 어려움이 있기 때문에 최근에는

NEXT.js 와 같은 프레임워크를 통해 서버사이드렌더링을 가능하게 하는 방식을 많이 도입하고 있다.

NEXT.js 는 pre-reloading 을 통해 미리 데이터가 렌더링된 페이지를 가져올 수 있게 해주므로 사용자에게 더 좋은 경험을 주고, 검색 엔진에 잘 노출 될 수 있도록 해주는 SEO 에서도 장점을 얻을 수 있다.

## SEO(Search Engine Optimization)

: 검색엔진 최적화로 검색 결과에서 상위에 노출될 수 있도록 내 콘텐츠를 최적화하는 방식을 말한다.

## Dynamic web project directory 구조

1) src/main/java : servlet 과 자바관련 파일 넣는곳( ~.java)

2) src/main/webapp : FrontEnd 관련 소스들

HTML, css, javaScript 등이 있는 곳

jsp 문서

\* src/main/webapp 이부분이 웹의 root 이다.

3) src/main/webapp/WEB-INF 폴더에 FrontEnd 관련 파일 넣으면 실행안 된다!!!

3) src/main/webapp/WEB-INF/lib 폴더에 BackEnd 관련 lib 넣는곳( ex) 오라클라이브러리 )

## -알아두면 좋은 사이트

1) <https://www.w3schools.com/>

: 온라인으로 웹 기술을 배우는 교육용 웹 사이트

2) <https://developer.mozilla.org/ko/>



: 모질라 재단 및 다른 IT 기업들이 사용하는 웹 개발을 위한 문서 저장소이자 수많은 프로그래밍 입문자들을 위한 학습 장소이다

3) <https://stackoverflow.com/>

: 개발자들이 프로그래밍을 하다 막히거나 할 때, 프로그래밍에 대한 질문을 하고 답변을 받는 사이트

4) <http://jsbin.com/>

: html, css, javascript 코드를 작성하고 실행 테스트를 할수 있는 웹서비스

5) <https://jsonplaceholder.typicode.com/>

: 샘플 데이터(가상데이터)를 제공하는 무료 온라인 REST API 서비스

6) <https://picsum.photos>

: 무료로 제공하는 샘플 이미지 가져오기

7) <https://caniuse.com/>

: 여러가지의 태그들이 각각의 웹 브라우저에서 지원되는지 알려주는 사이트

8) <https://imagecolorpicker.com/>

: 색상코드를 뽑아내는 사이트

[8253jang@daum.net](mailto:8253jang@daum.net)

장희정