

프론트 개발을 위한 웹퍼블리싱

2022.12

지도교수 : 박인상

웹 프로그래밍 학습 로드맵(전문 웹 프로그래머)



웹 표준은 왜 지켜야 하는가?

- 월드 와이드 웹(WWW)을 구현하기 위해 따라야 할 표준 또는 규격.
다양한 웹 기술을 통해 수많은 콘텐츠가 공유되고 확산되면서, **웹 접근성, 사생활 보호, 보안, 국제화** 등을 고려한 웹 기술의 표준화가 필요하게 되었다.
- 웹 표준의 시작은 내용과 디자인의 분리이다.

이유 1. 유지보수가 쉬워진다.

이유2. 웹 접근성을 확보할 수 있다.

웹 표준 기술

Front-end Roadmap



HTML5 - 웹 페이지 구성

CSS3 - 스타일 적용

JavaScript - 사용자 반응 처리 등

- HTML5로는 웹의 기본적인 문서 내용/골격(Structure; 웹 문서의 문법)을 작성하고
- CSS3로는 웹을 디자인(HTML 문서를 표현)하며 즉 **Presentation 출력스타일 정의**
- JavaScript로는 웹의 동작(**Behavior**)/기능(사용자 반응)을 구현한다.
즉 자바스크립트는 웹브라우저로 분류되는 S/W들을 **프로그래밍적으로 제어**하기 위한 언어이다. 브라우저에서 벗어나 **웹 서버까지 영역을 확장**
- 사용자들과의 '상호작용'에 필요한 동적인 기능들을 구현한다.



CSS Framework

=> **Bootstrap** : 미리 만들어진 UI 요소를 **class** 이름으로 지정하기만 하면 간단하게 스타일링할 수 있다.

PostCSS

Material UI

Styled-Components

Javascript Framework

=> **React** : 자바스크립트 UI 라이브러리로 개발의 복잡성을 낮춰주는 기술

Vue

Angular

Svelte

jQuery : 표준 자바스크립트 프레임워크

HTML5 기본 용어 정리

1. 요소(Element)

- HTML 문서의 개별적인 구성요소
(예) h1, h2, p, sub, sup 등
- 요소는 객체라고도 함. 객체는 존재하는 무언가를 의미한다.

2. 태그(Tag)

- 요소를 만들 때 사용하는 기호이다.
즉, 부등호 기호(< >)로 둘러싸인 엘리먼트를 조합해서 부르는 이름
(예) <h1>, <h2>, <p>, <sub>, <sup> 등

3. 속성(Attribute)

- 태그에 추가 정보를 부여할 때 사용한다.
(예)

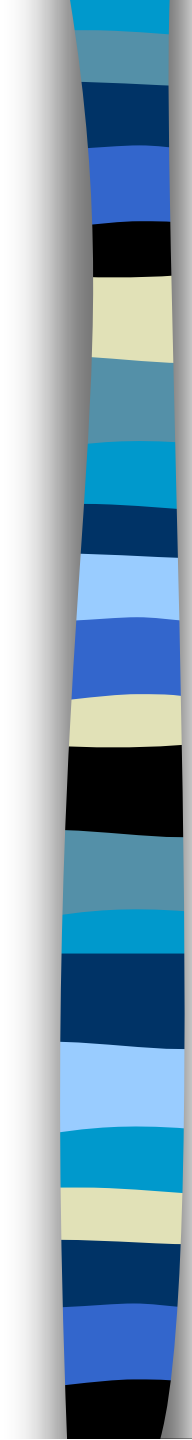
HTML5 페이지 기본 구조

```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>

</body>
</html>
```

■ head 태그 내부에 넣을 수 있는 태그

태그	설명
meta	웹 페이지에 추가 정보 전달
title	웹 페이지의 제목 지정
script	웹 페이지에 스크립트 추가 시 사용 즉 .js 파일
link	웹 페이지에 외부 스타일 시트 연결 시 사용 즉 .css 파일
style	웹 페이지에 스타일 정보를 정의할 때 사용
base	웹 페이지의 기본 경로 지정



HTML5

<!DOCTYPE html>

문서구조

부모, 자식, 형제 요소
조상, 자손 요소

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="EUC-KR">
    <title>Title</title>
  </head>
  <body>
    <h1>제목</h1>
    <p>내용</p>
  </body>
</html>
```

부모 요소는 여러 개의 자식 요소를 가질 수 있지만,
자식 요소는 하나의 부모 요소만 가진다.

HTML 기본 요소(element)

1. 글자 요소

- 제목 글자 요소 : h1 ~ h6
- 본문 요소: p, br, hr
- 하이퍼 링크 요소: a
- 글자 모양 요소: b, i, small, sub, sup, ins, del

2. 목록 요소 - ul, ol, li

3. 테이블 요소 - table, thead, tbody, tfoot, tr, th, td

4. 이미지 요소 - img

5. 공간 분할 요소 - 기본 공간 분할 요소: div, span

- 시맨틱 요소: header, nav, aside, section, article, footer



block 레벨 엘리먼트와 inline 레벨 엘리먼트

1. block 레벨 엘리먼트 종류

div, **p**, h1~h6, ol, ul, li, pre

2. inline 레벨 엘리먼트 종류

span, a, strong, em, img, ,button,
input, label, select, textarea

HTML5의 새로운 태그

- HTML5에서는 태그 자체에 의미를 부여하는 시맨틱 마크업(Semantic Markup)을 사용한다.

새로운 태그	설 명
<header>	문서의 머리말 지정
<nav>	메뉴 영역 지정, 즉 사이트 안의 내비게이션 요소를 표시한다.
<article>	콘텐츠 영역 지정, 즉 뉴스 기사나 블로그 글, 포럼에서의 포스팅 같은 독립적인 콘텐츠를 표시한다. 즉 홀로 설 수 있는 내용을 담는다.
<section>	문서 내용 구분, 즉 일반적인 문서나 애플리케이션 영역을 표시한다. 섹션의 제목을 나타내는 h1 ~ h6 태그 그리고 <p> 태그와 함께 사용할 수 있다. 즉 서로 관계 있는 문서를 분리하는 역할을 함.
<aside>	문서의 주요 부분을 표시하고 남는 사이드 바 콘텐츠를 표시한다.
<footer>	꼬리말 영역 지정, 즉 사이트 제작자나 저작권 정보 등을 나타낼 때 주로 사용된다.

HTML5 Semantic 태그를 적용한 메인 화면 설계

<header>

<div class="container1">

<nav>

<div class="menu">

<div id="wrapper">

<section id="main">

<section id="content">

<article> 블로그 글1</article>

<article> 블로그 글2</article>

<article> 블로그 글3</article>

</section>

<aside id="sidebar">

<footer>

<div class="container1">

HTML5 입력 양식 태그

태그	설명
<form>	입력 양식의 시작과 끝 표시
<input>	다양한 객체들을 생성
<textarea>	여러 행의 글자 입력 양식 생성
<select>	선택 양식 생성
<optgroup>	옵션 그룹화
<option>	옵션 생성
<fieldset>	입력 양식의 그룹 지정
<legend>	입력 양식 그룹의 이름 지정

젠 코딩(Zen Coding) = Emmet

젠 코딩은 HTML 코딩의 생산성을 향상시켜주는 도구로서 빠른 코딩과 코드 유형을 편집하기 쉽게 해 주기 위한 플러그인의 집합이다.

문법(기호)	의미
>	자식 요소 만듦
+	형제 요소 만듦
*	반복 시키기
\$	숫자 넣기
#	id
.	class
[attr]	속성 추가하기
{ }	텍스트 넣기

CSS3 선택자의 종류

(1) 선택자를 하나만 쓸 경우 – 단일 선택자

(예) `.navbar-default { background-color:#fece10; }`

(2) 선택자를 여러 개 쓸 경우 – 다중 선택자

(예) `h1, p { color:red; background:gold; font-size:20px; }`

(3) 자식 선택자와 후손 선택자

(예) `.navbar-default .navbar-nav > a { background-color:orange; }`

CSS3 선택자

기본 선택자

종류	형태
전체 선택자	*
태그 선택자	태그
아이디 선택자	#아이디
클래스 선택자	.클래스
속성 선택자	선택자[속성=값]
반응 선택자 (링크 선택자)	선택자:active 선택자:hover 선택자:link 선택자:visited
상태 선택자	:checked :focus :enabled :disabled
구조 선택자	선택자:first-child 선택자:last-child 선택자:nth-child(수열)
가상 선택자 (전후 문자 선택자)	::before ::after
자식 선택자 후손 선택자	부모태그 > 자식태그 조상태그 후손태그

CSS3 속성

종류	속성
박스 속성	width와 height, margin과 padding 속성 테두리 속성 border-width, border-style, border-color
display 속성	키워드 none, block, inline
배경 속성	background-image background-size background-repeat background-attachment background-position background
글자 속성	font-size font-family font-style font-weight text-align line-height text-decoration
그림자 속성	text-shadow box-shadow

회원가입

*필수입력사항

회원 기본 정보

아이디*	<input type="text" value="6자 이상의 영문과 숫자"/> ID중복확인		
비밀번호*	<input type="text"/> 8~12자리의 영문 대소문자와 숫자, 특수문자만 입력		
비밀번호확인*	<input type="text"/>		
이름*	<input type="text"/>		
닉네임	<input type="text"/> 2~8자리의 한글만 입력		
이메일*	<input type="text"/>	@ <input type="text"/>	직접 옵션 선택입력 ▼
주민번호*	<input type="text"/> - <input type="text"/>	주민번호 체크 버튼	
핸드폰번호*	<input type="text"/> 예)010-1234-4567		
성별	<input checked="" type="radio"/> 남성 <input type="radio"/> 여성		
생년월일*	2007 ▼ 년 1 ▼ 월 1 ▼ 일		
직업*	직업선택 ▼		
주소*	<input type="text" value="우편번호"/> <input type="button" value="우편번호 찾기"/>		
	<input type="text" value="주소"/>		
	<input type="text" value="참고항목(도로명인 경우: 지역 동이름과 아파트 이름)"/>		<input type="text" value="상세주소(아파트 동호수 또는 빌딩이름 입력요!!)"/>
<div>회원가입 등록 다시쓰기</div>			

자바스크립트(JavaScript)의 특징

- 웹 문서의 클라이언트 스크립트를 작성하는 표준언어이다
- Ajax 기술과 jQuery 라이브러리 등의 출현으로 계속 발전하면서 현재는 HTML5의 표준 스크립트 언어로 지정되어 모든 웹 브라우저들에서 기본으로 지원한다.
- 객체 기반(object based)의 언어로 C/C++, java 등과 유사한 문법적 특성을 가진다. 특별히 DOM, BOM과 같은 웹 브라우저 및 HTML 문서와의 인터페이스를 위한 다양한 객체들을 제공한다.
- 사용자들과의 상호작용에 필요한 동적인 기능들을 구현한다.

자바스크립트 코드 구조

```
<!DOCTYPE html>
<html>
<head>
  <script>
    var obj1 = "CSS3";

    function welcome(obj) {
      document.write(obj);
    }
  </script>
</head>

<body>
  <h3> Welcome! &nbsp; HTML5 웹 프로그래밍 </h3>
  <hr>
  <ol>
    <li> Using HTML! + <script> welcome(obj1); </script> ! </li>
    <li> <script> welcome('Using Javascript!'); </script> </li>
  </ol>
</body>
</html>
```

자바스크립트 코드

자바스크립트 코드

일반적으로 전역 변수와 함수 정의 그리고 HTML 문서를 웹 브라우저로 로딩할 때 실행해야 하는 자바스크립트 코드는 <head> 부분에 작성하고, <body> 부분에는 함수 호출 등과 같은 실행 코드들을 작성한다.

자바스크립트 변수 선언 방식

var - 변수 재선언 가능

유연한 변수 선언이 가능하다고 하기엔 중복 선언의 위험이 더 크기 때문에 단점으로 여겨진다. ES6 이후에 이 문제를 보완하기 위해서 **let**과 **const**가 추가되었다.

let - 변수 재선언 불가능, 변수 재할당 가능

const - 변수 재선언 불가능, 변수 재할당 불가능

결론

재할당이 필요없는 경우 **const**를 사용해 불필요한 변수의 재사용을 방지하고, 재할당이 필요한 경우 **let**을 사용하는 것이 좋음.



자바스크립트 기본 용어

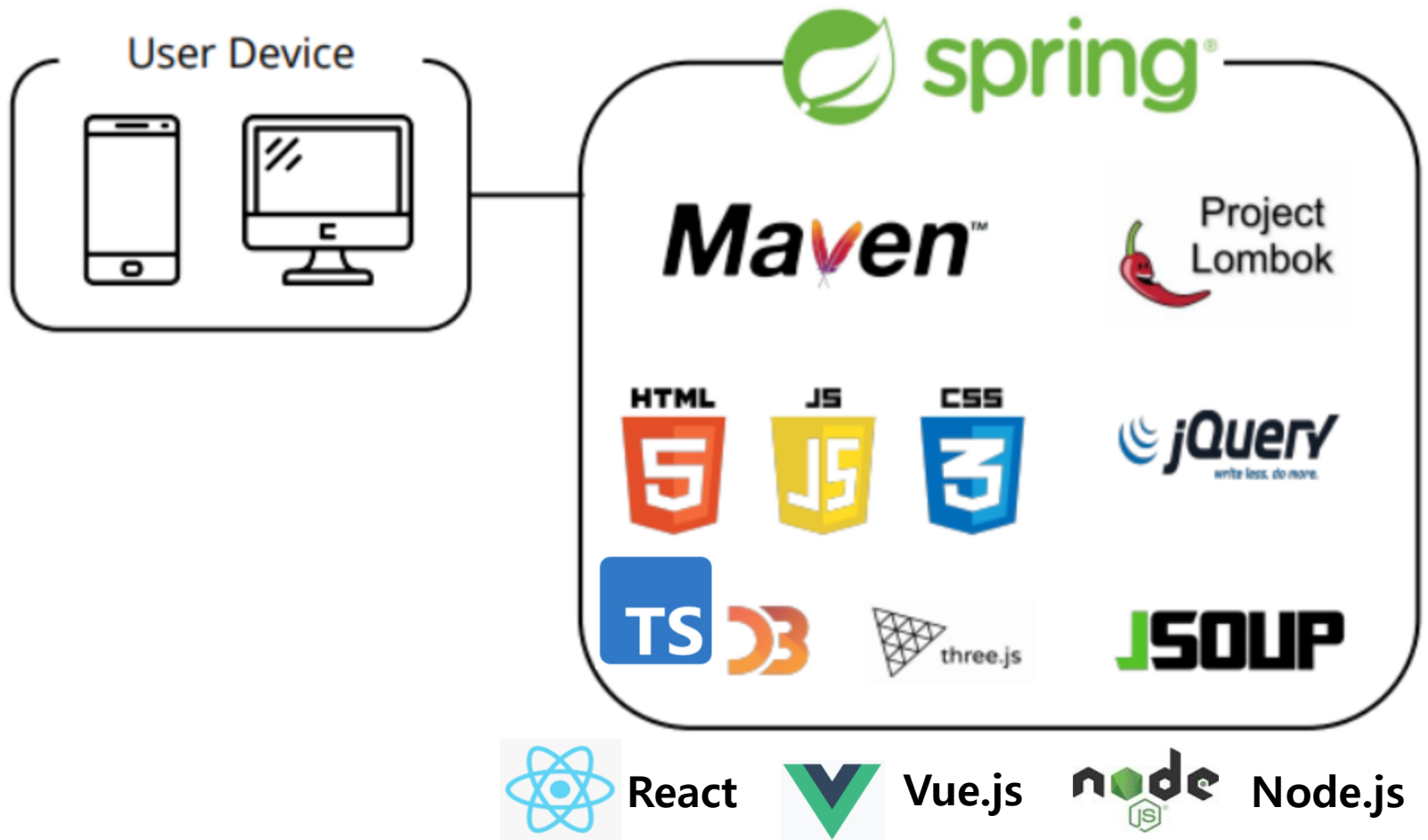
- 표현식과 문장
- 키워드
- 식별자
- 함수
- 속성
- 메소드



자바스크립트 객체

- 객체는 여러 개의 자료형을 한 번에 저장하는 자료형
- 배열은 인덱스를 기반으로 자료를 저장하지만
- 객체는 키를 기반으로 자료를 저장한다.

FRONT - END



- TypeScript는 활용도가 높아지는 웹 프론트엔드 언어이고 정적 타입의 컴파일 언어이며 타입스크립트 컴파일러 또는 바벨(Babel)을 통해 자바스크립트 코드로 변환된다. 코드 작성 단계에서 타입을 체크해 오류를 확인할 수 있고 미리 타입을 결정하기 때문에 실행 속도가 매우 빠르다는 장점이 있다.
- React는 자바스크립트 라이브러리의 하나로서 UI를 만들기 위해 사용한다.
- Vue는 컴포넌트 기반의 프레임워크이며, 오픈 소스 Progressive 자바스크립트 프레임워크
- Node.js는 Chrome V8 Javascript 엔진으로 빌드된 이벤트 기반 Javascript 런타임 환경이다. 내장 HTTP 서버 라이브러리를 포함하고 있다.

- 자바스크립트는 V8 엔진과 Node.js를 통해 많은 발전을 이루게 되었고 최근에는 프론트엔드, 백엔드, 모바일까지 완벽하게 지원함으로써 그 위상이 나날이 높아지고 있다.

자바스크립트의 자료형

자료형	변수의 자료형	구분
number	수치값(정수, 실수)	primitive type 또는 object
string	문자열	primitive type 또는 object
boolean	참, 거짓의 논리값	primitive type 또는 object
object	호출 불가능한 일반 객체	object
function	호출 가능한 함수 객체	object
undefined	자료형이 정의되지 않음	primitive type

[참고]

string 타입은 문자형 데이터를 말하는데, **작은따옴표**나 **큰따옴표**가 양쪽으로 감싸고 있는 데이터들을 모두 String 타입이라고 한다.

자바스크립트의 자료구조

- 배열 : 순서가 있는 컬렉션을 저장함
- 객체 : 키가 있는 컬렉션을 저장함

하지만 현실 세계를 반영하기엔 이 두 자료구조 만으론 부족해서
맵(Map)과 **셋(Set)**이 등장하게 되었다.

- Map : 키가 있는 데이터를 저장한다는 점에서 객체와 유사하다.
다만, 맵은 키에 다양한 자료형을 허용한다는 점에서 차이가 있다.
- Set : 중복을 허용하지 않는 값을 모아놓은 특별한 컬렉션이다.
셋에 키가 없는 값이 저장된다.
셋은 값의 유일무이함을 확인하는데 최적화되어 있다.

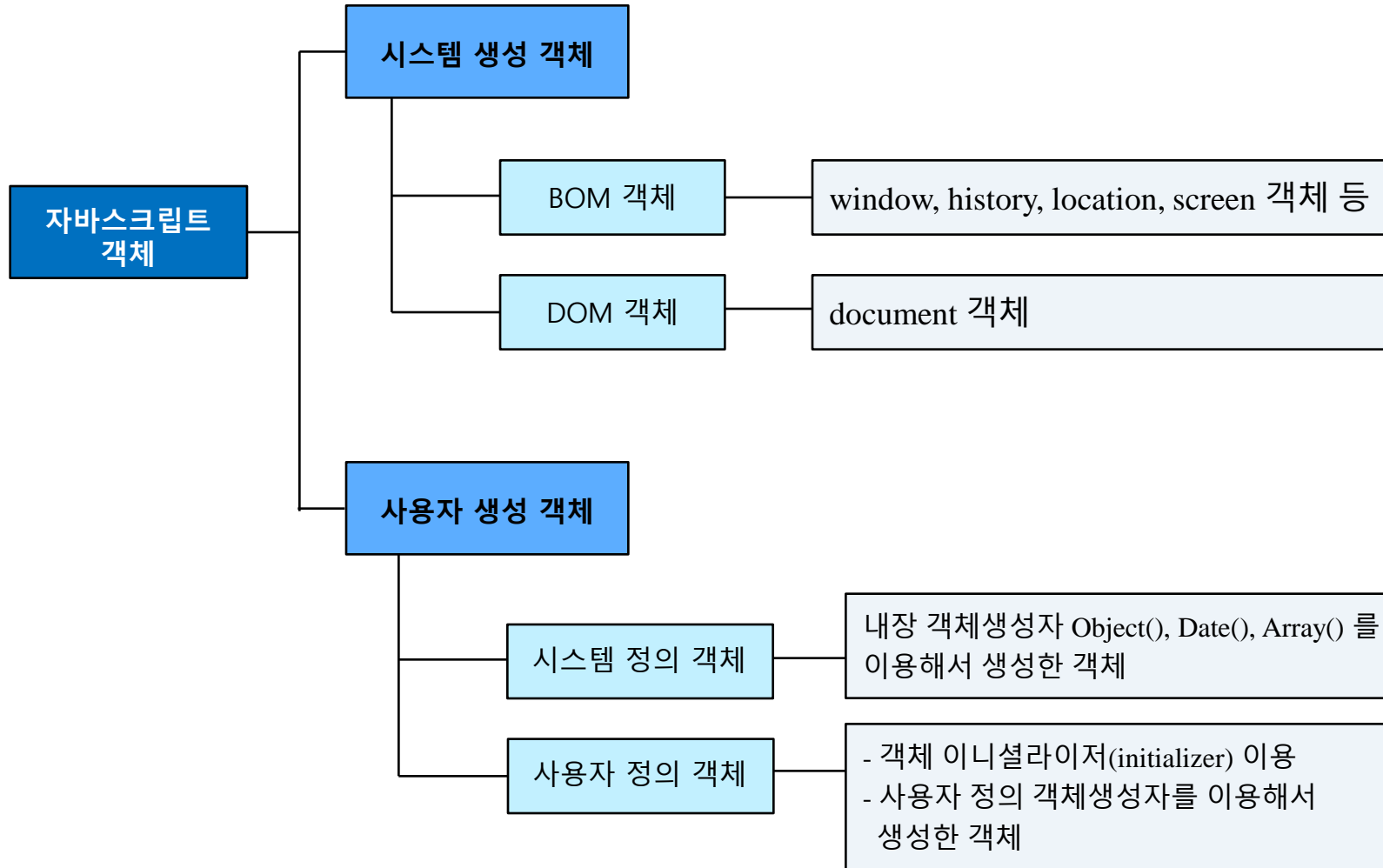
자바스크립트 배열 객체의 내장함수

내장함수	기능
concat()	- 인자로 주어진 배열이나 값들을 기존 배열에 합쳐 새 배열 반환
map()	- 배열 안의 각 원소를 변환할 때 사용되고, 이 과정에서 새로운 배열이 만들어짐 - 배열 내의 모든 요소 각각에 대하여 주어진 함수를 호출한 결과를 모아 새로운 배열을 반환
filter()	- 주어진 함수의 테스트를 통과하는 모든 요소를 모아 새로운 배열로 반환 - 배열에서 특정 조건을 만족하는 값들만 따로 추출하여 새로운 배열 생성
find()	- 주어진 판별 함수를 만족하는 첫 번째 요소의 값을 반환한다. 그런 요소가 없다면 undefined 를 반환
findIndex()	- 주어진 판별 함수를 만족하는 배열의 첫 번째 요소에 대한 인덱스를 반환합니다. 만족하는 요소가 없으면 -1을 반환한다.
indexOf()	- 배열에서 지정된 요소를 찾을 수 있는 첫 번째 인덱스를 반환하고 존재하지 않으면 -1을 반환
lastIndexOf()	- 배열에서 주어진 값을 발견할 수 있는 마지막 인덱스를 반환
join()	- 배열의 모든 요소를 연결해 하나의 문자열로 만든다.
pop()	- 배열에서 마지막 요소를 제거 하고 그 요소를 반환
push()	- 배열의 끝에 하나 이상의 요소를 추가 하고, 배열의 새로운 길이를 반환
unshift()	- 새로운 요소를 배열의 맨 앞쪽에 추가 하고, 새로운 길이를 반환

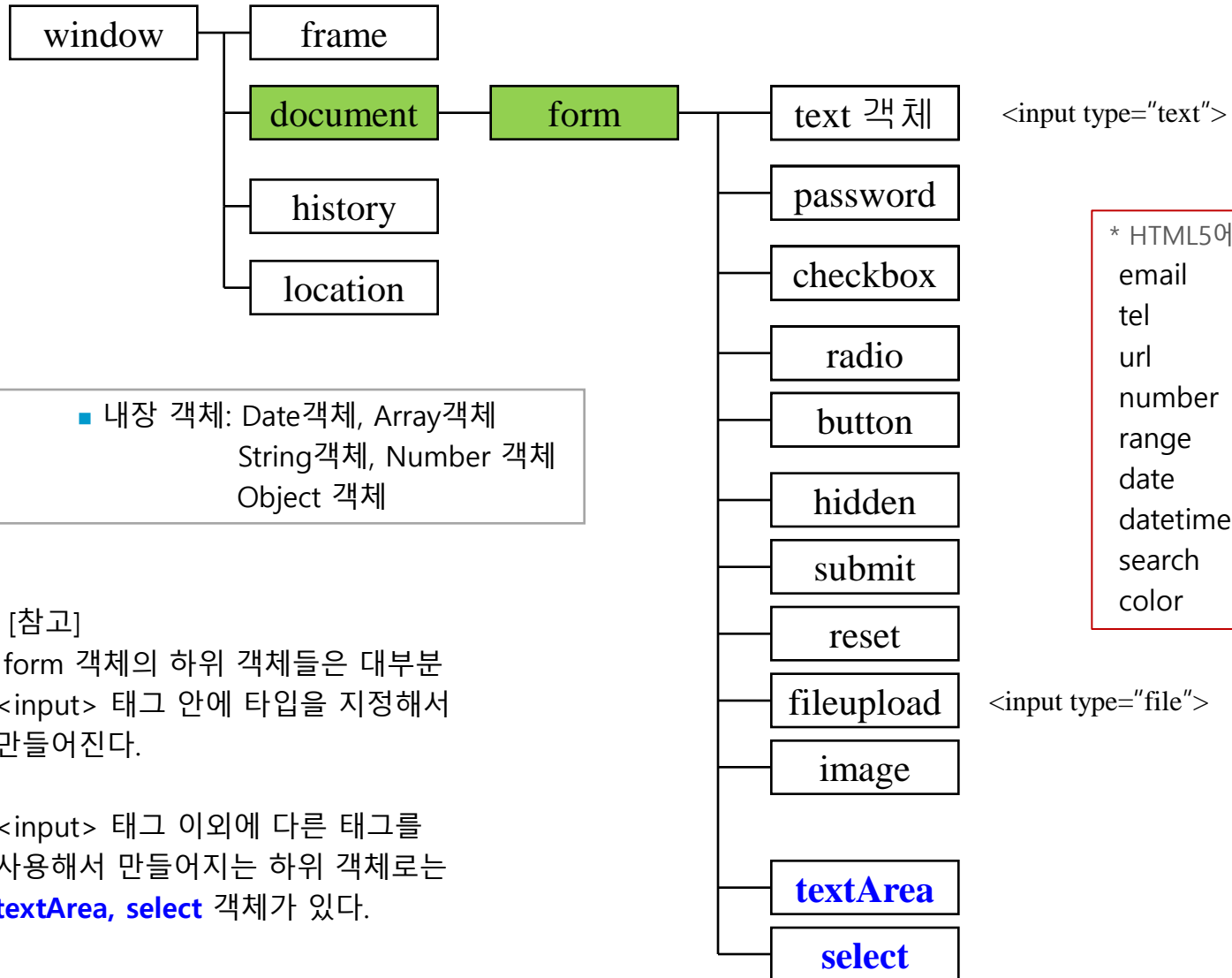
자바스크립트 배열 내장함수

내장함수	기능
reduce()	- 배열의 각 요소에 대해 주어진 리듀서(reducer) 함수를 실행하고, 하나의 결과 값을 반환
sort()	- 배열의 요소를 적절한 위치에 정렬한 후 그 배열을 반환
toString()	- 지정된 배열 및 그 요소를 나타내는 문자열을 반환
values()	- 배열의 각 인덱스에 대한 값을 가지는 새로운 Array Iterator 객체를 반환
slice()	- 배열 데이터를 새롭게 복제하여 반환하는 함수
reverse()	- 배열의 순서를 반대로 뒤집어주는 함수

자바스크립트 객체들



자바스크립트 객체들



window 객체의 중요 함수

함수	설명
alert()	경고용 대화상자를 만듦
confirm()	어떤 이벤트를 처리하기 전에 사용자로부터 확인을 요청받는 대화상자를 만든다. 확인, 취소 두 개의 버튼을 가짐
scrollTo(x, y)	창 안의 내용표시 영역을 지정된 위치로 이동한다.

window 객체의 타이머 함수

함수	설명
setInterval(함수, 시간)	특정한 시간마다 함수 실행
setTimeout(함수, 시간)	특정한 시간 후에 함수 실행
clearInterval(식별번호)	setInterval() 함수로 설정한 타이머 제거
clearTimeout(식별번호)	setTimeout() 함수로 설정한 타이머 제거

document 객체의 중요 함수

.getElementById(String elementId);

- id명으로 엘리먼트 찾기

.getElementsByName(String elementName);

- 태그의 이름(name) 속성값으로 엘리먼트 찾기

.createElement(String tagName);

- element를 만든다.

.createTextNode(String data);

- 선택한 요소에 텍스트를 추가

.appendChild(newNode);

- 선택한 요소 안에 자식 요소를 추가

.querySelector(String selectors);

- 선택자에 부합하는 요소 중에서 첫 번째 요소만을 반환

이벤트 핸들러(Event Handler)

- 이벤트 핸들러는 'on이벤트타입'과 같이 사용하여 이벤트를 다루는 것을 의미한다
- '어떠 어떠한 이벤트가 발생하면...' 이라는 뜻

이벤트핸들러 = "함수" or "명령줄"

이벤트 타입	이벤트 핸들러	설 명
Load	onLoad	문서가 브라우저에 의해 읽혀지는 순간 함수를 실행하고자 할 때 사용
Click	onClick	버튼이나 링크 등을 마우스로 클릭할 때
KeyUp	onKeyUp	키보드의 키를 눌렀다 떼는 순간
Input	onInput	사용자가 데이터를 입력 할 때마다 바로 바로 데이터를 확인 하고자 할 때 input , textarea 태그에서만 사용 가능
Change	onChange	텍스트, 텍스트 필드 등에서 포커스를 잃거나 내용이 수정될 때 input, textarea, select 태그에서만 사용 가능
Submit	onSubmit	폼의 입력 값을 전송하기 전에 어떤 작업을 할 수 있다. (예) '패스워드' 입력 값과 '패스워드확인' 입력 값이 일치하는지 확인할 때
MouseOver	onMouseOver	마우스 포인터를 특정 객체에 올렸을 때 발생하는 이벤트
MouseOut	onMouseOut	마우스 포인터를 특정 객체에서 다른 곳으로 옮겼을 때 발생하는 이벤트

자바스크립트 API 활용

■ 드래그 & 드롭 API

- 드래그 & 드롭 기능은 마우스를 이용하여 한 객체를 다른 객체 영역으로 끌어다 놓는 방식으로 두 객체 사이에 데이터를 전달하는 것으로, HTML5에서는 이 기능을 표준의 일부로 지원한다.

이벤트 타입	이벤트 핸들러	설 명
dragstart	ondragstart	사용자가 객체(object)를 드래그하려고 시작할 때 발생함. dataTransfer 객체에 setData로 데이터를 저장할 수 있다. 드래그 중에 css 속성이나 클래스를 추가할 수 있다.
drop	ondrop	드래그가 끝나서 드래그하던 객체를 놓는 장소에 위치한 객체에서 발생함. 이전의 ondragstart에서 설정해줬던 dataTransfer 객체의 데이터를 받아서 사용할 수 있다.
dragover	ondragover	드래그하면서 마우스가 대상 객체의 위에 자리 잡고 있을 때 발생함.

드래그 & 드롭 관련 API

요소/객체	메소드	의미
HTML 요소	draggable	true: 드래그 가능, false(기본값): 드래그 불가능
dataTransfer	setData(format, data)	전달할 데이터 설정, format: 데이터 유형, data: 전달 데이터
dataTransfer	getData(format)	전달되는 데이터 반환
event	preventDefault()	기본 이벤트의 발생을 중지시킴

자바스크립트 사용자 정의 객체 생성 => 객체 생성자 함수

```
function Item(title, price) {  
  //this = {};  
  //객체 속성 정의  
  this.title = title;  
  this.price = price;  
  
  //객체 메소드 정의  
  this.showPrice = function () {  
    console.log(`가격은 ${price}원 입니다.`);  
  };  
  //return this;  
}
```

```
const item1 = new Item("핸드폰", 1200000);  
const item2 = new Item("가방", 50000);  
const item3 = new Item("지갑", 200000);
```

```
console.log(item1, item2, item3);  
console.log(item1.showPrice());
```

문서 객체 선택

* 문서 객체 선택이란?

이미 존재하는 HTML 태그를 자바스크립트에서 문서 객체로 변환하는 것이다.

(1) 1개의 문서 객체를 선택하는 메소드

`document.getElementById(아이디)`

`document.querySelector(선택자)`

(2) 여러 개의 문서 객체를 선택하는 메소드

`document.querySelectorAll(선택자)`

`document.getElementsByClassName(클래스)`

위 메소드를 사용해 문서 객체를 선택하면 **배열의 형태**로 문서 객체가 반환 된다.

Bootstrap

- 부트스트랩은 웹사이트를 쉽게 만들 수 있게 도와주는 HTML, CSS, JS 프론트엔드 프레임워크이다.
- 부트스트랩도 일종의 웹 사이트 저작도로 (특히 반응형에 최적) HTML, CSS에 기반한 템플릿을 제공한다.
- 부트스트랩에서 반응형으로 사용할 HTML 요소들을 둘러싸는 기본 클래스로 **container**를 제공하고 있다.

<!-- Bootstrap을 적용하고자 할 때 아래 한 줄 추가요!! -->

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
```

jQuery

- jQuery는 javascript 함수들로 이루어진 라이브러리이다.
- jQuery는 Javascript 프로그래밍을 매우 단순하게 해준다.
- jQuery 라이브러리는 javascript 파일로 저장되어 있고, 모든 jQuery 메서드를 담고 있다.
- jQuery 라이브러리는 다음과 같은 특징이 있다.
 - HTML 엘리먼트 선택하고
 - HTML 엘리먼트 조작(즉, 그 엘리먼트에 "액션" 수행)
 - CSS 조작
 - HTML 이벤트 함수
 - javascript 효과 및 애니메이션

<!-- jQuery 인클루드 -->

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
```

<!-- jQuery 인클루드 -->

```
<script src="js/jquery.js"></script>
```

<!-- jQuery 플러그인 인클루드 -->

```
<script src="media/js/jquery.dataTables.js"></script>
```


jQuery 문법

- jQuery 문법은 HTML 엘리먼트를 원하는대로 선택하고 그 엘리먼트에 어떤 "액션"을 수행하게 만든다.
- 기본 문법은

```
$(selector).action();
```

- \$표시는 jQuery라는 것을 정의하고
- (selector)는 "질의하거나 찾을" HTML 엘리먼트를 의미하며
- jQuery action()은 이 엘리먼트가 수행할 액션이다.

Document Ready Function

- 모든 jQuery 메소드가 document.ready() 함수 안에 존재한다.

```
$(document).ready(function() {  
    // jQuery 함수들은 여기에  
    ...  
    ...  
});
```

- jQuery 메소드가 document.ready() 함수안에 넣는 이유는 => jQuery 코드가 문서가 로딩이 끝나기 전 (혹은 문서가 준비되기 전)에 실행되는 것을 막기 위한 것이다.

jQuery Selectors

- jQuery selectors는 여러개 또는 한 개의 HTML 엘리먼트를 선택하고 조작할 수 있게 한다.
키포인트는 jQuery를 이용하여 효과를 주기 원하는 엘리먼트를 정확하게 선택하는 방법을 배우는 것이다.
- jQuery selector는 엘리먼트 이름, 특성이름, 또는 content로 HTML 엘리먼트를 선택할 수 있게 한다.

[jQuery Element Selectors 예]

`$("p")`는 모든 `<p>` 엘리먼트를 선택한다.

`$(".p.intro")`는 `class="intro"`인 모든 `<p>` 엘리먼트를 선택한다.

`$("#p#demo")`는 `id="demo"`인 모든 `<p>` 엘리먼트를 선택한다.

jQuery 중요 메소드

메소드	기능
.append()	- 선택한 요소의 내용의 끝에 콘텐츠를 추가하는 메소드
.html()	- 선택한 요소 안의 내용을 가져오거나, 다른 내용으로 바꾸는 메서드로 .text()와 비슷하지만 태그의 처리가 다르다.
.on()	- 특정 요소에 이벤트 바인딩(event binding)하기 위해 사용하는 메소드
.prop()	- 속성값을 가져오거나 추가하는 메서드
.text()	- 선택한 요소 안의 내용을 가져오거나, 다른 내용으로 바꾸는 메서드
.parent()	- 어떤 요소의 부모 요소를 선택하는 메소드
.find()	- 어떤 요소의 하위 요소 중 특정 요소를 찾는 메서드
.each()	- 선택한 요소가 여러 개일 때 각각에 대하여 반복하여 함수를 실행
.attr()	- 속성(attribute)의 값을 가져오거나 속성을 추가하는 메서드
.val()	- 양식(form)의 값을 가져오거나 값을 설정하는 메소드
.trigger()	- 강제로 이벤트를 발생시키기 위한 메소드

jQuery 플러그인(plug in)

- 개발자가 jQuery 자체를 확장하여 사용할 수 있도록 제공하는 jQuery 라이브러리의 한 형태이다.
- jQuery의 다양한 기능을 쉽게 사용할 수 있도록 제공해 주는 라이브러리이다.

(예)
dataTables
rotator

jQuery UI

- 인터랙티브(interactive)한 웹을 제공할 수 있도록 하는 인터페이스 개발 도구이다.
- jQuery는 사이트의 사용자 인터페이스(UI)의 개선을 위해 다양한 jQuery UI를 제공해 준다.

(예)
autocomplete
datepicker
tabs

CSV 예시

```
no, name, score, hobby
1, 김정은, 95, (게임, 농구)
2, 오수철, 100, (수영, 축구)
```

XML 예시

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <user>
    <name>김정오</name>
    <score>95</score>
    <hobby>
      <element>게임</element>
      <element>농구</element>
    </hobby>
  </user>

  <user>
    <name>오수철</name>
    <score>100</score>
    <hobby>
      <element>수영</element>
      <element>축구</element>
    </hobby>
  </user>
</users>
```

JSON 예시

```
{
  "users": {
    "1": {
      "name": " 김정은",
      "score": 95,
      "hobby": [ " 게임", " 농구"]
    },
    "2": {
      "name": " 오수철",
      "score": 100,
      "hobby": [ " 수영", " 축구"]
    },
  }
}
```

● 데이터 전송 형식

서버와 클라이언트가 데이터를 주고 받을 때 특정한 형식을 맞춰야 한다.

서버와 클라이언트가 통신할 때 자주 사용하는 파일 형식인

- CSV(Comma Separated Values) : 각 항목을 쉼표(,)로 구분해 표현하는 방법
- XML : HTML 형식처럼 태그로 데이터를 표현하는 방법
- JSON(Javascript Object Notation) : CSV 형식과 XML 형식의 단점을 극복한 형식이다.
자바스크립트에서 사용하는 객체 형태로 데이터를
표현하는 방법

* 각 형식들의 장단점

CSV : 장점(용량이 적다) 단점(가독성이 떨어진다)

- 간단한 테이블 작성 또는 읽는 속도가 중요한 부분에서 사용

XML : 장점(가독성이 좋다) 단점(용량이 크다)

- 단순 게임 옵션, 직접 데이터 수정이 잦은 부분에 사용

JSON : 장점(용량이 적다, 가독성이 좋다)

- 서버 통신 REST API를 사용할 때 가장 많이 사용.

Ajax

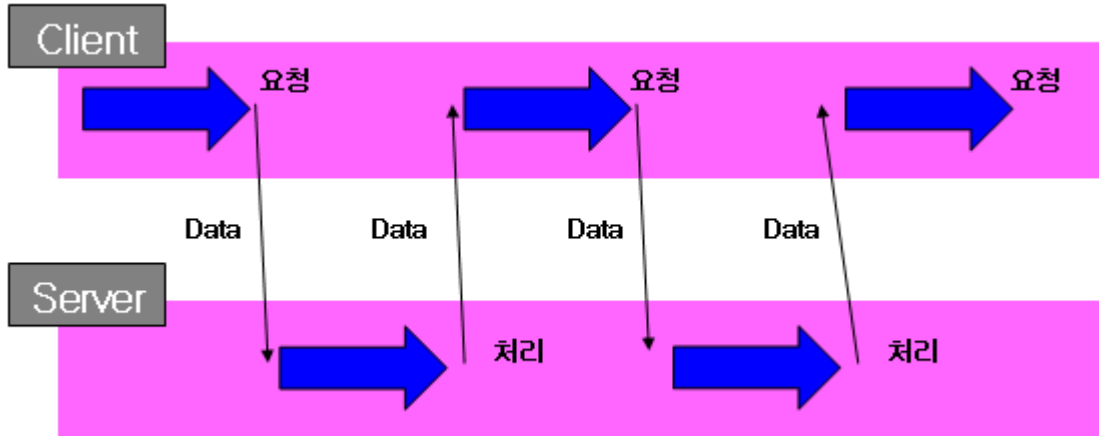
- Asynchronous Javascript and XML의 약자로 **‘구현 방식’**이다.
- Ajax를 사용하면 페이지를 전환하지 않고 서버에서 데이터를 받아와 사용자에게 보여줄 수 있다. (예) se.naver.com
- **비동기로 Javascript의 비동기 통신 기술을 이용해서 XML 데이터를 주고 받는 기술이다**
- 대화식 웹 애플리케이션의 제작을 위해 아래와 같은 조합을 이용하는 웹 개발 기법이다.
 - 표현 정보를 위한 HTML (또는 XHTML) 과 CSS
 - 동적인 화면 출력 및 표시 정보와의 상호작용을 위한 DOM, Javascript
 - 웹 서버와 비동기적으로 데이터를 교환하고 조작하기 위한 XML, XSLT, XMLHttpRequest



Ajax의 장점

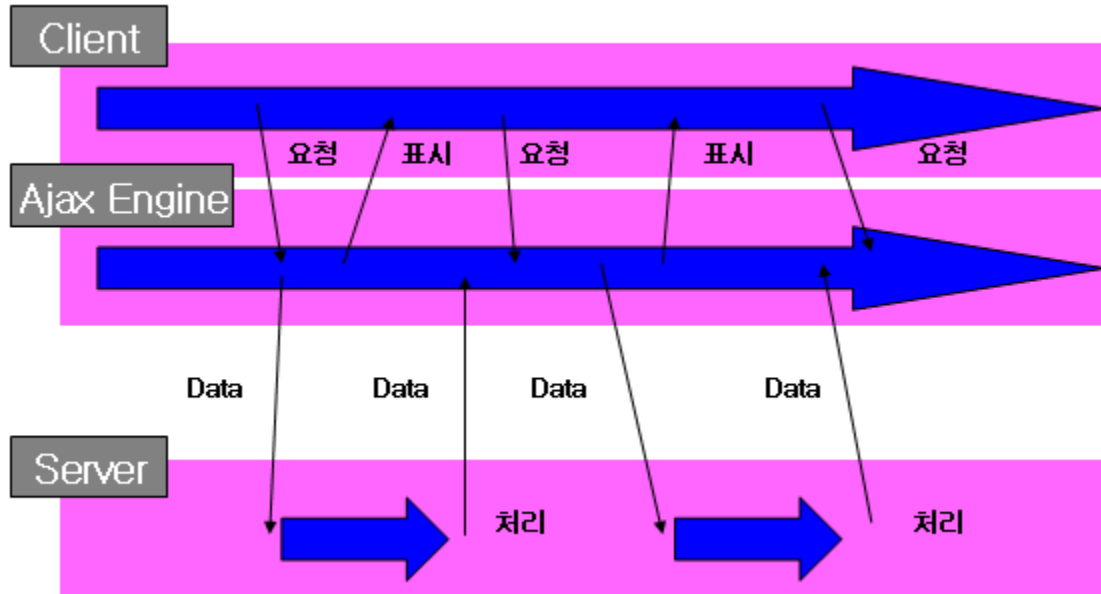
- 페이지 이동없이 고속으로 화면을 전환할 수 있다.
- 서버 처리를 기다리지 않고, 비동기 요청이 가능하다.
- 수신하는 데이터 양을 줄일 수 있고, 클라이언트에게 처리를 위임할 수도 있다.

■ 동기 통신 개념



- 클라이언트에서 사용자가 행동함에 따라 데이터가 서버로 전송되고
- 서버가 이를 받아 처리한 후 클라이언트로 보내면, 클라이언트가 이를
- 받아서 다시 처리하는 순서로 통신하게 된다.
- **서버의 처리를 기다리는 순간 동안에 사용자는 동작하지 못한다.**

■ 비동기 통신 개념

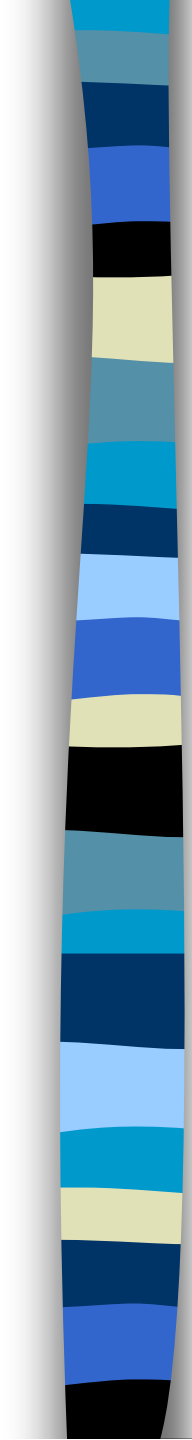


- 동기 통신과의 차이점은 동기 통신은 사용자의 화살표가 중간 중간에 끊기는 반면 비동기 통신은 끊기지 않고 계속해서 이어져 있다는 것이다.
위의 그림이 의미하는 것은 사용자가 일일이 새로운 페이지를 로드하는 것이 아니라 **중간에 Ajax 엔진이 백그라운드에서 새로운 데이터를 요청하고 받는 역할을 하게 된다는 것이다.**
결론은 XMLHttpRequest를 이용하면 페이지가 모두 로드된 이후에도 데이터를 요청할 수 있는 기능을 구현 할 수 있다는 것이다.

Ajax의 4단계 진행과정

1. XMLHttpRequest Object를 만든다.
 - request를 보낼 준비를 브라우저에게 시키는 과정
 - 이것을 위해서 필요한 method를 갖춘 object가 필요함
2. callback 함수를 만든다.
 - 서버에서 response가 왔을 때 실행시키는 함수
 - HTML 페이지를 업데이트 함
3. Open a request
 - 서버에서 response가 왔을 때 실행시키는 함수
 - HTML 페이지를 업데이트 함
4. send the request

```
function myGet(url, callback) {  
    const xhr = new XMLHttpRequest();  
    xhr.onreadystatechange = function (event) {  
        if (this.readyState === 4 && this.status === 200) {  
            callback(this.responseText, true, this);  
        }  
    };  
    xhr.open("GET", url, true);  
    xhr.send();  
}
```



Ajax가 xhr객체를 형성하고 이 객체의 콜백을 만들고
HTML 메소드와 URL을 결정한 뒤, xhr객체의 메소드로
정보를 보내는 방식이다.

- **var xhr= new XMLHttpRequest();** : browser response를 얻었을 때 작동하는 함수
(callback 함수)
- **xhr.onreadystatechange** : AJAX Request에 어떠한 변화라도 있으면 작동한다. (callback
함수를 포함하고 있다고 생각하면 된다.)
- **xhr.readyState** : response가 돌아왔는지 아닌지를 추적하는 property

XMLHttpRequest 클래스의 중요 메소드 및 이벤트 리스너

- XMLHttpRequest는 자바스크립트가 **Ajax**를 사용할 때 사용하는 객체이다.

open(Http Request Method, URL, async)	Ajax 요청을 초기화하면서 요청방식과 주소 및 동기화 여부를 지정
send()	작성된 Ajax 요청을 서버로 전달
onload 이벤트 리스너	요청에 대한 응답을 받는 이벤트 리스너