

프론트 개발을 위한 웹퍼블리싱

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.

HTML5와 웹 표준

HTML CSS JavaScript



- HTML5로는 웹의 내용/골격(웹 문서의 문법)을 작성하고
- CSS3로는 웹을 디자인(HTML 문서를 표현)하며
- JavaScript로는 웹의 동작/기능(사용자 반응)을 구현한다.

HTML5 페이지 기본 구조

```
<!DOCTYPE html>
<html>
<head>

</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
link	웹 페이지에 다른 파일 추가
style	웹 페이지에 스타일시트 추가
base	웹 페이지의 기본 경로 지정

HTML5 기본 태그

1. 글자 태그

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

2. 목록 태그

- ul, ol, li

3. 테이블 태그

- table, thead, tbody, tfoot, tr, th, td

4. 이미지 태그

- img

5. 공간 분할 태그

- 기본 공간 분할 태그 : div, span
- 시맨틱 태그 : header, nav, aside, section, article, footer

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 같은 독립적인 콘텐츠를 표시한다.
section	문서 내용 구분, 즉 일반적인 문서나 애플리케이션 영역을 표시한다. 섹션의 제목을 나타내는 h1 ~ h6 태그 그리고 <p> 태그와 함께 사용할 수 있다.
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">

<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

CSS3 선택자

기본 선택자

종류	형태
전체 선택자	*
태그 선택자	태그
아이디 선택자	#아이디
클래스 선택자	.클래스
속성 선택자	선택자[속성=값]
반응 선택자	선택자:active 선택자:hover
상태 선택자	:checked :focus :enabled :disabled
구조 선택자	선택자:first-child 선택자:last-child 선택자:nth-child(수열)
문자 선택자	::before ::after

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



자바스크립트 기본 용어

- 표현식과 문장
- 키워드
- 식별자
- 함수
- 속성
- 메소드



자바스크립트 자료형

- 숫자 – 정수와 실수를 구분하지 않음
- 문자열 – 문자의 집합
- 불(boolean) – 결과를 참과 거짓으로 표현
- 함수 – 코드의 집합



자바스크립트 객체

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

FRONT - END

User Device



Maven[™]



React



Vue.js



Node.js

- **React**는 자바스크립트 라이브러리의 하나로서 UI를 만들기 위해 사용한다.
- **Vue**는 컴포넌트 기반의 프레임워크이며, 오픈 소스 Progressive 자바스크립트 프레임워크
- **Node.js**는 Chrome V8 Javascript 엔진으로 빌드된 이벤트 기반 Javascript 런타임이다.
내장 HTTP 서버 라이브러리를 포함하고 있다.

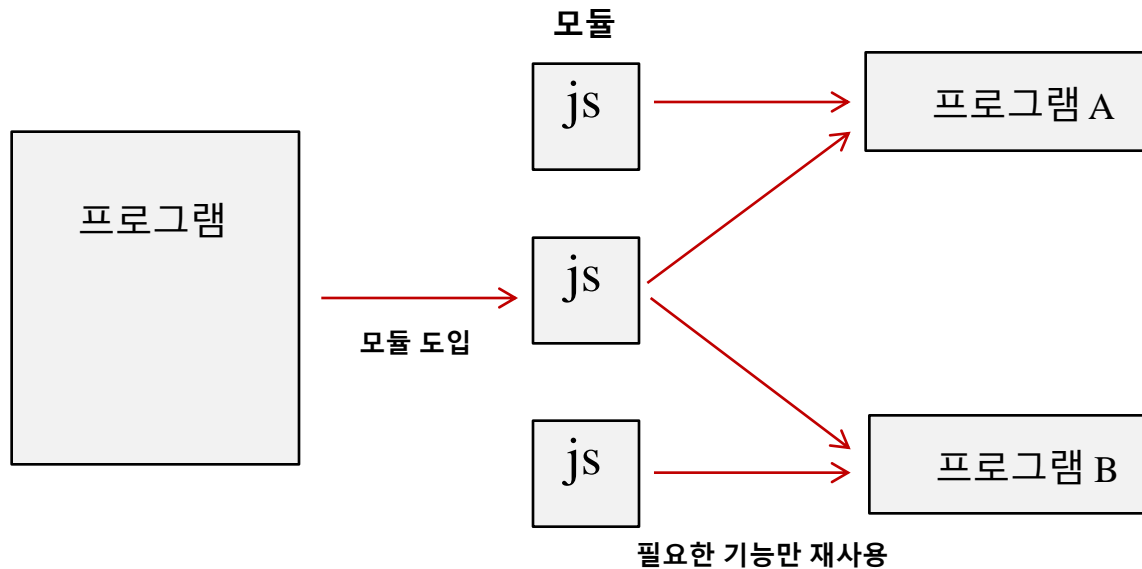
- **자바스크립트**는 V8 엔진과 Node.js를 통해 많은 발전을 이루게 되었고
최근에는 프론트엔드, 백엔드, 모바일까지 완벽하게 지원함으로써
그 위상이 나날이 높아지고 있다.



Node, VS Code 설치

- <https://nodejs.org/>
- <https://code.visualstudio.com/>

Node에서 모듈과 프로그램



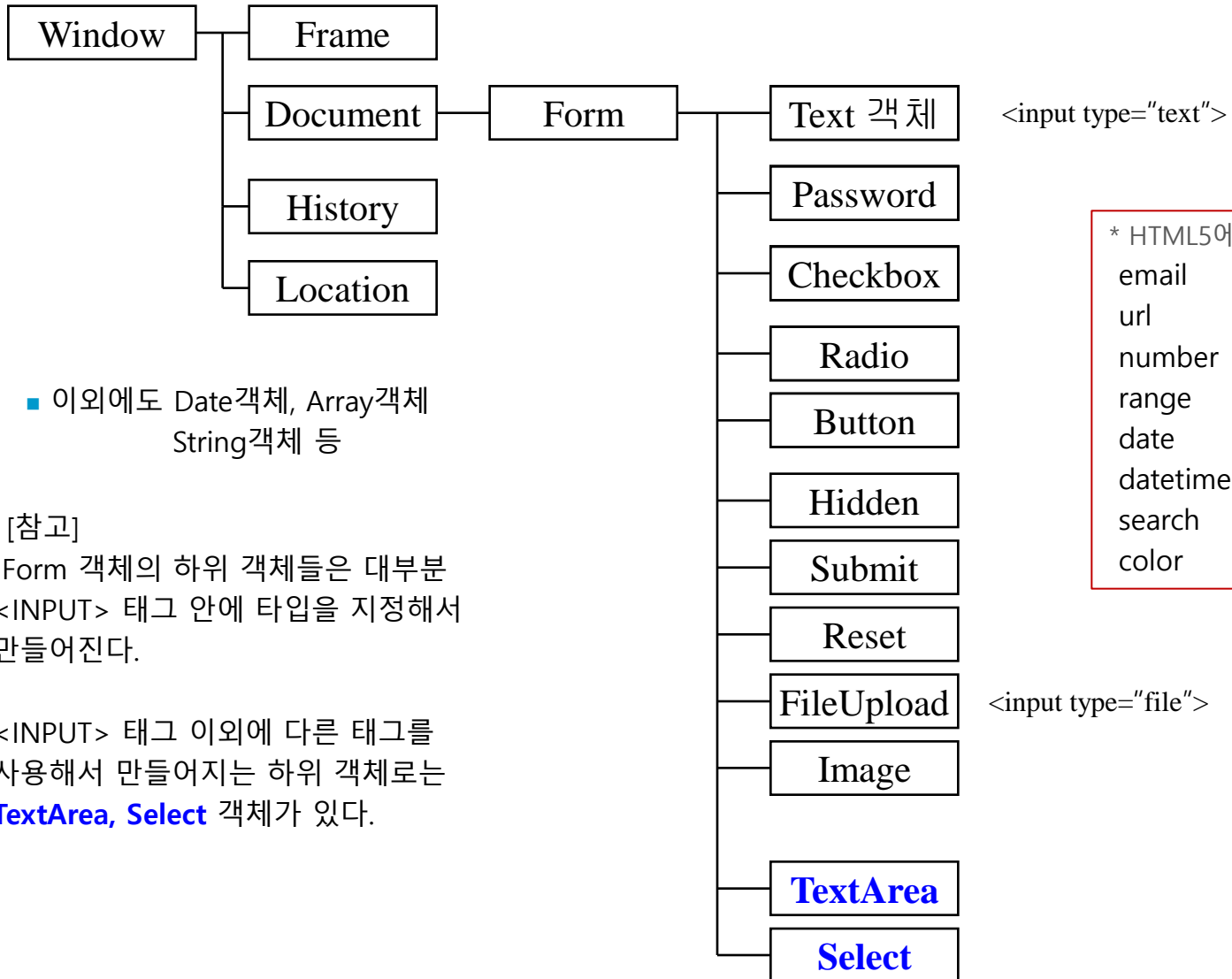
- 모듈 하나가 여러 개의 모듈을 사용할 수 있고,
모듈 하나가 여러 개의 모듈에 사용될 수도 있다.

자바스크립트의 타이머 함수

- 자바스크립트의 타이머 함수를 사용하면 특정 시간마다 코드를 실행시킬 수 있다.

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

자바스크립트 객체들



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
Change	onChange	텍스트, 텍스트 필드 등에서 포커스를 잃거나 내용이 수정될 때
Submit	onSubmit	Submit 버튼을 눌러 Form의 내용을 제출할 때
Focus	onFocus	창이나 텍스트 박스 등을 선택하여 포커스가 주어질 때
MouseOver	onMouseOver	마우스 포인터를 특정 객체에 올렸을 때 발생하는 이벤트
MouseOut	onMouseOut	마우스 포인터를 특정 객체에서 다른 곳으로 옮겼을 때 발생하는 이벤트

반응형 웹

- 반응형 웹은 웹 페이지 하나로도 데스크톱, 스마트폰, 태블릿PC에 맞게 디자인이 자동으로 반응해서 변경되는 웹 페이지를 의미한다.
그러므로 개발 효율성은 물론 유지 보수적 측면에서 큰 장점이 있다.
- 이러한 반응형 웹 페이지는 **미디어 쿼리(Media Query)**를 사용해 개발한다.
현대 웹 브라우저는 '미디어 쿼리(Media Query)' 기능을 지원한다.
- 미디어 쿼리는 미디어 타입에 따라 CSS를 적용할 수 있는 기능이다.
미디어 타입(Media Type)은 장치(Device)를 의미하는데,
프린터, 데스크톱, 스마트폰, 태블릿PC, 시청각 장애인 등이 사용하는
특별한 웹 브라우저를 나타낸다. 따라서 미디어 쿼리를 사용하면
프린터, 데스크톱, 스마트폰, 태블릿PC, 시청각 장애인용 웹 사이트를 구축할 수 있다.

미디어 쿼리

- 미디어 쿼리는 두 가지 방법으로 사용한다.

@media 규칙

media 속성

(1) @media 규칙

규칙은 스타일시트 내부에서 특정한 규칙을 표현하는데 사용하는 @로 시작한다.

외부 스타일을 가져오는 @import 규칙,

글꼴을 추가로 정의하는 @font-face 규칙 등이 있다.

최근 HTML 페이지가 다양한 장치에서 실행되면서 부각된 규칙이다.

```
@media (<미디어 쿼리>) {
```

```
<CSS 코드>
```

```
}
```

<미디어 쿼리> 부분에 모니터를 나타내는 'screen'과

프린터를 나타내는 'print' 미디어 쿼리를 사용한다.

(2) media 속성

```
<link rel="stylesheet" ref="<파일이름>" media="<미디어 쿼리>" />
```

(예)

```
<link rel="stylesheet" ref="screen.css" media="screen" />
```

```
<link rel="stylesheet" ref="print.css" media="print" />
```

문서 객체 모델(DOM)

- 웹 브라우저에 HTML 파일을 끌어다 놓으면 HTML 파일을 분석하고 표시한다. 이때 웹 브라우저가 HTML 파일을 분석하고 표시하는 방식을 **문서 객체 모델 (DOM, Document Object Model)**이라 한다.
- HTML 태그를 자바스크립트에서 이용할 수 있게 객체로 만든 것을 문서 객체라 한다. 따라서 자바스크립트에서 문서 객체를 사용하면 HTML 태그를 생성, 제거하거나 조작할 수 있다. 또한 문서 객체를 사용해 마우스를 클릭하거나 키보드를 누르는 등 이벤트에 반응할 수 있다.

문서 객체 선택

* **문서 객체 선택**이란?

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.

웹 페이지 생성 순서

- 웹 브라우저가 웹 페이지를 실행하는 순서는 HTML 코드를 위쪽에서 아래쪽으로 실행한다.

```
<!DOCTYPE html>
<html>
<head>
  <title>Document Object Model</title>
  <script>
    // h1 태그의 배경 색상을 변경한다.
    document.querySelector('h1').style.backgroundColor = 'red';

    // h2 태그의 글자 색상을 변경한다.
    document.querySelector('h2').style.color = 'red';
  </script>
</head>
<body>
  <h1>Process - 1</h1>
  <h2>Process - 2</h2>
</body>
</html>
```

위 코드는 **문서 객체 사용 오류**가 발생한다.

이유는 script 태그를 읽는 시점에 h1 과 h2 태그는 생성되지 않았기 때문이다.

즉 HTML 페이지의 생성 순서는 아주 중요하다.

해결 방법

■ 이벤트 활용하기

```
<!DOCTYPE html>
<html>
<head>
  <title>Document Object Model</title>
  <script>
    //이벤트 연결
    window.onload = function () {
      // 문서 객체를 선택한 후 문서 객체를 조작한다.
      document.querySelector('h1').style.backgroundColor = 'red';
      document.querySelector('h2').style.color = 'red';
    };
  </script>
</head>
<body>
  <h1>Process - 1</h1>
  <h2>Process - 2</h2>
</body>
</html>
```