Javascript

자바스크립트란 Ecma International의 프로토타입 기반의 프로그래밍 언어로 오늘날 HTML, CSS와 함께

웹을 구성하는 요소 중 하나다. HML이 웹 페이지의 기본 구조를 담당하고, CSS가 디자인을 담당한다면,

Javascript는 클라이언트 단에서 웹 페이지가 동작하는 것을 담당한다.

DOM Manipulation (DOM 조작)

오늘날 Javascript가 가장 널리 쓰이는 분야는 클라이언트용 인터페이스이다. 이 때 주로 웹 브라우저에서

제공되는 DOM API를 사용한다. HTML 문서로 구성된 웹 사이트는 정적이어서 변하지 않는다.

DOM은 정적 웹 사이트를 기능적으로 작동시키기 위해 브라우저에 의해 구현된 프로그래밍 인터페이스이다.

DOM API를 사용하면 문서의 구조, 스타일, 그리고 내용을 변경할 수 있다. 이를 바탕으로 보다 쉽게 동적인

웹사이트 개발을 위해 수많은 프론트엔드 프레임워크가 개발되었다.

EventListener

이벤트 리스너는 DOM 객체에서 이벤트가 발생할 경우 해당 이벤트 처리 핸들러를 추가할 수 있는 오브젝트다.

이벤트 리스너를 이용하면 특정 DOM에 Javascript 이벤트가 발생할 때 특정 함수를 호출한다.

특정 DOM 요소에 이벤트 리스너를 등록할 때는 addEventListener를 사용한다.

DOM객체.addEventListener(이벤트명, 실행할 함수명, 옵션)

이벤트명: Javascript에서 발생할 수 있는 이벤트 명을 입력한다.

함수명: 해당 변수는 생략 가능하며, 해당 이벤트가 발생할 때 실행할 함수명을 입력한다.

옵션: 옵션은 생략이 가능하며, 자식과 부모 요소에서 발생하는 버블링을 제어하기 위한 옵션이다.

Onclick과의 비교

Onclick은 클릭 시 자바스크립트를 수행하는 이벤트이다.

Onclick은 이벤트를 여러 개 적용하는 것이 불가능하다. Onclick 이벤트 핸들러를 이미 사용한 상태에서

새로운 onclick 이벤트를 추가한다면 이후에 추가된 이벤트가 기존의 이벤트를 덮어쓴다.

addEventListener를 사용할 경우 여러 이벤트를 추가하더라도 누적되어 모든 이벤트가 동작된다.

Onclick 이벤트는 모든 브라우저에서 호환이 가능하다.

addEventListener는 IE 6,7,8 버전에서는 호환되지 않는다. 만일 오래된 브라우저를 지원해야 할 경우

onclick 이벤트를 사용하는 것이 좋다.

addEventListener는 세 번째 파라미터로 이벤트가 발생할 때 버블링으로 작동될 지 캡쳐링으로

작동될 지 지정할 수 있다.

세 번째 파라미터가 TRUE일 경우 캡쳐링, FALSE일 경우 버블링을 사용한다.

Event loop

Javascript 엔진은 힙과 콜 스택으로 구성되어 있다. 가장 유명한 것은 구글의 V8 엔진이다.

Javascript는 단일 스레드 프로그래밍 언어이기 때문에 콜 스택이 하나다.

Javascript 엔진 밖에 있는 Web API는 브라우저에서 제공하는 API로, DOM, Ajax, Timeout 등이 있다.

콜 스택에서 실행된 비동기 함수는 웹 API를 호출하고 웹 API는 콜백함수를 Callback Queue에 넣는다.

Callback Queue는 비동기적으로 실행된 콜백함수가 보관되는 영역이다.

이벤트 루프는 콜 스택과 콜백 큐의 상태를 체크하여, 콜 스택이 빈 상태가 되면, 콜백 큐의 첫 번째

콜백을 큐 스택에 넣는다. 이러한 반복적인 행동을 틱(tick)이라고 한다.

이러한 이유로 단일 스레드 프로그래밍 언어임에도 멀티 스레드처럼 보인다.

getElementByID

태그에 있는 id 속성을 사용하여 해당 태그에 접근하여 하고 싶은 작업을 할 때 쓰는 함수이다.

해당 id가 없는 경우 null 에러가 발생한다.

getElementsByClassName

태그에 있는 class 속성을 사용하여 해당 태그에 접근하여 하고 싶은 작업을 할 때 쓰는 함수이다.

동일한 class명이 존재할 수 있기 때문에 id 속성을 사용하여 접근하는 getElementById와 달리

컬렉션 객체를 반환한다.

queryselector vs queryselectorall

querySelector는 지정된 선택자와 일치하는 도큐먼트의 첫 번째 element를 반환한다.

일치하는 요소가 없으면 null을 반환한다.

querySelectorAll는 지정된 셀렉터 그룹에 일치하는 도큐먼트의 element list를 나타낸다.

즉 NodeList를 반환한다. 지정된 셀렉터가 없는 경우 비어있는 NodeList로 반환된다.

Node

HTML DOM은 노드라고 불리는 계층적 단위에 정보를 저장한다. HTML DOM은 노드들을 정의하고,

그 사이의 관계를 설명해주는 역할을 한다. HTML 문서의 정보는 노드 트리라고 불리는 계층적 구조에 저장된다. 노드 트리는 노드들의 집합이며, 노드 간의 관계를 보여준다. 노드 트리는 최상위 레벨인 루트 노드로부터 시작하여, 가장 낮은 레벨인 텍스트 노드까지 내려간다. HTML DOM을 이용하여 노드 트리에 포함된 모든 노드에 접근할 수 있다.

노드 종류

문서 노드(document node): 문서 전체를 나타내는 노드

요소 노드(element node): HTML 요소는 요소 노드이며, 속성 노드를 가질 수 있는 유일한 노드이다.

속성 노드(attribute node): HTML 요소의 속성은 속성 노드이며, 요소 노드에 관한 정보를 가지고 있다.

하지만 해당 요소 노드의 자식 노드에는 포함되지 않는다.

텍스트 노드(text node): HTML 문서의 모든 텍스트는 텍스트 노드이다.

주석 노드(comment node): HTML 문서의 모든 주석은 주석 노드이다.

노드 트리의 가장 상위에는 단 하나의 루트 노드가 존재한다. 루트 노드를 제외한 모든 노드는 단 하나의 부모 노드만을 가진다. 모든 요소 노드는 자식 노드를 가질 수 있다. 형제 노드란 같은 부모 노드를 가진 모든 노드를 가리킨다. 조상 노드란 부모 노드를 포함해 계층적으로 현재 노드보다 상위에 존재하는 모든 노드를 가리킨다. 자손 노드란 자식 노드를 포함해 계층적으로 현재 노드보다 하위에 존재하는 모든 노드를 가리킨다.