11.1 이벤트의 종류

- 마우스, 키보드, HTML프레임, 입력 양식, 유저인터페이스, 구조변화, 터치

11.2 이벤트 관련 용어 정리

스크린샷, 텔레비전, 화면, 검은색이(가) 표시된 사진

자동 생성된 설명

핸들러/리스너 : 실제 이벤트가 발생 됐을 때 실행하는 함수 ( 이벤트에 할당한 함수 )

속성 : 바인딩 할 때 사용한 이벤트

- 이벤트 연결 하는 방식

DOM Level0 : 인라인 이벤트 모델, 기본 이벤트 모델

DOMLevel 2 : 마이크로소프트 인터넷 익스플로어 이벤트 모델 , 표준 이벤트 모델

\* DOMLevel : 표준화 버전(?) 이라고 생각하면 됨.

11.3 고전 이벤트 모델

- 태그에 해당하는 객체를 생성해 이벤트를 바인딩 한다.

- 문서 객체의 이벤트 속성으로 이벤트를 연결하는 방법

- 고전 이벤트 모델은 이벤트 하나에 리스너 하나만을 연결 할 수 있다.

방법 )

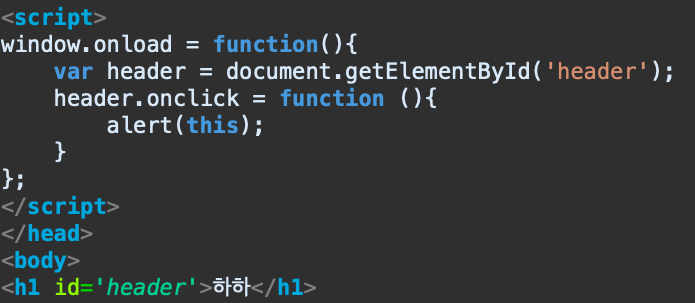
1. 변수 선언+변수와 DOM실제객체 바인딩 🡪 var eveObj = document.getElementById(‘anything’);

2. 해당 객체의 속성 중 원하는 이벤트 속성에 리스너 바인딩🡪 eveObj.onclick = function(){ //… 정의하세요}

\* 이벤트 제거 시 null 대입한다. 🡪 eveObj.onclick = null;

11.4 이벤트 발생 객체와 이벤트 객체

- 이벤트 리스너 안에서 this 키워드를 사용하면 이벤트가 발생한 객체를 찾을 수 있다.



11.5 이벤트 강제 호출

- 해당 이벤트 속성명을 메서드 호출처럼 호출하면 된다.

button.onclick = function(){

alert(“버튼 클릭”);

}

buttonA.onclick = function(){

button.onclick(); // 버튼A 클릭 시 위 선언한 버튼 클릭 메서드 실행.

}

11.6 인라인 이벤트 모델

- 태그 생성 시점에서 이미 이벤트가 바인딩 되어있음.

- script 에서 바인딩이 아닌 html 문서에서 이벤트를 바인딩하는것



- 여러줄의 경우는 정의부를 script로 옮긴 후 바인딩하는것이 좋다.

화면이(가) 표시된 사진

자동 생성된 설명

11.7 디폴트 이벤트 제거

- 디폴트 이벤트란? HTML 태그 일부 중 태그가 이미 기본적으로 이벤트 리스너를 갖고있을 때 해당 이벤트를 디폴트 이벤트라 한다.

고전 이벤트모델의 경우

ex) submit 버튼의 경우 버튼을 누르면 자동으로 입력 양식을 제출하고 페이지를 새로고침한다.

따라서 submit 의 form의 새로고침 속성을 false로 주면 해당 이벤트가 사라진다.

스크린샷, 모니터, 화면, 텔레비전이(가) 표시된 사진

자동 생성된 설명

인라인 이벤트의 경우 ( 비추 )

- 유효성 검사의 코드를 예로 들자면 제출 버튼을 누를 시 유효성 검사 후 제출을 해야함.

- 그러나 return false 이므로 검사 자체를 하지 않는다.

- 이벤트 속성의 리턴값으로 함수를 넣는다.(콜백함수?!)

화면, 텔레비전이(가) 표시된 사진

자동 생성된 설명

11.8 이벤트 전달

- 이벤트 버블링 :: 자식 🡪 부모 순으로 이벤트 실행

스크린샷, 화면, 전화, 노트북이(가) 표시된 사진

자동 생성된 설명 이 경우 alert(‘para…’);🡪 alert(‘header’); 실행

따라서 paragraph 이후의 이벤트를 막는 코드를 추가한다.

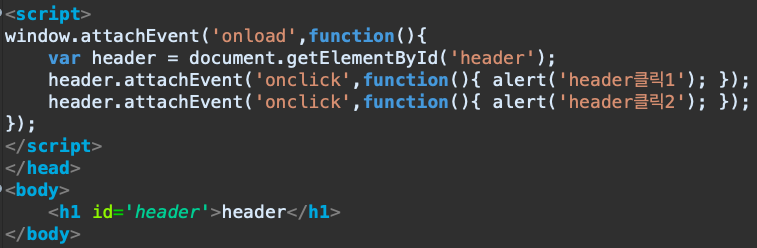
스크린샷, 전화, 화면, 검은색이(가) 표시된 사진

자동 생성된 설명

alert(‘para…’); 실행 후 alert(header’); 로 전달되는 이벤트를 막는다.

11.9 인터넷 익스플로러 이벤트 모델

- 위에서 설명한 DOMLevel 0 의 이벤트 모델은 한 번의 하나의 이벤트 리스너들만 가질 수 있다.



window.attachEvent('onload',**function**(){

**var** header = document.getElementById('header');

header.attachEvent('onclick',**function**(){ alert('header클릭1'); });

header.attachEvent('onclick',**function**(){ alert('header클릭2'); });

});

// 인터넷 익스플로러가 없어서 테스트 하지 못함. 누군가 해주셨으면…

- 제거 시는 header.detachEvent(이벤트 속성, 이벤트 리스너);

\*\* 참고로 인터넷익스플로러 이벤트 모델에서 이벤트 리스너의 this는 window객체이다.

\*\* 이벤트 발생 객체를 사용하기 위해서는 srcElement를 사용해야한다.

\*\* 모든 브라우저에서 사용 가능한 것이 아니므로 사용하지 않는 것을 권장한다.

11.10 표준 이벤트 모델

- W3C 에서 공식으로 지정한 이벤트 연결 방식

- 익스플로러 이벤트 모델 처럼 한 이벤트에 여러 리스너 등록 가능

- 이벤트 캡쳐링을 지원 하지만 사용하지 않는것을 추천함.

이벤트 추가 : addEventListener(eventName,handler, useCapture)

userCapture 생략 시 default 값 : false ~~~^

이벤트 제거 : removeEventListener(eventName, handler)