

# [Web API]이벤트 정리

## ▼ 이벤트

- 브라우저와 사용자가 상호작용 하면서 발생하는 사건들을 말함
- 마우스 이벤트 : click, mouseover, mouseout, mousedown, mouseup...
- 폼 요소 이벤트 : submit, focus...
- 키보드 이벤트 : keydown, keyup...
- 문서 이벤트 : DOMContentLoaded

## 이벤트 리스너(핸들러) 만드는 법

### 1. 인라인 방식

- html 태그의 속성(attribute)으로 등록하는 방식

```
<input value="클릭해 주세요." onclick="alert('클릭!')" type="button">클릭해 주세요.</input>

<!-- this로 요소에 접근 가능 : 버튼안에 클릭해 주세요를 삽입 -->
<button onclick="alert(this.innerHTML)">클릭해 주세요.</button>
```

### 2. 프로퍼티 방식

- onclick 프로퍼티는 하나 밖에 없기 때문에 프로퍼티 방식은 이벤트에 하나의 이벤트 리스너만 등록 가능
- 리스너를 하나 더 추가하면 기존 건 덮어씌워짐

```
<input id="elem" type="button" value="클릭해 주세요.">
<script>
  elem.onclick = function() {
    alert('감사합니다.');
```

### 3. addEventListener 메소드

*EventTarget*.addEventListener(*'eventType'*, *functionName* [, *useCapture*]);

대상요소      대상요소에 바인딩될 이벤트를 나타내는 문자열      이벤트 발생 시에 호출될 함수명 또는 함수 자체      capture 사용 여부  
true: capturing / false: Bubbling (Default)

- 동일한 요소에 여러개의 이벤트 리스너를 등록할 수 있음
- eventType : 이벤트 이름 문자열(click, keydown등)
- listener : 이벤트 리스너로 등록할 함수 이름
- option : {capture: false(캡처) / true(버블)} 기본 값은 false ...

```
element.addEventListener(event, handler, [options]);
```

## 주의사항

- JS : 함수를 이벤트 리스너에 할당할때 sayThanks라한다. 만약 함수를 쓸경우 sayThanks()가 아무것도 반환하지 않으면 onclick 프로퍼티에 undefined가 할당 됨
- HTML : onclick();

```
// 올바른 방법
button.onclick = sayThanks;

// 틀린 방법
button.onclick = sayThanks();
```

## 이벤트 객체

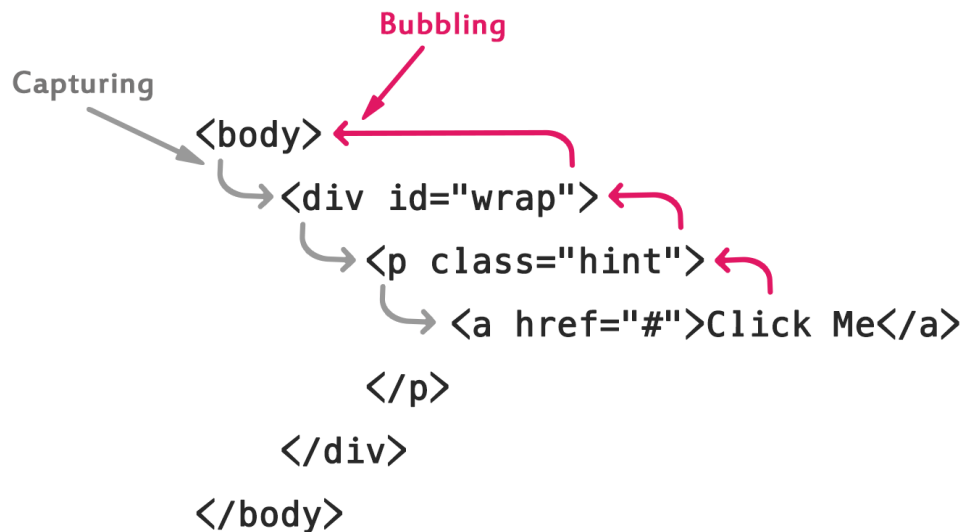
- 이벤트가 발생하면 브라우저는 이벤트 객체를 만들어 이벤트 리스너에 전달함
- 이벤트가 처리되고 나면 이벤트 객체는 소멸
- e, event
- target 프로퍼티 : 이벤트를 발생시킨 객체(태그) : event.target으로 접근할 수 있음

```
element.onclick = function(event) {};
```

## ▼ 이벤트 캡처링과 버블링

## 이벤트 흐름(event propagation)

- 이벤트가 발생하면 상위 요소로부터 DOM 트리를 타고 타겟 요소까지 들어왔다가 다시 상위 요소로 나감
- 캡처링 단계 - 이벤트가 하위 요소로 전파되는 단계
  - 타겟 단계 - 이벤트가 실제 타겟 요소에 전달되는 단계
  - 버블링 단계 - 이벤트가 상위 요소로 전파되는 단계



## 이벤트 버블링

- 자식 요소에 이벤트(click)가 발생하면 할당된 이벤트 리스너가 동작, 이후 부모 요소의 리스너가 동작하는 방식으로 작동함(기본값)
- `<p> → <div> → <form>`

## 이벤트 캡처링

- 자식 요소에서 발생한 이벤트가 부모 요소부터 시작해서 안쪽 자식 요소까지 도달
- 이벤트가 타겟 객체까지 내려가는 것(잘 사용할 일이 없다함)

## 이벤트 기본값 해제, 전파를 막는법

```
event.preventDefault(); // 이벤트의 기본 동작을 차단
event.stopPropagation(); // 이벤트 전파를 중지
event.stopImmediatePropagation(); // 이벤트 전파 중지 + 형제 이
```

## 이벤트 위임(Event Delegation)

- 다수의 자식 요소에게 각각 이벤트 리스너를 바인딩 하는 대신 하나의 부모 요소에 이벤트 리스너를 바인딩하는 방법

## 이벤트 흐름을 알아야 하는이유

- 이벤트 흐름을 이용하면 요소마다 이벤트 리스너를 만들어서 달 필요가 없음(상위 부모요소에 이벤트 리스너를 하나만 달면됨)
- todo 리스트나 게시글 삭제버튼을 만들때 유용하게 쓰일것으로 보임

출처 :

<https://ko.javascript.info/introduction-browser-events>

<https://ko.javascript.info/bubbling-and-capturing>

[https://inpa.tistory.com/entry/JS-📖-버블링-캡처링](https://inpa.tistory.com/entry/JS-%EB%B8%94%EB%B8%A0%EB%B8%A0-%EB%B8%A0%EB%B8%A0%EB%B8%A0)

[https://inpa.tistory.com/entry/JS-📖-이벤트-~~100~~-총-정리](https://inpa.tistory.com/entry/JS-📖-이벤트-<del>100</del>-총-정리)