

# 자바스크립트 (Javascript)

✓ 원리를 알면 IT가 맛있다

**JavaScript for Beginners**

chapter 09.

---

이벤트 처리  
( event handling )

- 자바스크립트 이벤트 모델
- DOM Level 0
- DOM Level 2
- 이벤트 전파 개념  
(캡처링과 버블링 개념)

### ○ 자바스크립트 이벤트 모델

- 자바스크립트는 이벤트 드리븐 ( event-driven) 모델에 기반한다.  
웹 페이지안에서 발생한 여러 가지 사건(이벤트)에 따라 대응하는 방법을(이벤트 핸들러) 이용한 처리이다.
- 이벤트에 대응해 처리를 담당하는 함수를 ‘이벤트 핸들러’라고 한다.

### ○ 이벤트 발생 형태

가. 애플리케이션 사용자가 발생시키는 이벤트  
예> 버튼클릭, 키보드 입력등.

나. 애플리케이션 스스로가 발생시키는 이벤트  
예> 페이지 로드 등

### ○ 주요 이벤트

분류	이벤트명	발생 타이밍	주된 대상 요소
읽기	abort	이미지의 로딩이 중단되었을 때	img
	load	페이지, 이미지의 로딩 완료 시	body, img
	unload	다른 페이지로 이동할 때	body
마우스	click	마우스 클릭 시	-
	dblclick	더블클릭 시	-
	mousedown	마우스 버튼을 눌렀을 때	-
	mousemove	마우스 포인터가 이동했을 때	-
	mouseout	요소에서 마우스 포인터가 떨어졌을 때	-
	mouseover	요소에 마우스 포인터가 겹쳤을 때	-
	mouseup	마우스 버튼을 떼어 놓았을 때	-
	contextmenu	context menu가 표시되기 전	body
키	keydown	키를 눌렀을 때	-
	keypress	키를 누른 상태	-
	keyup	키를 떼어 놓았을 때	-
폼	change	내용이 변경되었을 때	input(text), select
	reset	리셋 버튼이 눌렀을 때	form
	submit	서브밋 버튼이 눌렀을 때	form
포커스	blur	요소로부터 포커스가 벗어났을 때	-
	focus	요소가 포커스되었을 때	-
그 외	resize	요소의 사이즈를 변경했을 때	-
	scroll	스크롤했을 때	body

### ○ 주요 이벤트 핸들러

이벤트 핸들러	기능 설명
onAbort	사용자의 작업을 빠져나오는 이벤트
onBlur	입력 폼 양식을 다른 곳으로 이동하는 이벤트
onChange	입력 폼 양식을 변경해 주는 이벤트
onClick	입력 폼 양식을 마우스로 클릭해 주는 이벤트
onDbClick	마우스를 더블 클릭해 주는 이벤트
onDragDrop	마우스를 드래그하여 끌어주는 이벤트
onError	이미지나 윈도우를 읽는 도중 에러가 발생할 때 수행하는 메소드
onKeyDown	사용자가 키를 눌렀을 때 발생하는 이벤트

이벤트 핸들러	기능 설명
onKeyPress	사용자가 키를 눌렀다가 놓았을 때 발생하는 이벤트
onKeyUp	사용자가 키를 눌렀다가 떼었을 때 발생하는 이벤트
onLoad	이미지나 문서 프레임 등을 로드 시키는 이벤트
onMouseDown	사용자가 마우스 버튼을 눌렀을 때 발생하는 이벤트
onMouseMove	마우스를 이동시키는 이벤트
onMouseOut	링크나 클라이언트 측에서 마우스를 옮기는 이벤트
onMouseOver	마우스를 링크나 클라이언트 측으로 옮기는 이벤트
onMouseUp	사용자가 마우스를 놓는 이벤트
onMove	윈도우나 프레임을 옮기는 이벤트
onReset	폼을 리셋 시키는 이벤트
onResize	윈도우나 프레임의 크기를 재조정하는 이벤트
onSelect	필드나 영역을 선택하는 이벤트
onSubmit	폼을 보내는 이벤트
onUnload	문서나 프레임 세트를 종료하는 이벤트
onFocus	문서나 윈도우, 폼 요소 등에 입력 포커스를 설정하는 이벤트

이벤트 모델의 핵심은 ‘이벤트’ 와 ‘이벤트 핸들러’ 이다.

HTML의 어떤 요소에서 발생하는 이벤트에 대해서 어떤 이벤트 핸들러를 사용할지를 정해서 구현한다.

자바스크립트에서는 이벤트와 이벤트 핸들러를 연결하기 위한 방법으로 다음과 같이 2가지 방법이 제공된다.

가. DOM Level 0

인라인 이벤트 모델, 고전 이벤트 모델

나. DOM Level 2

표준 이벤트 모델방법이다.



- DOM Level 0 에서 제공하는 이벤트 핸들러의 이름은 주요 이벤트명 앞에 on접두사를 지정하여 사용된다.  
예> onload, onclick, onfocus, onblur , onchange, onkeyup, onmouseover 등

- 고전 이벤트 모델

```
<script>
window.onload = function() {
    document.getElementById('btn').onclick = function(){
        window.alert('안녕?');
    };
};
</script>
<input id="btn" type="button" value="다이얼로그 표시">
```

- 인라인 이벤트 모델

```
<script>
```

```
function btn_onclick(){
```

```
    window.alert('안녕?');
```

```
}
```

```
</script>
```

```
<input type="button" value="다이얼로그 표시" onclick="btn_onclick()">
```

### 인라인 이벤트 모델과 고전 이벤트 모델에 공통으로 적용되는 규칙

1. 각 요소에 이벤트 타입별로 하나의 이벤트 핸들러만 연결할 수 있다.
2. 이벤트 핸들러 내에서 이벤트 발생 객체를 사용하려면 `this` 를 사용한다.
3. 이벤트 핸들러 내에서 발생한 이벤트의 정보를 담고 있는 이벤트 객체를 사용하려면 핸들러 함수의 파라미터로 전달받는다. (IE 8 이하는 `window.event` 사용)
4. 요소에 등록되어 있는 기본 이벤트 핸들러를 해제하려면 이벤트 핸들러 함수에서 마지막에 `false` 를 되돌린다.
5. 고전 이벤트 모델은 요소에 연결된 이벤트 핸들러를 해제할 수 있지만 인라인 이벤트 모델은 해제할 수 없다.

- 2000년 11월 W3C에서 이벤트 처리 표준안으로 발표.
- IE에서는 9 버전부터 지원된다.
- 함수의 파라미터로 이벤트 정보가 전달된다.

- 표준 이벤트 모델

다음과 같은 이벤트 연결/해제 메서드들을 모든 DOM 객체들이 지원한다.

- `addEventListener(eventName, handler, useCapture)` → 연결
- `removeEventListener(eventName, handler)` → 해제

(IE 8 까지는 `attachEvent(eventProperty,eventHandler)`, `detachEvent(eventProperty,eventHandler)`를 지원)

### \* useCapture

- `false` : 기본값, 이벤트 버블링으로 이벤트 전파.
- `true` : 이벤트 캡처링으로 이벤트 전파.

표준 이벤트 모델에 적용되는 규칙

1. 각 요소에 이벤트 타입별로 여러 개의 이벤트 핸들러를 연결할 수 있다.
2. 이벤트 핸들러 내에서 이벤트 발생 객체를 사용하려면 `this` 를 사용한다.  
(IE 8 이하에서는 `window.srcElement` 를 사용한다.)
3. 이벤트 핸들러 내에서 발생한 이벤트의 정보를 담고 있는 이벤트 객체를 사용하려면 핸들러 함수의 파라미터로 전달받는다. (IE 8 이하는 `window.event` 사용)
4. 요소에 등록되어 있는 기본 이벤트 핸들러를 해제하려면 이벤트 객체의 `preventDefault()` 를 호출한다. (IE 8 이하에서는 `window.event.returnValue = false;` 을 사용)

- 이벤트가 처음 발생되면 DOM의 최상위 객체인 document 객체로 이벤트가 전달된다.
- 자바스크립트에서는 발생한 이벤트를 다음 과정을 거쳐 전파시킨다.

### 가. 이벤트 캡처( 캡처링)

document 부터 아래로 내려가서 이벤트 발생요소의 부모객체까지 전달되는 것을 의미한다.

### 나. 이벤트 타겟

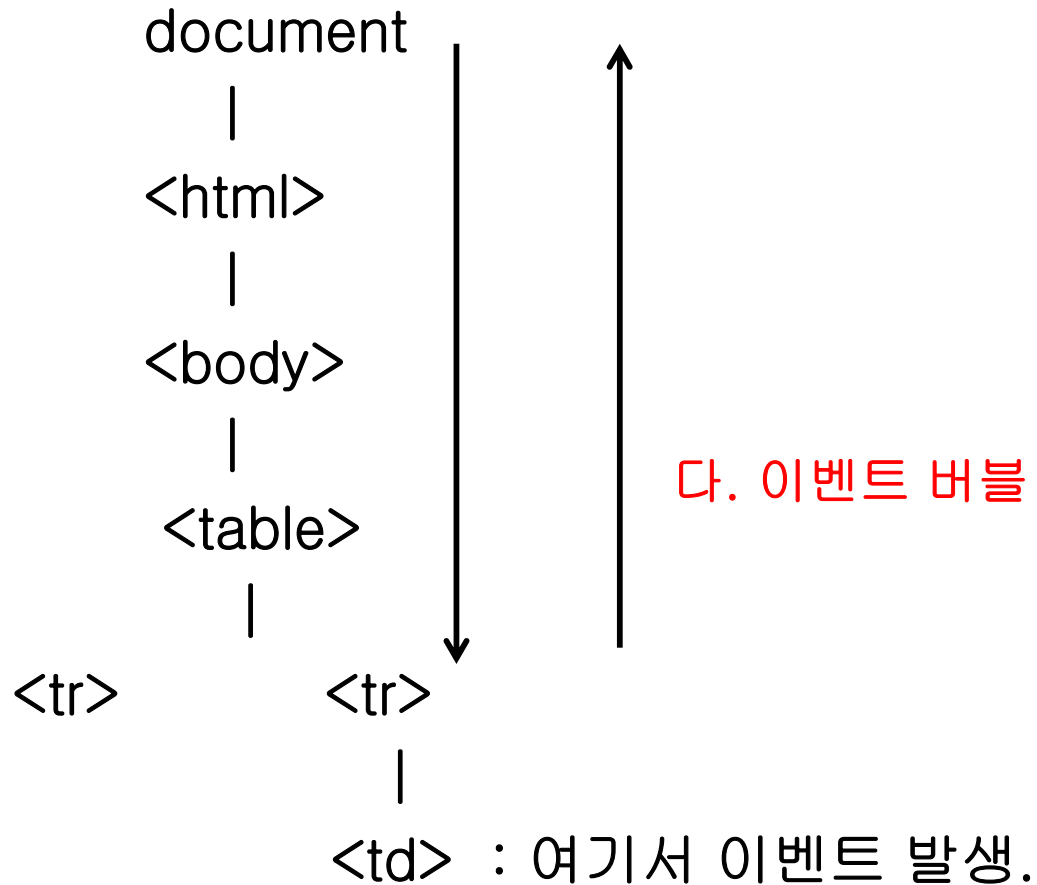
이벤트 발생요소까지 전달.

### 나. 이벤트 버블 (버블링): JavaScript의 기본 처리 방식

이벤트 발생요소의 부모로부터 위로 document까지 거슬러 올라가는 것을 의미한다.

예>

가. 이벤트 캡처



버블링의 이벤트 전파를 막아야 된다.  
e.stopPropagation() 함수 사용.

나. 이벤트 타겟

- 자바스크립트 이벤트 모델
- DOM Level 1
- DOM Level 2
- 이벤트 전파 개념  
(캡처링과 버블링 개념)





**Thank you**

---