

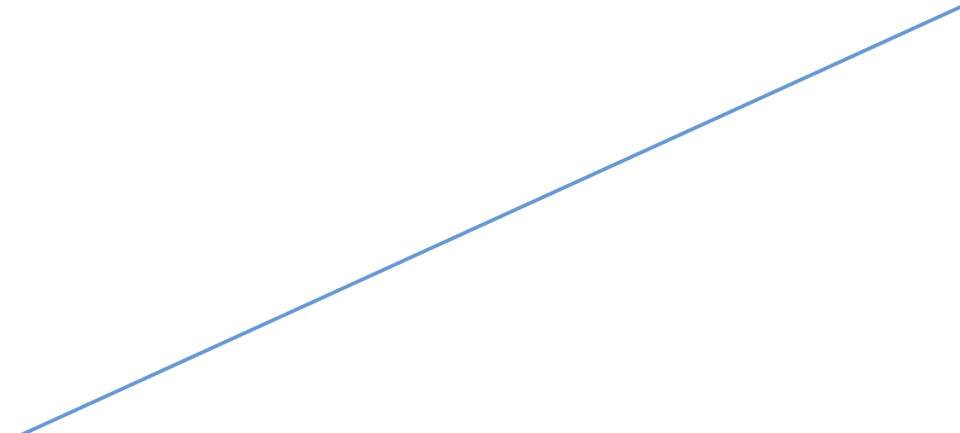
Events

Front-End 개발 실무

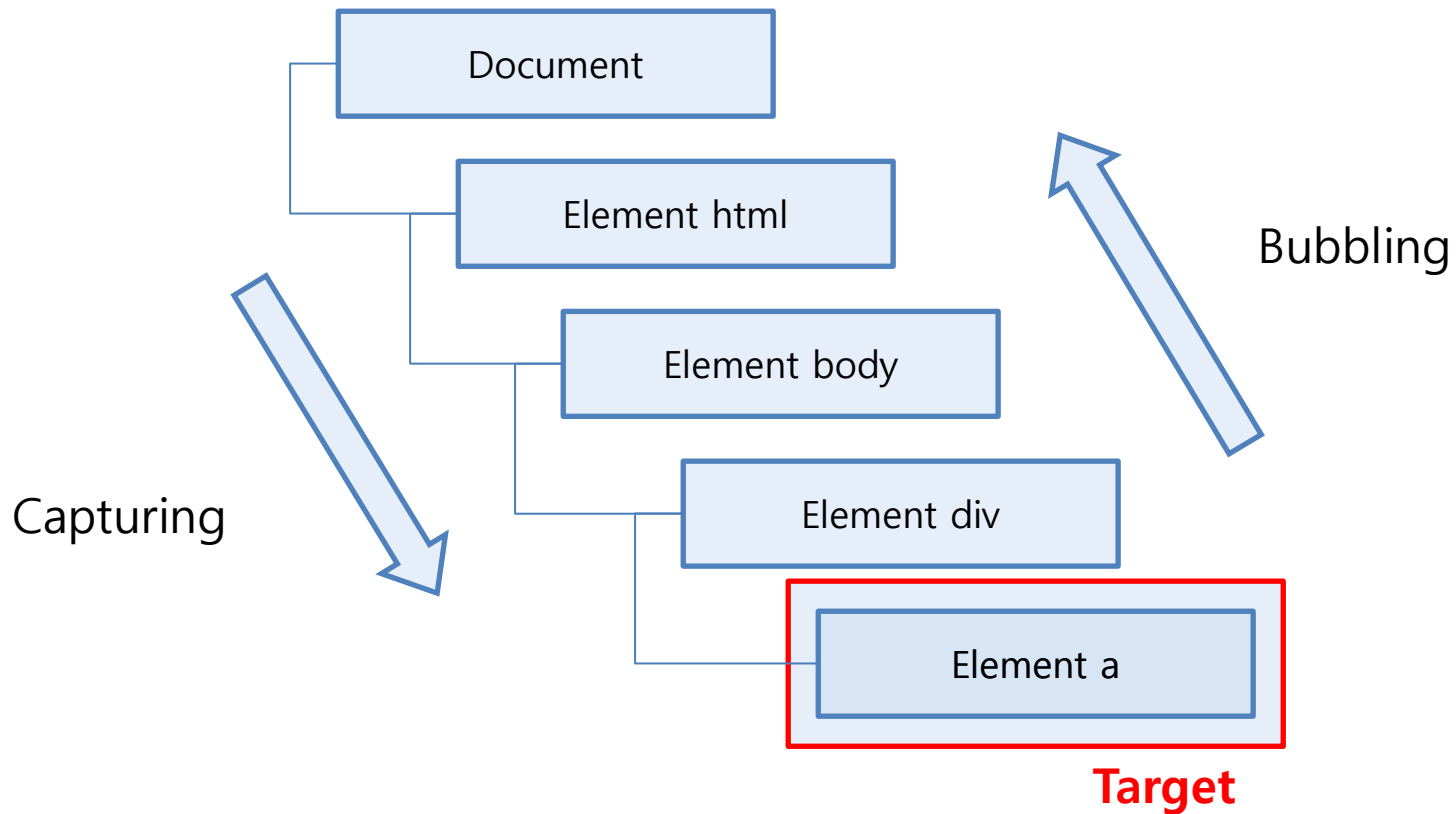


DinoWorks
DIGITAL NOMAD

이벤트 (Events)

- 이벤트 흐름
 - 이벤트 핸들러
 - 이벤트 객체
 - 이벤트 타입
- 

- 이벤트 버블링(Bubbling)
- 이벤트 캡처링(Capturing)



➤ HTML 이벤트 핸들러

```
<input type="button" value="Click Me" onclick="alert('clicked')" />
```

➤ DOM 레벨 0 이벤트 핸들러

```
var btn = document.getElementById("myBtn");  
btn.onclick = function(){  
    alert("clicked");  
}
```

```
btn.onclick = null; // 이벤트 제거
```

➤ DOM 레벨 2 이벤트 핸들러 (표준)

```
var btn = document.getElementById("myBtn");  
btn.addEventListener("click", function(){  
    alert(this.id);  
}, false);  // 이벤트 제거는 못함  
  
btn.addEventListener("click", handler, false);  
btn.removeEventListener("click", handler, false); // 이벤트 제거  
var handler = function(){  
    alert("Clicked");  
}
```

➤ IE 이벤트 핸들러 (IE8 이하)

```
var btn = document.getElementById("myBtn");  
btn.attachEvent("onclick", function(){  
    alert(this.id);  
}); // IE8 이하에서는 버블링(false)만 지원  
  
btn.attachEvent("onclick", handler, false);  
btn.detachEvent("onclick", handler, false); // 이벤트 제거  
var handler = function(){  
    alert("Clicked");  
}
```

크로스브라우징 : jQuery를 쓰는 이유

➤ DOM 이벤트 객체

- bubbles
- cancelable
- **currentTarget**
- defaultPrevented
- detail
- eventPhase
- **preventDefault()**
- stopImmediatePropagation()
- **stopPropagation()**
- **target**
- trusted
- **type**
- view

e

event

➤ 이벤트 객체 (jQuery)

```
$("#button1").on("click", function(event){  
    event.preventDefault();  
    alert(event.type);  
});
```

```
$('a[href="#"]').on("click", function(e){  
    e.preventDefault();  
});
```


➤ UI 이벤트

○ load

- 페이지를 완전히 불러왔을 때 **window**에서, 모든 프레임을 완전히 불러왔을 때 프레임셋에서, 이미지나 객체를 완전히 불러왔을 때 **** 요소나 **<object>** 요소에서 발생

○ unload

- 페이지를 완전히 종료했을 때 **window**에서, 모든 프레임을 완전히 종료했을 때 프레임셋에서, 객체를 완전히 종료했을 때 **<object>** 요소에서 발생

○ abort

- **<object>** 요소의 콘텐츠를 완전히 내려받기 전에 사용자가 취소했을 때 해당 요소에서 발생

○ error

- 자바스크립트 에러가 발생했을 때 **window**에서, 이미지를 불러올 수 없을 때 해당 **** 요소에서, **<object>** 요소 콘텐츠를 불러올 수 없을 때 해당 요소에서, 프레임을 불러올 수 없을 때 해당 프레임셋에서 발생

○ select

- 사용자가 텍스트 박스(**<input>**)이나 **<textarea>**)에서 글자를 선택할 때 발생

○ resize

- **window**나 프레임의 크기를 바꿀 때 발생

○ scroll

- 사용자가 **스크롤바 있는 요소**를 스크롤할 때 발생. **<body>** 요소에는 페이지 전체에 대한 스크롤바가 있음

➤ Focus 이벤트

- blur
 - 요소가 포커스를 잃을 때 발생. 이 이벤트는 **버블링 되지 않음** (모든 브라우저 지원)
- focus
 - 요소가 포커스를 받을 때 발생. 이 이벤트는 **버블링 되지 않음** (모든 브라우저 지원)
- focusin
 - 요소가 포커스를 받을 때 발생. HTML 이벤트 focus의 **버블링 버전**
- focusout
 - 요소가 포커스를 잃을 때 발생. HTML 이벤트 blur의 **버블링 버전**

➤ Mouse/Wheel 이벤트

- click
 - 사용자가 주요 마우스 버튼(일반적으로 왼쪽 버튼)을 클릭하거나 엔터키를 누를때 발생. **키보드와 마우스에 모두 반응하므로 접근성 구현에 중요** (mousedown → mouseup → click)
- dblclick
 - 사용자가 주요 마우스 버튼(일반적으로 왼쪽 버튼)을 더블클릭할 때 발생. (DOM 레벨 3 이벤트에서 표준화)
- mousedown
 - 사용자가 마우스 버튼을 누를때 발생
- mouseenter
 - 마우스 커서가 요소 밖에서 요소 경계 안으로 처음 이동할 때 발생. 이 이벤트는 **버블링되지 않으며** 커서가 **자손 요소 위에 올라갈 때 발생하지도 않음**. (DOM 레벨 3 이벤트에서 추가)
- mouseleave
 - 마우스 커서가 요소 위에 있다가 요소 경계 밖으로 이동할 때 발생. **버블링되지 않으며** 커서가 **자손 요소 위에 올라갈 때 발생하지도 않음**. (DOM 레벨 3 이벤트에서 추가)
- mousemove
 - 마우스 커서가 요소 위를 이동하는 동안 **계속** 발생.
- mouseout
 - 마우스 커서가 요소 위에 있다가 다른 요소(경계 밖의 요소 혹은 자식 요소) 위로 이동할 때 발생.
- mouseover
 - 마우스 커서가 요소 바깥에 있다가 요소 경계 안으로 이동할 때 발생
- mouseup
 - 사용자가 마우스 버튼을 누르고 있다가 놓을 때 발생.
- mousewheel (DOMMouseScroll)
 - 사용자가 마우스 휠을 세로 방향으로 움직일 때 발생.

➤ 키보드/텍스트 이벤트

- keydown
 - 사용자가 키를 처음 누를 때 발생하며 누르고 있는 동안 계속 발생
- keypress
 - 사용자가 키를 누른 결과로 문자가 입력되었을 때 처음 발생하며 누르고 있는 동안 계속 발생. ESC 키에서도 발생. (DOM 레벨 3에서는 **keypress 이벤트를 폐기**했으며, textInput 이벤트를 권장)
- keyup
 - 사용자가 키에서 손을 떼 때 발생
- textInput
 - 텍스트가 텍스트 박스에 삽입되기 직전에 발생. (문자 입력전에 이벤트를 가로챌 수 있음)
 - **DOM 레벨 3**에서 추가

➤ 조합(Composition) 이벤트

- compositionstart

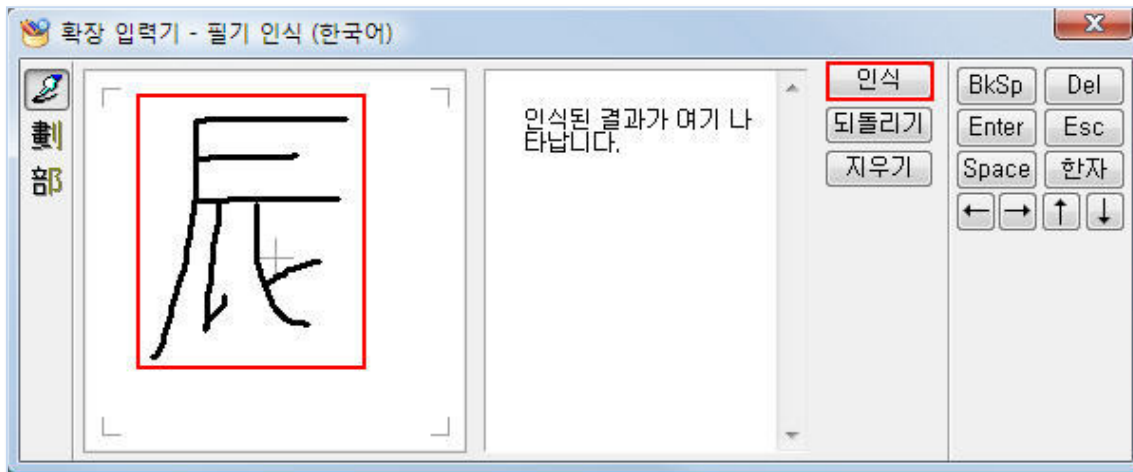
- IME의 텍스트 조합 시스템이 열리는 순간 발생하며 곧 문자가 입력될 것임을 나타냄.

- compositionupdate

- 입력 필드에 새 문자가 삽입될 때 발생.

- compositionend

- 텍스트 조합 시스템이 닫힐 때 발생하며 일반적인 키보드 입력으로 돌아갈 것을 나타냄.



➤ mutation(변경) 이벤트

- DOMSubtreeModified
 - DOM 구조가 어떻게든 바뀌었을 때 발생. 이벤트는 다른 이벤트가 모두 발생한 후에 발생. (DOM 레벨 2)
- DOMNodeInserted
 - 노드가 다른 노드의 자식으로 삽입될 때 발생.
- DOMNodeRemoved
 - 노드가 부모 노드로부터 제거될 때 발생.

➤ HTML5 이벤트

○ contextmenu

- **컨텍스트 메뉴**가 표시되려는 순간에 발생. 버블링되어 document 이벤트 핸들러 하나만 할당해서 페이지에서 발생하는 contextmenu 이벤트를 모두 처리 가능. (마우스 이벤트로 간주)

○ beforeunload

- window에서 발생. 브라우저가 페이지를 unload하기 직전에 발생하며 페이지를 언로드하겠다고 확인하지 않으면 계속 페이지에 머무르게 함.

○ DOMContentLoaded

- DOM 트리가 완전히 구성되는 즉시 발생.
- window의 load 이벤트는 페이지를 완전히 불러와야 발생하므로 외부 자원(이미지, JS, CSS 등)이 많이 포함된 페이지에서는 시간이 걸릴 수 있음.

○readystatechange


- 문서나 요소를 불러오는 상황에 대한 정보를 제공
- uninitialized, loading, loaded, interactive, complete

○ pageshow, pagehide

- 페이지가 표시될 때 발생(pageshow) : load 이벤트 보다 늦게 발생(**cache에서 가져온 경우에도 발생**)
- 페이지가 사라질 때 발생(pagehide) : unload 이벤트 직전에 발생

○ hashchange

- URL 해시(URL에서 #기호 다음 전부)가 바뀔 때 발생

 <https://mail.google.com/mail/u/0/#inbox>

➤ 장치 이벤트

- orientationchange
 - 사용자가 장치를 가로 모드나 세로 모드로 바꿀 때 발생
- deviceorientation
 - 가속도계가 부착된 장치에서 관련 있는 동작을 감지했을 때 window에서 발생
 - x, y, z 축의 변화 정도(각도)를 감지
- devicemotion
 - 장치가 실제로 움직이고 있을 때 발생
 - 장치가 떨어지고 있거나 누군가가 장치를 걷고 있는지 판단할 때 유용
 - x, y, z 축의 가속도를 감지(중력 감안여부 선택 가능)

➤ 터치와 제스처 이벤트

- touchstart
 - 손가락으로 화면을 터치할 때 발생. 이미 다른 손가락을 화면에 대고 있어도 다른 손가락을 대면 또 발생
- touchmove
 - 손가락을 화면에서 움직일 때 계속 발생.
 - 이 이벤트가 일어나는 동안 preventDefault()를 호출하면 스크롤을 막을 수 있음.
- touchend
 - 손가락을 화면에서 땔 때 발생.
- touchcancel
 - 시스템에서 터치를 더 이상 추적하지 않을 때 발생. (명확하게 문서화되지는 않음)
- gesturestart
 - 한 손가락을 화면에 얹은 채 다른 손가락으로 화면을 터치할 때 발생.
- gesturechange
 - 화면에서 두 손가락 중 하나의 위치가 바뀔 때 발생.
- gestureend
 - 두 손가락 중 하나를 화면에서 땔 때 발생.

- 가상 포커스의 동작(voice over / iOS)
 - 요소선택 더블탭
 - 이벤트 생성(➔ click)
 - 더블탭 종료 후 ➔ touchstart ➔ touchend ➔ mousedown ➔ focusout() ➔ blur() ➔ mouseup ➔ click
 - 가상포커스와 실제 웹 브라우저의 포커스는 다른 형태의 존재임
 - 이벤트 설계시 복잡하지 않도록 구현(click 위주)

다음강의

➤ jQuery 기초 및 실습



강사 연락처

➤ 디노웍스(주) 이시영 대표

➤ 문의

- E-mail : lsy@dinoworks.kr
- Mobile : 010-5179-6455

