

# UI 이벤트 1

## 마우스 이벤트

- 마우스 이벤트는 마우스 장치를 통해서만 생기는 것이 아니라 **핸드폰이나 태블릿 같은 다른 장치에서도 발생함**

### 마우스와 관련된 이벤트 종류

#### `mousedown, mouseup`

요소 위에서 마우스 왼쪽 버튼을 누를 때, 마우스 버튼 누르고 있다가 뗄 때 발생합니다.

#### `mouseover, mouseout`

마우스 커서가 요소 바깥에 있다가 요소 안으로 움직일 때, 커서가 요소 위에 있다가 요소 밖으로 움직일 때 발생합니다.

#### `mousemove`

마우스를 움직일 때 발생합니다.

#### `click`

마우스 왼쪽 버튼을 사용해 동일한 요소 위에서 `mousedown` 이벤트와 `mouseup` 이벤트를 연달아 발생시킬 때 실행됩니다.

#### `dblclick`

동일한 요소 위에서 마우스 왼쪽 버튼을 빠르게 클릭할 때 발생합니다. 요즘엔 잘 쓰이지 않습니다.

#### `contextmenu`

마우스 오른쪽 버튼을 눌렀을 때 발생합니다.

- 사용자는 단 하나의 동작을 했어도 실행되는 이벤트는 여러 개일 수 있음
  - 단순히 마우스 왼쪽 버튼을 클릭하더라도 `mousedown`, `mouseup`, 마지막으로 `click` 이벤트가 순서대로 발생 가능
- 그러나 보통은 `click` 이벤트만 거는 경우가 대부분이라서 이런 부분을 걱정 안해도 무방
- 클릭과 연관된 이벤트는 **어떤 버튼에서 이벤트가 발생했는지를 알려주는 `button` 프로퍼티를 지원**

### 주요 `event.button` 프로퍼티 값

버튼	<code>event.button</code>
왼쪽(주요 버튼)	0
가운데(보조 버튼 (=휠 버튼))	1
오른쪽 (두 번째 버튼)	2
X1(뒤로 가기 버튼)	3
X2(앞으로 가기 버튼)	4

- (마우스 휠이 없거나, 보조 버튼이 없는 마우스도 있으므로) **상당수의 마우스는 왼쪽, 오른쪽 버튼만 가지고 있기 때문에** 이 마우스들이 만들어내는 event.button 값은 0이나 2가 됨
- 마우스 이벤트는 이벤트가 발생할 때 **함께 누른 보조키가 무엇인지를 알려주는 프로퍼티**를 지원
  - 마우스 이벤트가 발생하는 도중에 해당 키가 함께 눌려졌다면 프로퍼티 값은 true가 됨

```
shiftkey : shift 키
altkey   : Alt 키 (MacOS에선 Opt 키)
ctrlkey  : Ctrl 키
metakey  : MacOS에서 Cmd 키
```

- 마우스 이벤트는 **두 종류의 좌표 정보**를 지원
  - clientX, clientY (클라이언트 좌표)
    - "웹 문서"를 기준으로 각각 왼쪽에서 얼마나 떨어져 있는지, 위에서 얼마나 떨어져 있는지를 나타내는데 페이지가 스크롤 되어도 변하지 않습니다.
  - pageX, pageY (페이지 좌표)
    - "창 왼쪽 위"를 기준으로 얼마나 떨어져 있는지를 나타내며 페이지를 스크롤하면 값도 변합니다.

```
<style>
body {
  width: 10000px;
  height: 10000px;
}
</style>
<body>
</body>
<script>
document.body.onmousemove = function(e) {
  console.log('clientX', e.clientX, 'clientY', e.clientY);
  console.log('pageX', e.pageX, 'pageY', e.pageY);
}
</script>
```

- 콘텐츠를 보호하려는 목적으로 방문자가 내용을 복사, 붙여넣기 하는 걸 막으려면 oncopy라는 이벤트를 활용 가능 (Ctrl + C 키를 누르는 순간 이벤트 발생)

## [TODO] Moving the mouse: mouseover/out, mouseenter/leave

- The mouseover event occurs when a mouse pointer comes over an element, and mouseout – when it leaves.
- they have property relatedTarget. This property complements target. When a mouse leaves one element for another, one of them becomes target, and the other one – relatedTarget

For mouseover:

`event.target` - is the element where the mouse came over.

`event.relatedTarget` - is the element from which the mouse came (`relatedTarget` → `target`).

For mouseout the reverse:

`event.target` - is the element that the mouse left.

`event.relatedTarget` - is the new under-the-pointer element, that mouse left for (`target` → `relatedTarget`).

- The mousemove event triggers when the mouse moves. But that doesn't mean that every pixel leads to an event.
  - The browser checks the mouse position from time to time. And if it notices changes then triggers the events.
- An important feature of mouseout - it triggers, when the pointer moves from an element to its descendant,
  - According to the browser logic, the mouse cursor may be only over a single element at any time - the most nested one and top by z-index.
- Events mouseenter/mouseleave are like mouseover/mouseout. They trigger when the mouse pointer enters/leaves the element.

## [TODO] 드래그 앤 드롭과 마우스 이벤트

---