

21-1학기 COMCODE

웹프로그래밍 심화

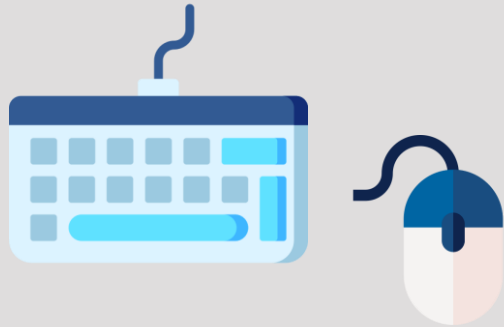
#5 Event



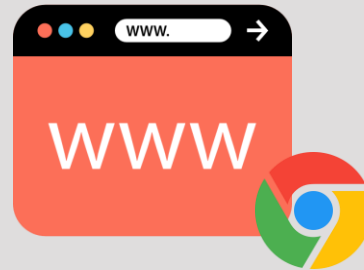
이벤트 핸들러



Event



클릭, 키보드 입력,
마우스 이동 등

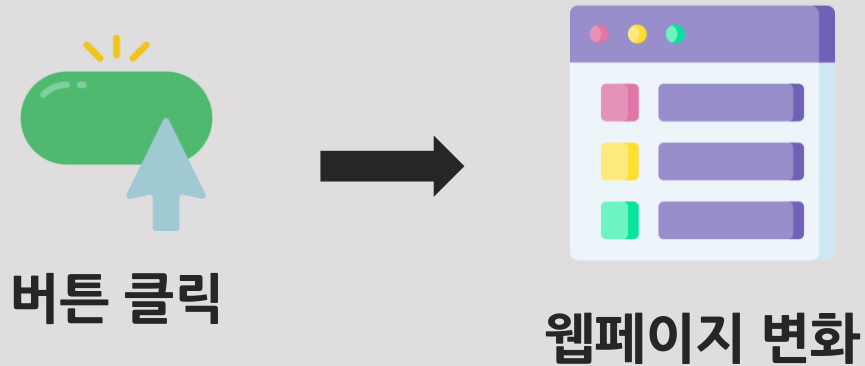


브라우저에서 감지

이벤트 발생

Event Handler

: 이벤트가 발생했을 때 호출될 함수



사용자와 애플리케이션의
상호작용

이벤트 드리븐 프로그래밍

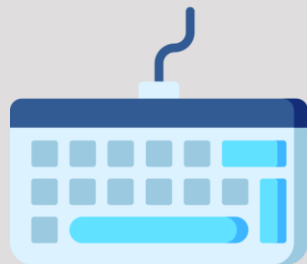
이벤트 타입



Event Type → 약 200개



마우스 이벤트



키보드 이벤트



포커스 이벤트



폼 이벤트



값 변경 이벤트


Event Type

그 외 이벤트 타입 (MDN 참고)

→ <https://developer.mozilla.org/ko/docs/Web/Events>

 MDN Web Docs
mozilla

▶ Technologies▶ References & Guides▶ Feedback

Search MDN 

Sign in

개발자를 위한 웹 기술 > 이벤트 참조  Change language | View in English 이 페이지는 영어로부터 커뮤니티에 의하여 번역되었습니다. MDN Web Docs에서 한국 커뮤니티에 가입하여 자세히 알아보세요.

Table of contents

- [가장 일반적인 카테고리](#)
- [덜 일반적이고 비표준인 이벤트](#)
- [표준 이벤트](#)
- [비표준 이벤트](#)
- [Mozilla-specific events](#)
- [See also](#)

이벤트 참조

DOM 이벤트는 발생한 흥미로운 것을 코드에 알리기 위해 전달됩니다. 각 이벤트는 `Event` 인터페이스를 기반으로한 객체에 의해 표현되며 발생한 것에 대한 추가적인 정보를 얻는데 사용되는 추가적인 커스텀 필드 또는 함수를 가질수도 있습니다. 이벤트는 렌더링 모델에서 기본적인 사용자 인터렉션부터 발생한 것에대한 자동 알림까지 모든 것을 나타낼 수 있습니다.

이 글은 전달될 수 있는 이벤트의 목록을 제공합니다. 일부는 공식 명세에 정의된 표준 이벤트이며, 다른 일부는 특정 브라우저에서 내부적으로 사용되는 이벤트입니다. 예로, Mozilla 특정 이벤트는 [부가 기능](#)이 브라우저와 상호작용하기 위해 나열되었습니다.



마우스 이벤트

click	마우스 버튼을 클릭했을 때
dblclick	마우스 버튼을 더블 클릭했을 때
mousedown	마우스 버튼을 눌렀을 때
mouseup	누르고 있던 마우스 버튼을 놓았을 때
mousemove	커서를 움직였을 때



마우스 이벤트

자신만 인식	mouseenter	커서를 HTML 요소 안으로 이동했을 때
	mouseleave	커서를 HTML 요소 밖으로 이동했을 때
자식도 인식	mouseover	커서를 HTML 요소 안으로 이동했을 때
	mouseout	커서를 HTML 요소 밖으로 이동했을 때



키보드 이벤트

keydown	모든 키를 눌렀을 때
keypress	문자 키를 눌렀을 때
keyup	누르고 있던 키를 놓았을 때



포커스 이벤트

자신만 인식	focus	HTML 요소가 포커스를 받았을 때
	blur	HTML 요소가 포커스를 잃었을 때
자식도 인식	focusin	HTML 요소가 포커스를 받았을 때
	focusout	HTML 요소가 포커스를 잃었을 때



폼 이벤트 / 값 변경 이벤트

submit	form 요소 내의 submit 버튼을 클릭했을 때
reset	form 요소 내의 reset 버튼을 클릭했을 때

input	input, select, textarea 요소의 값이 입력되었을 때
change	input, select, textarea 요소의 값이 변경되었을 때

이벤트 핸들러 등록



이벤트 핸들러 등록

: 이벤트가 발생했을 때 브라우저에게 이벤트 핸들러 호출 위임



HTML 요소의
어트리뷰트 활용



DOM 객체의
프로퍼티 활용



addEventListener
메서드 활용



HTML 요소의 어트리뷰트 활용

```
<button onEventType="function()">Hello</button>
```

```
function function() {  
    // code  
};
```



DOM 객체의 프로퍼티 활용

<button>Hello</button>

```
const button = document.querySelector('button');  
button.onEventType = function() {  
    // code  
};
```




addEventListener 메서드 활용

```
<button>Hello</button>
```

```
const button = document.querySelector('button');  
button.addEventListener('EventType', function() {  
    // code  
});
```

이벤트 핸들러 제거

```
<button>Hello</button>
```

```
const button = document.querySelector('button');  
button.addEventListener('EventType', function() {  
    // code  
});
```

```
button.removeEventListener('EventType', function);
```

(실습) 이벤트 타입 출력

1. MDN 사이트에서 여러 가지 이벤트 타입을 찾기
2. 이벤트가 발생하면 이벤트 타입을 브라우저 화면에 출력
3. 전에 있던 텍스트가 없어지지 않도록 출력하기

이벤트 객체



이벤트 객체

: 이벤트 핸들러의 첫번째 인수로 전달되는 객체

```
<div class="msg"></div>
```

```
const msg = document.querySelector('.msg');  
function showCoords(e) {  
    msg.textContent = `X: ${e.clientX}, Y: ${e.clientY}`;  
}  
document.onclick = showCoords;
```

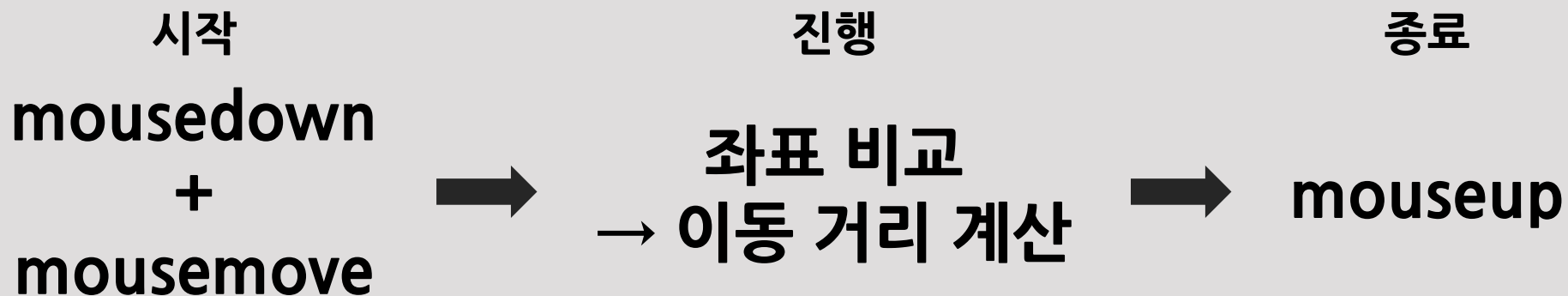
이벤트 객체의 프로퍼티

마우스 포인터의 좌표 정보	screenX/screenY, clientX/clientY, pageX/pageY, offsetX/offsetY
키보드 버튼 정보	altKey, ctrlKey, shiftKey, metaKey, key

(실습) 박스 드래그

드래그 시작 : 마우스 버튼을 누른 상태에서 마우스 이동

드래그 종료 : 마우스 버튼을 떼는 상태



(실습) 박스 드래그

hint

1. 시작 좌표와 이동 좌표를 객체로 할당하기
2. box 이동 → `style.transform`의 `translate3d()` 이용
3. `mousedown`가 발생하면 바로 `mousemove` 호출
4. `mouseup`가 발생하면 `mousemove` 제거

☐ (실습) 박스 드래그 (+추가)

드래그가 종료되면 박스와 테두리의 색깔이 랜덤으로 변경

hint

랜덤 색 설정

→ `parseInt(Math.random()*0xffffffff).toString(16)`

(과제) 웹페이지 기획하기

개인 프로젝트 기획서를 README.md 파일로 작성하기

예시)

- 개인 포트폴리오 홈페이지
- 쇼핑몰, 영화 예매 사이트
- 기존 웹사이트를 나만의 방식으로 UI/UX 개선하기