

삼육대학교 SW융합교육원 김진호

자바스크립트 기초교육

2-2. Event (이벤트 객체)

2-2-1. event object

2-2-2. this

❖ 2. Event

2-2. Event (이벤트 객체)

2-2-1. event object (이벤트 객체)

- 이벤트 발생 시 이벤트에 관련한 다양한 정보를 가진 이벤트 객체가 동적으로 생성된다.
- 생성된 이벤트 객체는 이벤트 핸들러의 첫 번째 인수로 전달된다.

```
const $msg = document.querySelector('.message');

// 클릭 이벤트에 의해 생성된 이벤트 객체는 이벤트 핸들러의 첫 번째 인수로 전달 된다.
function showCoords(e) {
  console.log(e);
  $msg.textContent = `clientX : ${e.clientX}, clientY : ${e.clientY}`;
}

document.onclick = showCoords;
```

```
<h1 class="message">아무곳이나 클릭해보세요.  
클릭한 좌표를 알려드릴게요.</h1>
```

❖ 2. Event

2-2. Event (이벤트 객체)

2-2-1. event object (이벤트 객체)

- 이벤트 핸들러 어트리뷰트 방식으로 이벤트 핸들러를 등록했다면 반드시 event라는 이름을 사용해야 한다.

```
const $area = document.querySelector('.area');

function showDivCoords(e) {
  console.log(e);
  $area.textContent = `clientX : ${e.clientX}, clientY : ${e.clientY}`;
}

/*
onclick="showDivCoords(event)" 어트리뷰트는 아래와 같은 함수를 암묵적으로 생성하여
onclick 이벤트 핸들러 프로퍼티에 할당하는데
이 때 onclick 이벤트 핸들러의 첫 번째 매개변수의 이름이 event로 암묵적으로 명명되기 때문에
event가 아닌 다른 이름으로는 이벤트 객체를 전달받지 못한다.

function onclick(event) {
  showDivCoords(event);
}
*/
```

```
<div class="area" onclick="showDivCoords(event)">
  이 영역 내부를 클릭해보세요. 클릭한 좌표를 알려드릴게요.
</div>
```

❖ 2. Event

2-2. Event (이벤트 객체)

2-2-2. this (이벤트 핸들러 내부의 this)

- 이벤트 핸들러 어트리뷰트 방식의 경우 이벤트 핸들러에 의해 일반 함수로 호출 되고 일반 함수 내부의 this는 전역 객체 window를 가리킨다.
- 이벤트 핸들러 호출 시 인수로 전달한 this는 이벤트를 바인딩한 DOM 요소를 가리킨다.

```
<button onclick="handleClick1()">클릭해보세요</button>  
<button onclick="handleClick2(this)">클릭해보세요</button>
```

```
function handleClick1() {  
    console.log(this);  
}  
  
function handleClick2(button) {  
    console.log(button);  
}
```

❖ 2. Event

2-2. Event (이벤트 객체)

2-2-2. this (이벤트 핸들러 내부의 this)

- 이벤트 핸들러 프로퍼티 방식과 addEventListener 메서드 방식 모두 이벤트 핸들러 내부의 this는 이벤트를 바인딩한 DOM 요소를 가리킨다.
- 즉, 이벤트 핸들러 내부의 this는 이벤트 객체의 currentTarget 프로퍼티와 같다.

```
const $btn1 = document.getElementById('btn1');
const $btn2 = document.getElementById('btn2');

// 이벤트 핸들러 프로퍼티 방식
$btn1.onclick = function (e) {
  console.log(this);
  console.log(e.currentTarget);
  console.log(this === e.currentTarget);
}

// addEventListener 메서드 방식
$btn2.addEventListener('click', function (e) {
  console.log(this);
  console.log(e.currentTarget);
  console.log(this === e.currentTarget);
});
```

```
<button id="btn1">클릭해보세요</button>
<button id="btn2">클릭해보세요</button>
```

❖ 2. Event

2-2. Event (이벤트 객체)

2-2-2. this (이벤트 핸들러 내부의 this)

- 화살표 함수로 정의한 이벤트 핸들러 내부의 this는 상위 스코프의 this를 가리킨다.
- 화살표 함수는 함수 자체의 this 바인딩을 갖지 않는다는 점에 유의한다.

```
const $btn3 = document.getElementById('btn3');  
const $btn4 = document.getElementById('btn4');
```

// 이벤트 핸들러 프로퍼티 방식

```
$btn3.onclick = e => {  
  console.log(this);  
  console.log(e.currentTarget);  
  console.log(this === e.currentTarget);  
}
```

// addEventListener 메서드 방식

```
$btn4.addEventListener('click', e => {  
  console.log(this);  
  console.log(e.currentTarget);  
  console.log(this === e.currentTarget);  
});
```

```
<button id="btn3">클릭해보세요</button>  
<button id="btn4">클릭해보세요</button>
```