

이벤트 처리

#01. 이벤트를 처리하는 방법

1) 이벤트 핸들러

HTML 태그의 속성 형태로 존재하는, 이벤트 발생시 실행할 자바스크립트 코드 영역

Javascript가 고도화 됨에 따라 HTML과 Javascript 코드를 분리하는 경향이 생기면서 이벤트 핸들러는 거의 사용되지 않는 추세.

```
<태그이름 on이벤트이름="...JS코드영역..."></태그이름>
```

2) 이벤트 리스너

어떤 이벤트가 발생하기를 기다리고 있다가 이벤트 발생시 연결된 함수를 호출해 주는 기능.

이벤트를 감지하는 것은 Javascript 안에 이미 내장되어 있지만 어떤 이벤트가 발생했을 수행할 동작은 웹 페이지마다 다를 수 밖에 없으므로 미리 구현해 놓을 수 없다. (페이지마다 개별 구현이 필요함)

이벤트를 감지하는 기능이 수행할 동작을 콜백함수로 요구한다.

바닐라JS의 이벤트 리스너

이벤트가 발생했을 때 호출되는 함수를 콜백으로 등록한다.

이 콜백함수를 **이벤트 핸들러** 라고 한다.

```
객체.addEventListener('이벤트이름', 콜백함수);
```

이벤트 핸들러 함수에는 **e**라는 파라미터가 전달된다. (파라미터 이름은 개발자 편의에 따라 변경 가능)

이 객체에는 이벤트 발생에 대한 정보가 포함되어 있는데 발생 이벤트의 유형에 따라 객체에 담기는 내용이 달라진다.

jQuery의 이벤트 리스너

```
객체.on("이벤트이름", 콜백함수);
```

이벤트 핸들러 처리는 바닐라JS와 동일하다.

\$(e.currentTarget)

복수 요소에 대한 이벤트에 전달된 이벤트 핸들러 안에서 이벤트가 발생한 주체를 의미하는 객체

```
<button class='btn'>button1</button>
<button class='btn'>button1</button>
<button class='btn'>button1</button>

<script>
  // `btn`이라는 클래스를 갖는 모든 요소에 대해서 일괄적으로 적용되는 이벤트
  $('`btn`').on("click", (e) => {
    // 버튼은 한번에 하나씩만 누를 수 있다.
    // 이 안에서 $(e.currentTarget)는 클릭된 주체를 의미한다.
  });
</script>
```

#02. jQuery 이벤트 종류

마우스 이벤트

이벤트	설명
click *	마우스를 클릭할 때
dblclick	마우스를 더블클릭할 때
mousedown	마우스 버튼을 누를 때
mouseup	마우스 버튼을 떼어낼 때
mousemove	마우스를 움직일 때
mouseenter *	마우스를 요소 안에 들어올 때
mouseleave *	마우스가 요소를 벗어날 때

키보드 이벤트

이벤트	설명
keydown	키보드가 눌릴 때
keypress	글자가 입력될 때
keyup *	키보드가 떼어질 때

윈도우 이벤트

이벤트	설명
ready	문서 객체가 준비를 완료함
load	윈도우(문서 객체)를 불러들일 때

이벤트	설명
unload	윈도우(문서 객체)를 닫을 때
resize	윈도우의 크기를 변화시킬 때
scroll *	윈도우를 스크롤할 때
error	에러가 있을 때

입력 양식 이벤트

이벤트	설명
change *	입력 양식의 내용을 변경 되었을 때(checkbox, radio, select)
focus	입력 양식 포커스가 맞춰졌을 때
blur	입력 양식에 포커스가 사라지면
submit	submit 버튼을 누르면
reset	reset 버튼을 누르면