

DOM의 기초6 이벤트

참고 URL

https://www.w3schools.com/jsref/dom_obj_event.asp

1) 인라인 방식

초창기 인터넷 넷스케이프 당시부터 사용하던 이벤트 방식입니다. 전통적인 방법으로 오랫동안 사용해 와서 많은 사람들이 아직도 사용하고 있습니다.

하지만 HTML을 문서로서 최적화하기 위하여, HTML은 오로지 document를 표시해주는 역할을 담당하게 하기 위해서는 이벤트 핸들러를 연결하는 것이 좋은 방법입니다.

활용 구문입니다.

HTML :

```
<div class="container">
  <button id="call" onclick="writeTxt()">event call</button>
  <textarea id="demo" class="note"></textarea>
</div>
```

JavaScript :

```
function writeTxt(){
  var area=document.querySelector(".note");
  // area.textContent="Hello JavaScript!!";
  // area.innerText="Hello JavaScript!!";
  area.innerHTML="Hello JavaScript!!";
}
```

2) 속성 방식

MVC HTML 패턴(Model View Controller)에 맞게 함수를 콜백 함수 형태(Listener, 리스너)로 만듭니다.

콜백 함수는 대입 형식의 함수입니다. 이 타입의 이벤트를 지양하는 이유는 중복된 이벤트의 실행이 되지 않기 때문입니다.

일반적인 함수 형식입니다.

```
function myFn(){
    // console.log("function call!!");
}
myFn();
```

대입 형식의 함수 형식입니다.

```
var myFn=function(){
    // console.log("callback function call!!");
};
myFn();
```

이벤트의 속성 방식입니다.

```
window.onload=function(){
    // console.log("load event!!");

    var button=document.getElementById("call");

    button.onclick=function(){
        // console.log("Hello JavaScript!!");
    };
    button.onclick=function(){
        // console.log("Hello jQuery!!");
    };
};
```

3) addEventListener() 방식

위에서 언급한 콜백 함수 방식의 오류로 인해서, 일반적인 JavaScript에서의 이벤트는 대부분 addEventListener(), attachement() 방식을 사용합니다.

특히 load 이벤트의 중복 호출이 문제가 될 때가 많이 있습니다. 하지만 Internet Explorer 8.0과 Opera 6.0 브라우저에서는 지원되지 않으므로 이 경우에는 attachEvent() 이벤트 방식으로 작성합니다.

addEventListener() 메서드는 지정된 요소에 이벤트 핸들러를 연결합니다.

```
element.addEventListener(event, function, useCapture)
```

Return Value : No return value

활용 구문입니다.

```
window.addEventListener("load", function(){
    // console.log("load event!!");

    var button=document.getElementById("call");

    button.addEventListener("click", function(){
        // console.log("Hello JavaScript!!");
    });
    button.addEventListener("click", function(){
        // console.log("Hello jQuery!!");
    });
});
```

4) attachEvent() 방식

addEventListener() 이벤트 방식은 일반 브라우저에서 적용하며, Internet Explorer 8.0과 Opera 6.0 브라우저에서는 attachEvent() 방식으로 작성합니다.

IE 8.0 브라우저를 위한 조건부 주석으로 작성할 필요가 있습니다.

```
element.getAttribute(attributename)
```

Return Value : A String

활용 구문입니다.

```
// console.log(window.addEventListener);

if(window.addEventListener){
    window.addEventListener("load", function(){
        var button=document.getElementById("call");
        var note=document.getElementById("demo");

        button.addEventListener("click", function(){
            note.innerHTML="Hello JavaScript!!";
        });
    });
}
else{
    window.attachEvent("onload", function(){
        var button=document.getElementById("call");
        var note=document.getElementById("demo");

        button.attachEvent("onclick", function(){
            note.innerHTML="Hello JavaScript!!";
        });
    });
}
```