

6장 이벤트

6.1 이벤트

6.1.1 이벤트란?

- 사이트에서 방문자가 취하는 모든 동작을 이벤트라 한다. 이벤트가 발생했을 때 실행되는 실행문을 이벤트 핸들러라 한다.

■ 이벤트 종류

종류		설명
마우스 이벤트	onmouseover	마우스가 지정한 요소에 올라갔을 때 발생한다.
	onmouseout	마우스가 지정한 요소에 벗어났을 때 발생한다.
	onmousemove	마우스가 지정한 요소 영역에서 움직일 때 발생한다.
	onclick	마우스가 지정한 요소를 클릭했을 때 발생한다.
	ondblclick	마우스가 지정한 요소를 연속 두 번 클릭했을 때 발생한다.
키보드 이벤트	onkeypress	지정한 요소에서 키보드가 눌렀을 때 발생한다.
	onkeydown	지정한 요소에서 키보드를 눌렀을 때 발생한다.
	onkeyup	지정한 요소에서 키보드를 눌렀다 떼었을 때 발생한다.
기타 이벤트	onfocus	지정한 요소에 포커스가 갔을 때 발생한다.
	onblur	지정한 요소에 포커스가 다른 요소로 이동되어 잃었을 때 발생한다.
	onchange	지정한 요소에 value 속성값이 바뀌고 포커스가 이동되었을 때 발생한다.
	onload	지정한 요소의 하위 요소를 모두 로딩했을 때 발생한다.
	onunload	문서를 닫거나 다른 문서로 이동했을 때 발생한다.
	onsubmit	폼 요소에 전송 버튼을 눌렀을 때 발생한다.
	onreset	폼 요소에 취소 버튼을 눌렀을 때 발생한다.
	onresize	지정한 요소의 크기가 변경되었을 때 발생한다.
	onerror	문서 객체가 로드되는 동안 문제가 발생되었을 때 발생한다.

```
// 직접 요소 이벤트 등록 방식
<button id="btn" onclick="alert('Event')">버튼</button>

// DOM을 이용한 이벤트 등록 방식
document.getElementById("btn").onclick=function(){
    alert('Event');
}
```

[실습파일: event_1_test.html, 완성파일: event_1.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트</title>
<script type="text/javascript">
    //
    var color = [ "white", "yellow", "aqua", "purple" ];
    var i = 0;
    function colorBg() {
        i++;
        if (i &gt;= color.length)
            i = 0;
        var bodyTag = document.getElementsByTagName("body")[0];
        bodyTag.style.backgroundColor = color[i];
    }

    window.onload = function() {
        //②, ④은 DOM을 이용해 이벤트를 등록하였습니다.

        //② [버튼2]를 클릭할 때 마다 colorBg함수를 호출합니다.</pre></div>
```

```

        var btn2 = document.getElementById("btn2");
        btn2.onclick = function() {
            colorBg();
        }

        //④ [버튼4]에 마우스를 올릴 때 마다 colorBg함수를 호출합니다.
        var btn4 = document.getElementById("btn4");
        btn4.onmouseover = function() {
            colorBg();
        }
    }
    //]]>
</script>
</head>
<body>
    <p>
        <!--① [버튼1]을 클릭할 때 마다 colorBg함수를 호출합니다. -->
        <button onclick="colorBg();">버튼1</button>
    </p>
    <p>
        <button id="btn2">버튼2</button>
    </p>
    <p>
        <!--③ [버튼3]을 클릭할 때 마다 colorBg함수를 호출합니다. -->
        <button onmouseover="colorBg();">버튼3</button>
    </p>
    <p>
        <button id="btn4">버튼4</button>
    </p>
</body>
</html>

```

6.1.2 키보드 접근성

- 요소에 마우스 이벤트를 등록할 경우에는 마우스가 없어도 접근(작동)할 수 있도록 해야 하는데, 이것을 키보드 접근성이라 한다.

- 마우스를 대신한 키보드 대응 이벤트

마우스 이벤트	키보드 대응 이벤트
onmouseover	onfocus
onmouseout	onblur

[실습파일: event_2_test.html, 완성파일: event_2.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트</title>
<script type="text/javascript">
    //
    var color = [ "white", "yellow", "aqua", "purple" ];
    var i = 0;
    function colorBg() {
        i++;
        if (i &gt;= color.length)
            i = 0;
        var bodyTag = document.getElementsByTagName("body")[0];
        bodyTag.style.backgroundColor = color[i];
    }

    window.onload = function() {
</pre>
</div>
```

```

        var btn2 = document.getElementById("btn2");
        btn2.onclick = function() {
            colorBg();
        }

        /*④ 마우스가 없을 경우 키보드 [tab]키를 사용해 이용할
        수 있도록 키보드 대응하는 이벤트 onfocus를 추가하였습니다.*/
        var btn4 = document.getElementById("btn4");
        btn4.onmouseover = btn4.onfocus = function() {
            colorBg();
        }
    }
    //]]>
</script>
</head>
<body>
    <p>
        <button onclick="colorBg();">버튼1</button>
    </p>
    <p>
        <button id="btn2">버튼2</button>
    </p>
    <p>
        <!--
        ③ 마우스가 없을 경우 키보드 [tab]키를 사용해 이용할
        수 있도록 키보드 대응하는 이벤트 onfocus를 추가하였습니다.
        -->
        <button onmouseover="colorBg();" onfocus="colorBg();">버튼3</button>
    </p>
    <p>
        <button id="btn4">버튼4</button>
    </p>
</body>
</html>

```

6.1.3 이벤트 중복 등록 해결법

(1) 한 요소에 같은 이벤트를 중복으로 등록하는 잘못된 사례

```

<button id="btn">버튼</button>

<button id="btn" onclick="alert('실행문1');" onclick="alert('실행문2');">버튼</button>
// 두 번째 경고 창인 alert('실행문2')만 실행된다.

document.getElementById("btn").onclick=function(){ // 실행 안 됨.
    alert('실행문1');
}
document.getElementById("btn").onclick=function(){ // 실행됨.
    alert('실행문2');
}

```

(2) 한 요소에 같은 이벤트를 중복으로 등록하는 방법

```

<button id="btn">버튼</button>

// IE 8 이하 외의 브라우저(파이어폭스, 크롬, 사파리 등) 이벤트 등록 방법
document.getElementById("btn").addEventListener("click",function(){
    alert('실행문1');
},false);

```

```
document.getElementById("btn").addEventListener("click",function(){
    alert('실행문2');
},false);

// IE 8 이하 브라우저 이벤트 등록 방법
document.getElementById("btn").attachEvent("onclick",function(){
    alert('실행문1');
});
document.getElementById("btn").attachEvent("onclick",function(){
    alert('실행문2');
});
```

[실습파일: double_event_1_test.html, 완성파일: double_event_1.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트</title>
<script type="text/javascript">
    <![CDATA[
        window.onload = function() {
            function fnc1() {
                alert('실행문1');
            }
            function fnc2() {
                alert('실행문2');
            }

            //id의 값이 "btn"인 <button>인 요소를 선택합니다.
            var myBtn = document.getElementById("btn");

            //파이어폭스, 크롬 등 브라우저에서 true를 반환하여 조건식을 만족함.
            if (window.addEventListener) {

                //<button id="btn">인 요소를 클릭할 때 마다 함수 fnc1을 호출합니다.
                myBtn.addEventListener("click", fnc1, false);

                //<button id="btn">인 요소를 클릭할 때 마다 함수 fnc2를 호출합니다.
                myBtn.addEventListener("click", fnc2, false);

            } else { //IE(8 이하)에서만 실행

                myBtn.attachEvent("onclick", fnc1);
                myBtn.attachEvent("onclick", fnc2);

            } //End : else
        } //End : onload
    </script>
</head>
<body>
    <button id="btn">버튼</button>
</body>
</html>
```

6.1.4 this 키워드

- 방문자가 사이트에 방문하여 이벤트를 등록한 요소에서 이벤트 발생시킬 때마다 익명 함수 내에 실행문이 실행될 것이다. 이때 이 실행문에 포함되어 있는 this라는 키워드는 방문자가 이벤트를 발행한 요소를 가리킨다. 가령, 클릭(click) 이벤트를 등록한 <a>의 요소가 8개 있는데 이때 방문자가 세 번째의 <a>를 클릭했다면, 이때 실행되는 일련의 함수 내의 this는

세 번째 <a>를 가리키는 것이다.

[실습파일: event_3_test.html, 완성파일: event_3.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트</title>
<script type="text/javascript">
  <![CDATA[
    function chColor(obj, state) {
      var myColor;
      if (state == "over") {
        myColor = "red";
      } else {
        myColor = "black";
      }
      obj.style.color = myColor;
    } //END:function chColor( )
  </![CDATA[
</script>
</head>
<body>
  <button onmouseover="chColor(this, 'over')"
    onmouseout="chColor(this, 'out')">마우스 아웃</button>
</body>
</html>
```

- 다음은 전송(submit) 버튼을 클릭했을 때 this를 이용해 이벤트가 발생한 <form> 요소를 함수 매개 변수에 전달한다.

[실습파일: event_4_test.html, 완성파일: event_4.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트</title>
<style type="text/css">
body {
  font: 12px gulim, "굴림", sans-serif;
}

fieldset {
  border: none 0;
}

legend {
  position: absolute;
  left: -9999px;
}
</style>
<script type="text/javascript">
  <![CDATA[
    function submit_chk(myForm) {
      var reg1 = /^[가-힣]{2,5}$/;
      var reg2 = /^(010|016|011)\d{3,4}\d{4}$/;

      /*이름 입력하는 <input name="user_name">을
      변수 name에 참조시켰습니다.*/
      var name = myForm.user_name;

      /*연락처 입력하는 <input name="user_tel">을
```

```

        변수 tel에 참조시켰습니다.*/
        var tel = myForm.user_tel;

        /*방문자가 입력한 이름이 정규표현 규칙에 어긋나면
        false를 반환합니다.*/
        var result1 = reg1.test(name.value);

        /*방문자가 입력한 전화번호가 정규표현 규칙에 어긋나면
        false를 반환합니다.*/
        var result2 = reg2.test(tel.value);

        /*이름 또는 연락처를 정규표현 규칙에 어긋나면
        경고 창을 띄우고 해당 입력 창에 포커스를 이동시킵니다.*/
        if (!result1) {
            alert("이름 입력이 잘못되었습니다.");
            name.value = "";
            name.focus();
            return false;
        } else if (!result2) {
            alert("연락처 입력이 잘못되었습니다.");
            tel.value = "";
            tel.focus();
            return false;
        }
    }

    /*취소 버튼을 눌렀을 때 확인/취소 창을 띄우고, 다시 한번
    회원가입 취소 여부를 묻습니다.*/
    function reset_chk(myForm) {
        var result = confirm("정말로 회원가입을 취소하시겠습니까?");
        if (result) {
            return false;
        }
    }
    //]]>
</script>
</head>
<!-- <body>에 하위 요소를 모두 로딩한 후 이름 입력하는
    곳으로 포커스 이동시킵니다. -->
<body onload="document.f1.user_name.focus();">
    <form action="#" method="post" name="f1"
        onsubmit="return submit_chk(this);" onreset="return reset_chk(this);">
        <fieldset>
            <legend>회원가입</legend>
            <p>
                <label for="user_name">이름</label> <input type="text"
                    name="user_name" id="user_name" />
            </p>
            <p>
                <label for="user_tel">연락처</label> <input type="text"
                    name="user_tel"
                    id="user_tel" />
            </p>
            <p>
                <input type="submit" value="회원가입" /> <input type="reset"
                    value="가입취소" />
            </p>
        </fieldset>
    </form>
</body>
</html>

```

6.2 이벤트 객체

- 이벤트 객체는 이벤트가 발생했을 때 생성되는 객체를 가리킨다. 가령, 마우스를 움직이면 마우스 이벤트 객체가 생성되는데 이 객체를 이용하면 마우스의 좌표 값을 구해 올 수 있다.

[실습파일: event_object_1_test.html, 완성파일: event_object_1.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트</title>
<script type="text/javascript">
    <![CDATA[
        document.onkeydown = function(e) {
            alert(e);
        }
    </![CDATA[
</script>
</head>

<body>

</body>
</html>
```

[실습파일: event_object_4_test.html, 완성파일: event_object_4.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>이벤트 객체</title>
<style type="text/css">
body {
    height: 2000px;
    width: 2000px;
    background: url("images/event_obj_bg.gif") 0 0 repeat-x;
}

#wrap {
    position: fixed;
    width: 80%;
}

#key {
    ime-mode: disabled;
}
</style>
<script type="text/javascript">
    <![CDATA[
        window.onload = function() {
            var form1 = document.f1;
            var form2 = document.f2;
            var eventObj;
            document.onmousemove = function(e) {
                eventObj = e ? e : window.event;
                var cx = eventObj.clientX;
                var cy = eventObj.clientY;
                form1.client_x.value = cx;
                form1.client_y.value = cy;

                var px = eventObj.pageX;
                var py = eventObj.pageY;
                form1.pg_x.value = px;
                form1.pg_y.value = py;
            }
        }
    </![CDATA[
</script>
```

```

    }

    form2.key.onkeydown = function(e) {
        eventObj = e ? e : window.event;
        form2.code.value = eventObj.keyCode;
        if (eventObj.ctrlKey == true && eventObj.keyCode == 71) {
            alert("단축키 실행");
        }
    }
}
//]]>
</script>
</head>
<body>
    <div id="wrap">
        <h1>마우스 이벤트 객체 속성</h1>
        <form action="#" method="get" name="f1">
            <fieldset>
                <legend>커서 위치 추적</legend>
                <p>
                    <label for="client_x">clientX :</label> <input type="text"
                        name="client_x" id="client_x" />
                </p>
                <p>
                    <label for="pg_x">pageX :</label> <input type="text"
                        id="pg_x" />
                </p>
                <p>
                    <label for="client_y">clientY :</label> <input type="text"
                        name="client_y" id="client_y" />
                </p>
                <p>
                    <label for="pg_y">pageY :</label> <input type="text"
                        id="pg_y" />
                </p>
            </fieldset>
        </form>
        <h1>키보드 이벤트 객체 속성</h1>
        <form action="#" method="get" name="f2">
            <fieldset>
                <legend>키 코드 값</legend>
                <p>
                    <label for="key">입력 키보드 :</label> <input type="text"
                        id="key" /> <label for="code">유니코드
                        값 :</label> <input type="text"
                        name="code" id="code" />
                </p>
            </fieldset>
        </form>
    </div>
</body>
</html>

```

[과제] 도전 Mission!

1. 다음은 이벤트가 발생했을 때 이미지를 변경하는 실행문이다. 코드를 완성하라.

- 클릭 이벤트가 발생한 요소(this)의 src 속성을 바꾼다.
- 마우스 오버 이벤트와 키보드에 대응하는 이벤트를 등록한다.
- addEventListener()를 이용해 mouseover 이벤트를 등록한다.

[실습파일: event_test_1_test.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 이벤트 </title>
<script type="text/javascript">
//
window.onload=function(){
    var btn1=document.getElementById("btn1");
    var btn2=document.getElementById("btn2");
    var btn3=document.getElementById("btn3");

    btn1.onclick=function(){//id 값이"btn1"인 요소를 클릭했을 때
        =this.children[0].src.replace("out.png","on.png");
        return false;
    }

    //id 값이"btn2"인 요소에 마우스를 올리거나 또는 포커스가 생성되었을 때
    =function(){
        this.children[0].src=this.children[0].src.replace("out.png","on.png");
    }

    function chBtn(){
        this.children[0].src=this.children[0].src.replace("out.png","on.png");
    }
    //id 값이"btn3"인 요소에 마우스가 벗어나거나 또는 포커스를 잃었을 때
    if(window.addEventListener){

        btn3.addEventListener("blur",chBtn,false);
    }else{
        btn3.attachEvent("onmouseout",chBtn);
        btn3.attachEvent("onblur",chBtn);
    }
}
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;마우스 클릭했을 때 이미지 변경&lt;/h1&gt;
&lt;p&gt;( images/click_btn_out.png → images/click_btn_on.png )&lt;/p&gt;
&lt;p&gt;&lt;a href="#" id="btn1"&gt;
    &lt;img src="images/click_btn_out.png" alt="버튼1" /&gt;
&lt;/a&gt;&lt;/p&gt;
&lt;h1&gt;마우스 오버했을 때 이미지 변경&lt;/h1&gt;
&lt;p&gt;( images/click_btn_out.png → images/click_btn_on.png )&lt;/p&gt;
&lt;p&gt;&lt;a href="#" id="btn2"&gt;
    &lt;img src="images/over_btn_out.png" alt="버튼2" /&gt;
&lt;/a&gt;&lt;/p&gt;
&lt;h1&gt;마우스 아웃했을 때 이미지 변경&lt;/h1&gt;
&lt;p&gt;( images/click_btn_out.png → images/click_btn_on.png )&lt;/p&gt;
&lt;p&gt;&lt;a href="#" id="btn3"&gt;
    &lt;img src="images/out_btn_out.png" alt="버튼3" /&gt;
&lt;/a&gt;&lt;/p&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="117 848 230 864" data-label="Section-Header"><h2>■ 결과 화면</h2></div>
```

