



Chapter 07

jQuery Events



jQuery 이벤트의 개요

jQuery 이벤트의 개요

■ 이벤트의 개요

• 이벤트 처리의 필요성

- 웹 프로그래밍에서의 처리는 대부분 이벤트를 기반으로 함
 - 함수의 내부에 존재하는 코드는 호출될 때만 수행이 가능하며,
 - 함수의 호출은 주로 이벤트를 사용함
- 웹 프로그램의 수행을 위한 적절한 이벤트의 사용이 중요함

• jQuery에서의 이벤트

- 문서 로딩 이벤트 (Document Loading Events)
- 마우스 관련 이벤트 (Mouse Events)
- 포커스 관련 이벤트 (Focus Events)
- 키보드 관련 이벤트 (Keyboard Events)

jQuery 이벤트의 개요

■ 이벤트의 등록

• 단독 이벤트 등록

- 대상에 한 가지 동작에 대한 이벤트만 등록할 수 있음
- 이벤트 등록 메서드를 사용한 직접 지정

```
$("#요소").이벤트등록메서드( function() {  
    이벤트핸들러 내용;  
});
```

```
<button id='btn1'> 버튼 </button>
```

```
$('#btn1').click( function() {  
    alert('welcome');  
});
```

jQuery 이벤트의 개요

- 그룹 이벤트 등록

- 대상에 한 가지 동작 이상의 이벤트를 등록할 수 있음
- 이벤트 연결 메서드를 사용해 지정
 - on(), delegate() 등

```
$("#요소").이벤트연결메서드( 이벤트, function() {  
    이벤트핸들러 내용;  
});
```

```
<button id='btn1'> 버튼 </button>
```

```
$('#btn1').on( 'click', function() {  
    alert('welcome');  
});
```

jQuery 이벤트의 개요

- 그룹 이벤트 등록 형식

- 형식 1 : 동일한 이벤트 핸들러를 가지는 서로 다른 이벤트들을 동시에 연결

```
$("#요소").이벤트연결메서드( "이벤트1, 이벤트2, ..., 이벤트n", function() {  
    이벤트핸들러 내용;  
});
```

```
$('#btn1').on( "mouseover focus", function() {  
    alert('welcome');  
});
```

```
$("#요소").이벤트연결메서드( { 이벤트1, 이벤트2, ..., 이벤트n : function() {  
    이벤트핸들러 내용;  
}  
});
```

```
$('#btn1').on( { "mouseover focus" : function() {  
    alert('welcome');  
}  
});
```

jQuery 이벤트의 개요

- 형식 2 : 서로 다른 이벤트 핸들러를 가진 다수의 이벤트를 동시에 연결

```
$("#요소").이벤트연결메서드(  
    "이벤트1" : function() { 이벤트핸들러 내용 },  
    "이벤트2" : function() { 이벤트핸들러 내용 },  
    .....  
    "이벤트n" : function() { 이벤트핸들러 내용 }  
);
```

```
<button id='btn1'> 버튼 </button>
```

```
$('#btn1').on(  
    { 'onmouseover', : function() { } },  
    { 'focus', : function() { } },  
);
```

jQuery 이벤트의 개요

■ 이벤트의 제거

- 특정 이벤트에 연결을 제거하여 이벤트의 수행을 중지시키는 것을 의미
 - 해당 이벤트에 연결된 이벤트들 중 일부 이벤트를 제거하거나 전체를 제거할 수 있음
- 이벤트 해제 방법
 - 이벤트 해제 메서드 사용

```
$( "요소" ).이벤트연결해제메서드( 이벤트 );
```

```
$( '#btn1' ).on( 'click' );
```


이벤트의 등록과 해제

이벤트의 등록과 해제

■ 이벤트 등록과 해제를 위한 메서드

메서드	설명	비고
delegate()	이벤트를 등록하기 위한 메서드	deprecate 되었음 거의 사용되지 않음
undelegate()	delegate() 메서드를 이용해 등록한 이벤트를 해제하기 위한 메서드	
live()	이벤트를 등록하기 위한 메서드	deprecate 되었음 거의 사용되지 않음
die()	live() 메서드를 이용해 등록한 이벤트를 해제하기 위한 메서드	
bind()	이벤트를 등록하기 위한 메서드	deprecate 되었음 거의 사용되지 않음
unbind()	bind() 메서드를 이용해 등록한 이벤트를 해제하기 위한 메서드	
on()	이벤트를 등록하기 위한 메서드	주로 사용됨
off()	on() 메서드를 이용해 등록한 이벤트를 해제하기 위한 메서드	
one()	이벤트가 한 번 수행되면 자동으로 소멸하는 메서드	

이벤트의 등록과 해제

■ on() 메서드

- 선택자에 의해 선택된 객체에 하나 이상의 이벤트를 등록함

```
$("요소"). on(  
    "이벤트1" : function() { 이벤트핸들러 내용 },  
    "이벤트2" : function() { 이벤트핸들러 내용 },  
    .....  
    "이벤트n" : function() { 이벤트핸들러 내용 }  
));
```

- 선택된 요소에 대해 하나 이상의 이벤트를 등록하면 각 이벤트마다 각각의 이벤트핸들러가 수행됨
 - 이벤트와 이벤트 핸들러는 : (콜론) 기호로 구분
 - 이벤트와 이벤트 핸들러의 쌍은 , (콤마) 기호로 구분

이벤트의 등록과 해제

■ off() 메서드

- on() 메서드에 의해 선택된 요소에 연결된 이벤트 중 인자로 지정된 이벤트를 제거함

```
$("#요소").off( [이벤트] );
```

- off() 메서드가 인자로 이벤트이름을 가질 경우
 - 해당 객체에 연결되어 있는 모든 이벤트들 중 off() 메서드의 인자로 지정된 이벤트를 제거함
- off() 메서드가 인자를 가지지 않을 경우
 - 해당 객체에 on() 메서드를 통해 연결되어 있는 모든 이벤트들을 제거함

이벤트의 등록과 해제

7-on-1.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .myclass {
      background:blue;
      color:white;
    }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $('ul li').on('click', function() {
        alert($(this).text());
      });

      $('ul li').on( {
        mouseenter: function() { $(this).addClass('myclass');},
        mouseleave: function() { $(this).removeClass('myclass');}
      });

      $('#btn1').click( function() {
        $('ul li').off('click');
      });
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li>1</li>
      <li>2</li>
      <li>3</li>
      <li>4</li>
      <li>5</li>
      <li>6</li>
      <li>7</li>
    </ul>
  </div>
  <div>
    <button type="button">7개 모두 해제</button>
  </div>
</body>
</html>
```

이벤트의 등록과 해제

```
        $('#btn2').click( function() {
            $('ul li').off();
        });

    });
</script>
</head>

<body>

    <ul id="nav">
        <li>item-1</li>
        <li>item-2</li>
    </ul>

    <button id="btn1">click 이벤트 제거</button>
    <button id="btn2">모든 이벤트 제거</button>

</body>
</html>
```

- item-1
- item-2

click 이벤트 제거

모든 이벤트 제거

이벤트의 등록과 해제

■ on() 메서드의 문제점

- 동적으로 생성되는 객체에 대해 이벤트를 연결하지 못함
 - 현재 존재하는 객체(정적인 객체)에 대해서만 이벤트 적용 가능
 - 동적으로 생성되는 동일한 객체에 대해서는 적용되지 않음
- [예] 버튼에 클릭 이벤트를 연결할 경우
 - 이미 생성되어 있는 버튼에는 클릭 이벤트 연결 가능 (정상 동작)
 - 동적으로 생성되는 새로운 버튼에는 클릭 이벤트가 적용되지 않음

이벤트의 등록과 해제

7-on-2.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <style type="text/css">
    * { margin:0px; padding:0px; }
    div {
      margin:10px; padding:10px; width:300px;
      border:3px Solid Black; border-radius:5px;
    }
  </style>

  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $('div').on('click', function () {
        $(this).clone().appendTo('body');
      });

    });
  </script>
</head>
```


이벤트의 등록과 해제

```
<body>

  <div>
    <h1>한국소프트웨어</h1>
    <p> 홈페이지에 오신 것을 환영합니다.</p>
  </div>

</body>
</html>
```

한국소프트웨어

홈페이지에 오신 것을 환영합니다.

한국소프트웨어

홈페이지에 오신 것을 환영합니다.

새로 생성되는 요소에는 click 이벤트가 복사되지 않음

이벤트의 등록과 해제

7-on-3.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .myclass { background:blue; color:white; }
  </style>

  <script src="script/jquery-3.5.1.js"></script>

  <script type="text/javascript">
    $(function(){

      $('#apnd').click( function(index) {
        $('#nav').append("<li>item-3</li>");
      });
      $('ul li').on('click', function() {
        alert($(this).text());
      });
    });
  </script>
</head>
```

이벤트의 등록과 해제

```
<body>

  <ul id="nav">
    <li>item-1</li>
    <li>item-2</li>
  </ul>

  <button id="apnd">adding item-3</button>

</body>
</html>
```

- item-1
- item-2

adding item-3

이벤트의 등록과 해제

■ 동적 객체에 대한 해결 방안

```
$("요소").on( 이벤트, 객체, function() (  
    이벤트 핸들러의 내용  
});
```

- 객체
 - 이벤트의 대상이 되는 요소
 - 요소
 - 객체의 상위 요소
- 요소를 동적으로 생성하지 않는 경우에는 이전 형식을 사용해도 무관함

이벤트의 등록과 해제

7-on-4.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">

  <style type="text/css">
    * { margin:0px; padding:0px; }
    div {
      margin:10px; padding:10px; width:300px;
      border:3px Solid Black; border-radius:5px;
    }
  </style>

  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $('body').on('click', 'div', function () {
        $(this).clone().appendTo('body');
      });

    });
  </script>
</head>
```

이벤트의 등록과 해제

```
<body>

  <div>
    <h1>한국소프트웨어</h1>
    <p>홈페이지에 오신 것을 환영합니다.</p>
  </div>

</body>
</html>
```

한국소프트웨어

홈페이지에 오신 것을 환영합니다.

한국소프트웨어

홈페이지에 오신 것을 환영합니다.

이벤트의 등록과 해제

7-on-5.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .myclass { background:blue; color:white; }
  </style>

  <script src="script/jquery-3.5.1.js"></script>

  <script type="text/javascript">
    $(function(){

      $('#apnd').click( function(index) {
        $('#nav').append("<li>item-3</li>");
      });
      $('ul').on('click', 'li', function() {
        alert($(this).text());
      });

    });
  </script>
</head>
```

이벤트의 등록과 해제

```
<body>

  <ul id="nav">
    <li>item-1</li>
    <li>item-2</li>
  </ul>

  <button id="apnd">adding item-3</button>

</body>
</html>
```

- item-1
- item-2

adding item-3

이벤트의 등록과 해제

■ off() 메서드

- on() 메서드에 의해 연결된 이벤트를 제거

```
$("#요소").off( [이벤트] )
```

- off() 메서드가 인자로 이벤트이름을 가질 경우
 - 선택된 객체에 연결되어 있는 이벤트들 중 on() 메서드의 인자로 지정된 이벤트를 제거함
- off() 메서드가 인자를 가지지 않을 경우
 - 선택자에 의해 선택된 객체에 on() 메서드를 통해 연결되어 있는 모든 이벤트들을 제거함

이벤트의 등록과 해제

■ one() 메서드

- 선택된 요소에 이벤트를 한번만 연결함

```
$(“요소”).off( [이벤트] )
```

- 이벤트의 수행 후 이벤트를 자동 소멸되므로 별도의 제거 메서드는 없음

이벤트의 등록과 해제

7-one.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .myclass { background:black; color:white; }
  </style>

  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function() {

      $('ul li').one('click', function() {
        alert($(this).text());
      });

    });
  </script>
</head>
<body>
  <ul id="nav">
    <li>item-1</li>
    <li>item-2</li>
  </ul>
</body>
</html>
```

- item-1
- item-2

localhost:8080 내용:

item-1

확인

localhost:8080 내용:

item-2

확인

이벤트의 등록과 해제

■ trigger() 메서드

- 특정 요소에 연결된 이벤트를 강제로 발생할 수 있는 기능 수행

```
$(“요소”).trigger( 이벤트 )
```

- 다른 객체에 연결된 이벤트를 실행 할 수 있음
 - [예] [버튼A]를 클릭하여 [버튼B] 연결되어 있는 이벤트를 강제로 실행할 수 있음
- 요소
 - 수행하고자 하는 이벤트가 지정되어 있는 요소
- 이벤트
 - 해당 요소에 지정되어 있는 이벤트

이벤트의 등록과 해제

7-trigger.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    div {
      width:200px;
      height:200px;
      background:red;
      border-radius: 10px;
    }
    .myclass { background:blue;    }}
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $('div#test1').click( function() {
        $(this).toggleClass('myclass');
      });
      $('button').click( function() {
        $('div#test1').trigger('click');
      });
    });
  </script>
</head>
```

이벤트의 등록과 해제

```
<body>
```

```
  <div id="test1"> click </div>
```

```
  <br>
```

```
  <button>trigger</button>
```

```
</body>
```

```
</html>
```



<div>를 클릭하거나 [trigger] 버튼을 클릭해도 색상이 변경됨

이벤트의 종류

문서 로딩 이벤트

■ 문서 로딩 이벤트 메서드

- 브라우저에서 HTML 문서를 불러올 때 객체(요소)의 로딩/loading이 완료되면 발생

메서드	설명
load()	외부에서 연동되는 특정 이미지나 콘텐츠가 로드되면 이벤트를 발생시킴 - jQuery의 버전업 과정에서 제거됨
ready()	문서 내의 요소가 로드 되었을 때 이벤트를 발생시킴 - 문서 준비 이벤트

문서 로딩 이벤트

■ ready() 이벤트 메서드

- 문서에 존재하는 모든 요소가 완전히 로드 되었을 때 인자로 지정한 콜백 함수를 수행

```
$("document").ready( callback function )
```

- DOM 요소가 완전히 로드되었을 때 수행되어야 하는 jQuery 코드를 콜백함수로 구현
- 콜백함수를 구성하는 jQuery 코드가 <head> 태그 내에 존재할 경우 반드시 사용해야 함

■ load() 이벤트 메서드

- jQuery 버전3에서는 완전히 제거됨

문서 로딩 이벤트

7-ready.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(document).ready(function(){
      $("#wrap1").append("<p>문서 로딩 완료</p>");
    });

  </script>
</head>

<body>

  <div id="wrap1"></div>
  

</body>
</html>
```

문서 로딩 완료



마우스 이벤트

■ 마우스 관련 이벤트 메서드

• 마우스의 각종 동작에 관련된 이벤트

메서드	설명
click()	요소를 마우스로 클릭했을 때 발생하는 click이벤트에 의해 실행되는 메서드
dblclick()	요소를 마우스로 더블클릭 했을 때 발생하는 dblclick이벤트에 의해 실행되는 메서드
mouseout()	요소 위에서 마우스의 포인터가 벗어났을 때 실행되는 이벤트 메서드
mouseover()	요소 위로 마우스의 포인터가 올라갔을 때 실행되는 이벤트 메서드
hover()	요소 위로 마우스 포인터가 올라갔을 때와 벗어났을 때 실행되는 이벤트 메서드
mousedown()	요소 위에서 마우스의 포인터를 눌렀을 때 실행되는 이벤트 메서드
mouseup()	요소 위에서 마우스의 포인터를 눌렀다 놓았을 때 실행되는 이벤트 메서드
mouseenter()	요소 위로 마우스의 포인터가 올라갔을 때 실행되는 이벤트 메서드
mouseleave()	요소 위에서 마우스의 포인터가 벗어났을 때 실행되는 이벤트 메서드
mousemove()	요소 위에서 마우스의 포인터가 이동되었을 때 실행되는 이벤트 메서드
Scroll()	가로 세로 스크롤 바를 움직일 때 실행되는 이벤트 메서드

마우스 이벤트

■ click() 이벤트 메서드와 dbclick() 이벤트 메서드

• click() 이벤트 메서드

```
$("#요소").click( 이벤트핸들러 )
```

- 선택된 요소를 마우스로 클릭했을 때 발생하는 click이벤트에 의해 실행되는 메서드
- 인자로 콜백 함수를 가질 때 인자로 지정된 이벤트핸들러가 수행됨
 - 인자를 가지지 않는 경우 클릭 이벤트를 강제로 발생시킴 (인위적으로 마우스를 클릭하는 효과)

• dbclick() 이벤트 메서드

```
$("#요소").dbclick( 이벤트핸들러 )
```

- 선택된 요소를 마우스로 더블 클릭했을 때 발생하는 dbclick이벤트에 의해 실행되는 메서드
- 인자로 콜백 함수를 가질 때 인자로 지정된 이벤트핸들러가 수행됨
 - 인자를 가지지 않는 경우 더블 클릭 이벤트를 강제로 발생시킴 (인위적으로 마우스를 더블 클릭하는 효과)

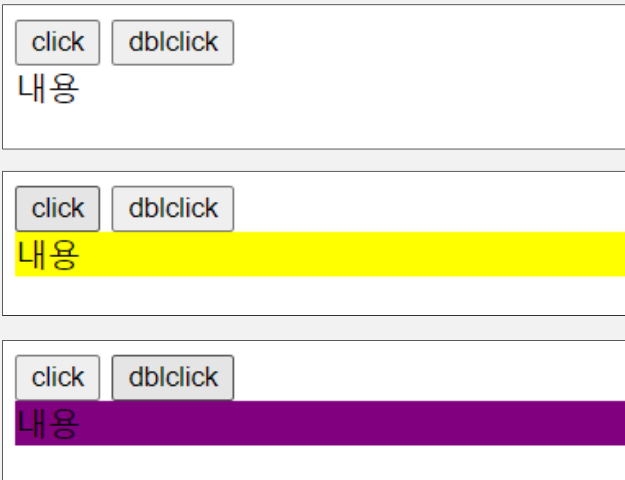
마우스 이벤트

7-event-click-dblclick.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
    div { width:200px; }
  </script>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#btn1").click(function(){
        $("#div").css("background-color","yellow");
      });
      $("#btn2").dblclick(function(){
        $("#div").css("background-color","purple");
      });

    });
  </script>
</head>
<body>
  <button id="btn1">click</button>
  <button id="btn2">dblclick</button>
  <div>내용</div>
</body>
</html>
```



마우스 이벤트 (기본 이벤트 차단)

■ 기본 이벤트 차단

- 이벤트 객체를 사용해 요소의 기본 기능을 제거하는 것
- `preventDefault()` 메서드를 사용
 - `preventDefault()` 메서드를 사용하기 위해서는 이벤트 객체를 생성하고, 이 이벤트 객체에 적용해야 함
 - [예] 하이퍼링크의 기본 기능인 주소 이동의 기능을 제거함

```
$("#요소").click( function(e) {  
    e.preventDefault();  
    실행문  
});
```

```
$("#btn2").click(function(e){  
    e.preventDefault();  
    $("p").css("background-color","yellow");  
});
```

마우스 이벤트 (기본 이벤트 차단)

7-event-preventDefault-1.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $(".btn1").on("click", function(e){
        e.preventDefault( );
        $(".txt1")
          .css( {"background-color":"#ff0"} );
      });
      $(".btn2").on("click", function(e){
        $(".txt2")
          .css( {"background-color":"#0ff"} );
      });
      $(".btn3").on("dblclick", function(){
        $(".txt3")
          .css( {"background-color":"#0f0"} );
      });

    });
  </script>
</head>
```

마우스 이벤트 (기본 이벤트 차단)

```
<body>
```

```
<p><a href="http://www.easyspub.co.kr/" class="btn1">버튼1</a></p>
```

```
<p class="txt1">내용1</p>
```

```
<p><a href="http://www.easyspub.co.kr/" class="btn2">버튼2</a></p>
```

```
<p class="txt2">내용2</p>
```

```
<p><button class="btn3">버튼3</button></p>
```

```
<p class="txt3">내용3</p>
```

```
</body>
```

```
</html>
```

버튼1

내용1

버튼2

내용2

버튼3

내용3

마우스 이벤트 (기본 이벤트 차단)

■ 기본 실행을 막는 다른 방법

- return 값으로 false를 지정하는 방법

```
$("#요소").click( function(e) {  
    실행문  
    return false;  
});
```

- return 문은 자비스크립트를 강제적으로 종료시키는 구문이므로 실행문의 마지막에 사용
- 요소의 기본 기능을 차단하기 위해서는 return false보다는 가급적 preventDefault를 사용

마우스 이벤트 (기본 이벤트 차단)

7-event-preventDefault-2.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    div { width:200px }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#btn1").click(function(){
        $("div").css("background-color","purple");
        return false;
      });
    });
  </script>
</head>
<body>
  <a href="http://www.naver.com" id="btn1">버튼1</a>
  <div>내용</div>
</body>
</html>
```

[버튼1](#)

내용

[버튼1](#)

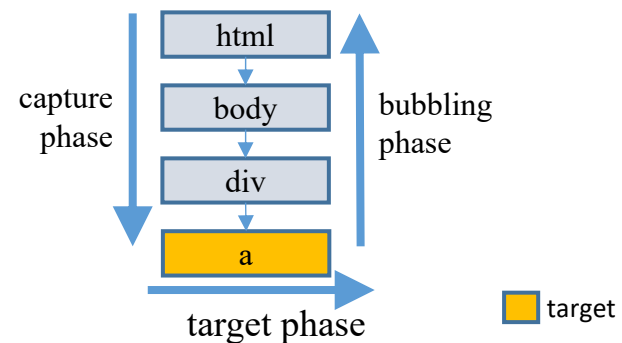
내용

마우스 이벤트 (이벤트의 흐름과 차단)

■ 이벤트의 흐름

- 여러 요소가 중첩된 구조에서 이벤트가 발생할 경우 중첩된 구조 사이로 이벤트가 전달되는 현상
 - Capture phase, target phase, bubbling phase의 세 단계로 구성됨

```
<html>
  <body>
    <div>
      <a href="#">test</a>
    </div>
  </body>
</html>
```



- Capture phase : 최상위 부모 요소에서 target 요소 전까지의 단계
 - Target phase : 현재 클릭한 요소
 - Bubbling phase : target 요소부터 최상위 부모 요소까지의 단계
- jQuery에서는 bubbling phase만을 지원함

마우스 이벤트 (이벤트의 흐름과 차단)

■ 이벤트 전달의 예

```
<html>
  <body>
    <div id="t1">
      <div id="t2"> </div>
    </div>
  </body>
</html>
```

- <DIV#t1>에서 클릭 이벤트가 발생할 경우
 - 발생한 클릭 이벤트는 <DIV#t2>에 전달되지 않음
 - jQuery는 Capture phase를 지원하지 않기 때문
- <DIV#t1>에서 클릭 이벤트가 발생할 경우
 - 발생한 클릭 이벤트는 <DIV#t1>에 전달됨
 - jQuery는 bubbling phase를 지원하기 때문

마우스 이벤트 (이벤트의 흐름과 차단)

■ 이벤트의 흐름 차단

- 이벤트의 흐름이 진행되지 못하게 차단하는 것
 - 이벤트의 흐름은 비정상적인 상황을 발생시킬 수 있기 때문임
- 이벤트를 차단하기 위해서는 이벤트객체와 함께 `stopPropagation()` 메서드를 사용

```
$("#요소").click( function(e) {  
    실행문  
    e.stopPropagation();  
});
```

- `click` 이벤트는 버블링이 차단되어 상위 요소로 전달되지 않음

마우스 이벤트 (이벤트의 흐름과 차단)

7-event-stopPropagation.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">

    div#box1 {
      width:60px; height:60px;
      background:#999;
      padding:20px 20px 20px 20px;
    }

    div#box2 {
      display:block;
      width:60px; height:60px;
      background:#0F0;
      border:#000 thin solid;
    }

  </style>

  <script src="script/jquery-3.5.1.js"></script>
```

마우스 이벤트 (이벤트의 흐름과 차단)

```
<script type="text/javascript">

    $(function(){

        $("div#box1").click(function(e) {
            $(this).css("background","yellow");
            alert("바깥쪽 div를 클릭하면 출력되는 대화상자입니다.");
        });

        $("div#box2").click(function(e) {
            $(this).css("background","yellow");
            alert("안쪽 div를 클릭하면 출력되는 대화상자입니다.");

            //이벤트 흐름을 정지하여 더 이상 부모 요소에 전달되지 않음
            //stopPropagation()을 주석처리하면 이벤트가 div로 전달됨

            e.stopPropagation();
        });

    });

</script>
</head>
```

마우스 이벤트 (이벤트의 흐름과 차단)

```
<body>
```

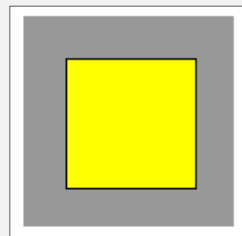
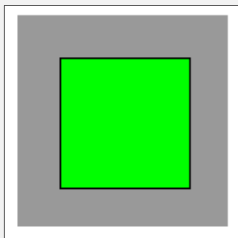
```
  <div id="box1">
```

```
    <div id="box2"></div>
```

```
  </div>
```

```
</body>
```

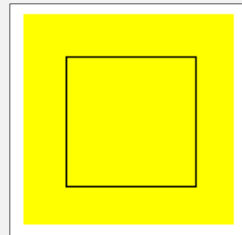
```
</html>
```



localhost:8080 내용:

안쪽 div를 클릭하면 출력되는 대화상자입니다

확인



localhost:8080 내용:

바깥쪽 div를 클릭하면 출력되는 대화상자입니다.

확인

마우스 이벤트

■ mouseover() 이벤트 메서드와 mouseout() 이벤트 메서드

• mouseover() 이벤트 메서드

```
$(“요소”).mouseover( 이벤트핸들러 )
```

- 선택된 요소의 영역으로 마우스 포인터가 올라갔을 때 실행되는 이벤트 메서드
- 주로 mouseout() 이벤트와 함께 사용
 - 버블링을 사용하기 때문 거의 사용하지 않는 추세임

• mouseout() 이벤트 메서드

```
$(“요소”).mouseout( 이벤트핸들러 )
```

- 선택된 요소의 영역에서 마우스 포인터가 벗어났을 때 실행되는 이벤트 메서드
 - 버블링을 사용하기 때문 거의 사용하지 않는 추세임

마우스 이벤트

■ mouseenter() 이벤트 메서드와 mouseleave() 이벤트 메서드

• mouseenter() 이벤트 메서드

```
$("#요소").mouseenter( 이벤트핸들러 )
```

- 선택된 요소의 영역으로 마우스 포인터가 올라갔을 때 실행되는 이벤트 메서드
 - mouseleave() 이벤트와 함께 사용
 - 버블링을 지원하지 않으므로 주로 사용됨

• mouseleave() 이벤트 메서드

```
$("#요소").mouseleave( 이벤트핸들러 )
```

- 선택된 요소의 영역에서 마우스 포인터가 벗어났을 때 실행되는 이벤트 메서드
 - 주로 버블링을 지원하지 않으므로 주로 사용됨

마우스 이벤트

7-event-mouseenter-leave.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">

    div.out {
      width:200px; height:200px;
      margin:0 15px;
      background-color:#D6EDFC;
      float:left;
    }
    div.in {
      width:100px; height:100px;
      background-color:#FFCC00;
      margin:10px auto;
    }
    p {
      line-height:1em;
      margin:0; padding:0;
    }

  </style>
  <script src="script/jquery-3.5.1.js"></script>
```

마우스 이벤트

```
<script type="text/javascript">

    $(function(){

        var i = 0;
        $("div.overout").mouseover(function() {
            i += 1;
            $(this).find("span").text( "mouse over x " + i );
        }).mouseout(function() {
            $(this).find("span").text("mouse out ");
        });

        var n = 0;
        $("div.enterleave").mouseenter(function() {
            n += 1;
            $(this).find("span").text( "mouse enter x " + n );
        }).mouseleave(function() {
            $(this).find("span").text("mouse leave");
        });

    });

</script>
</head>
```

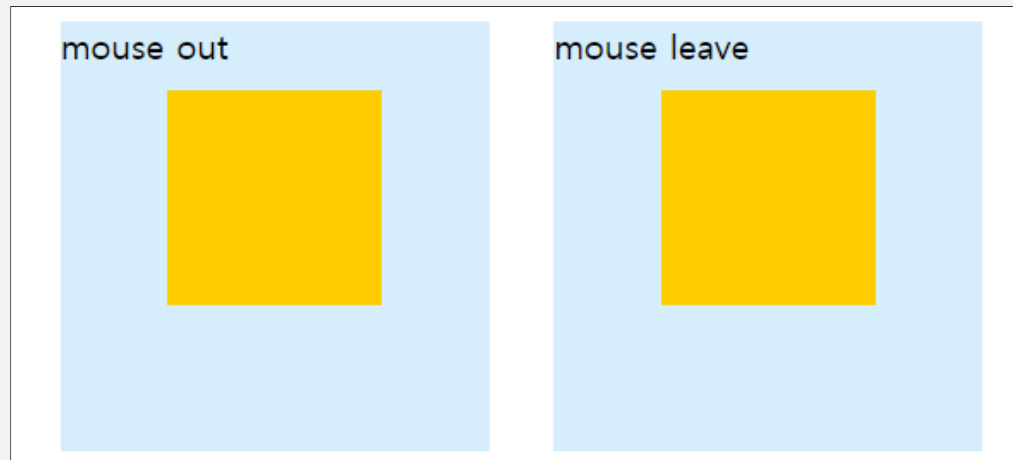
마우스 이벤트

```
<body>

  <div class="out overout">
    <span>move your mouse</span>
    <div class="in"> </div>
  </div>

  <div class="out enterleave">
    <span>move your mouse</span>
    <div class="in"> </div>
  </div>

</body>
</html>
```



마우스 이벤트

■ mousemove() 이벤트 메서드

- 마우스 포인터가 좌우로 이동될 때 발생하는 실행되는 이벤트 메서드

```
$(“요소”).mousemove( 이벤트핸들러 )
```

마우스 이벤트

7-event-mousemove.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $(document).on("mousemove", function(e) {
        $(".posX").text(e.pageX);
        $(".posY").text(e.pageY);
      });

    });
  </script>
</head>

<body>

  <h1>mousemove</h1>
  <p>X : <span class="posX">0</span>px</p>
  <p>Y : <span class="posY">0</span>px</p>

</body>
</html>
```

mousemove

X : 227px

Y : 120px

마우스 이벤트

■ hover() 메서드

- 요소 위로 마우스를 올리거나 내릴 때 실행되는 이벤트 메서드

```
$(“요소”).hover( 이벤트핸들러1, 이벤트핸들러-2 )
```

- mouseenter 이벤트의 핸들러와 mouseleave 이벤트의 핸들러를 동시에 인자로 가짐
- 이벤트 핸들러 1
 - 마우스가 들어갈 때 발생하는 이벤트
 - 내부적으로 mouseenter() 메서드가 사용됨
- 이벤트 핸들러 2
 - 마우스가 떠날 때 발생하는 이벤트
 - 내부적으로 mouseleave() 메서드가 사용됨

마우스 이벤트

7-event-hover.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .myclass1 { background:red;    color:white; }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $('ul li').hover(function(){
        $(this).addClass('myclass1');
      },
      function(){
        $(this).removeClass('myclass1');
      });
    });
  </script>
</head>
<body>
  <ul>
    <li>KOREA</li>
    <li>JAPAN</li>
    <li>CHINA</li>
  </ul>
</body>
</html>
```

- KOREA
- JAPAN
- CHINA

마우스 이벤트

■ mousedown() 이벤트 메서드와 mouseup() 이벤트 메서드

• mousedown() 이벤트 메서드

```
$(“요소”).mousedown( 이벤트핸들러 )
```

- 선택된 요소를 마우스의 버튼이 눌렸을 때 인자로 지정된 이벤트 핸들러가 수행됨
 - 이벤트핸들러는 마우스 버튼을 놓을 때까지 지속적으로 수행됨

• mouseup() 이벤트 메서드

```
$(“요소”).mouseup( 이벤트핸들러 )
```

- 선택된 요소를 마우스의 버튼으로 눌렀다 놓았을 때 인자로 지정된 이벤트핸들러가 수행됨

마우스 이벤트

7-event-mousedown-up.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("button").mousedown(function() {
        $("div").text("마우스가 눌러져 있습니다");
      })
      .mouseup(function() {
        $("div").text("마우스를 놓았습니다");
      });

    });
  </script>
</head>

<body>
  <button>click me!</button>
  <br>
  <div></div>
</body>
</html>
```

click me!

마우스가 눌러져 있습니다

click me!

마우스를 놓았습니다

마우스 이벤트

■ scroll 이벤트 메서드

- 스크롤 바가 이동될 때 실행되는 이벤트 메서드

```
$(“요소”).scroll( 이벤트핸들러 )
```

- 스크롤 바의 위치 정보 추출

- scrollTop()
 - 스크롤 바의 상하 이동 위치
- scrollLeft()
 - 스크롤 바의 좌우 이동 위치

마우스 이벤트

7-event-scroll.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    body{
      height:10000px;    width:5000px;
    }
    #wrap{
      position: fixed;
      left: 10px; top: 10px;
    }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(window).on("scroll",function(){

      var sc_top = Math.round( $(this).scrollTop() );
      var sc_left = Math.round( $(this).scrollLeft() );

      $(".top").text(sc_top);
      $(".left").text(sc_left);

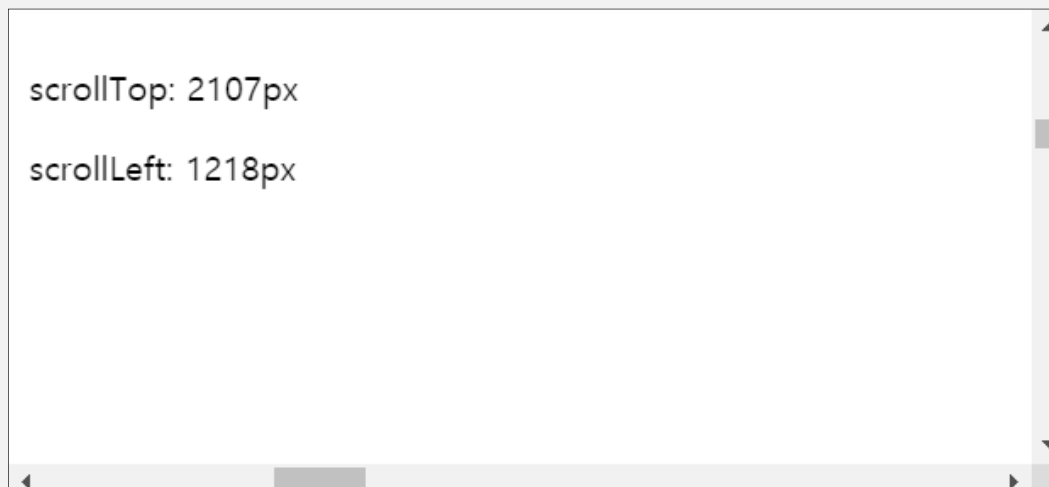
    });
  </script>
</head>
```

마우스 이벤트

```
<body>

  <div id="wrap">
    <p>scrollTop: <span class="top">0</span>px</p>
    <p>scrollLeft: <span class="left">0</span>px</p>
  </div>

</body>
</html>
```



포커스 이벤트

■ 포커스 관련 이벤트 메서드

- 마우스의 클릭과 같은 동작으로 선택된 요소가 마우스 포인터를 얻거나 잃었을 때 발생하는 이벤트

메서드	설명
focus()	요소가 포커스를 얻었을 때 발생 (버블링이 발생하지 않음)
blur()	요소가 포커스를 잃을 때 발생 (버블링이 발생하지 않음)
focusin()	요소가 포커스를 얻었을 때 발생 (버블링이 발생)
focusout()	요소가 포커스를 잃을 때 발생 (버블링이 발생)

포커스 이벤트

■ focus()와 blur() 이벤트 메서드

- 포커스를 얻었을 때 (focus())와 포커스를 잃었을 때 (blur()) 실행되는 이벤트 메서드
- 버블링이 발생되지 않음
 - 요소에서 발생한 이벤트가 상위 요소로 전달되지 않음

■ focusin()와 focusout() 이벤트 메서드

- 포커스를 얻었을 때 (focusin())와 포커스를 잃었을 때 (focusout()) 실행되는 이벤트 메서드
- 버블링이 발생함
 - 요소에서 발생한 이벤트가 상위 요소로 전달됨

포커스 이벤트

7-event-focusin-out.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("div#outer").focusin( function() {
        $("div#state1").text("div에서 발생한 focus 이벤트입니다.");

      });

      $("input#userid").focusin(function() {
        $(this).css({"background-color":"yellow"});
        $("div#state1").text("입력상자가 포커스를 얻었습니다");
      }).focusout( function() {
        $(this).css({"background-color":"white"});
        $("div#state1").text("입력상자가 포커스를 잃었습니다");
      });

    });
  </script>
</head>
```

포커스 이벤트

```
<body>

  <p>이름을 입력하세요</p>
  <div id='outer'>
    <input id="userid" type="text" size=20>
  </div>
  <div id='state1'></div>
</body>

</html>
```

이름을 입력하세요

div에서 발생한 focus 이벤트입니다.

이름을 입력하세요

입력상자가 포커스를 잃었습니다

포커스 이벤트

7-event-focus-blur.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("div#outer").focus( function() {
        $("div#state1").text("div에서 발생한 focus 이벤트입니다.");

      });

      $("input#userid").focus(function() {
        $(this).css({"background-color":"yellow"});
        $("div#state1").text("입력상자가 포커스를 얻었습니다");
      }).blur( function() {
        $(this).css({"background-color":"white"});
        $("div#state1").text("입력상자가 포커스를 잃었습니다");
      });

    });
  </script>
</head>
```

포커스 이벤트

```
<body>

  <p>이름을 입력하세요</p>
  <div id='outer'>
    <input id="userid" type="text" size=20>
  </div>
  <div id='state1'></div>
</body>

</html>
```

이름을 입력하세요

입력상자가 포커스를 얻었습니다

이름을 입력하세요

입력상자가 포커스를 잃었습니다

키보드 이벤트

■ 키보드 이벤트 메서드

• 키보드의 입력에 따라 발생하는 이벤트

메서드	설명
keypress()	선택한 요소에서 키보드 키를 눌렀을 때 실행되는 이벤트 메서드
keydown()	선택한 요소에서 키보드 키를 눌렀을 때 실행되는 이벤트 메서드
keyup()	선택한 요소에서 키보드 키를 눌렀다 놓았을 때 실행되는 이벤트 메서드

키보드 이벤트

■ 키보드 접근성

- 마우스가 없는 상황에서 키보드를 사용해 동작할 수 있도록 보장하는 것
 - [예] 마우스가 없어도 키보드 탭(tab) 키를 사용해 onmouseover, onmouseout 이벤트가 정상적으로 수행
- 마우스를 대신하는 키보드 대응 이벤트

마우스 메서드	키보드 이벤트
mouseover	focus
mouseout	blur

```
$('.btn').on("mouseover focus" function( ) {  
    자바스크립트(제이쿼리) 코드;  
});
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

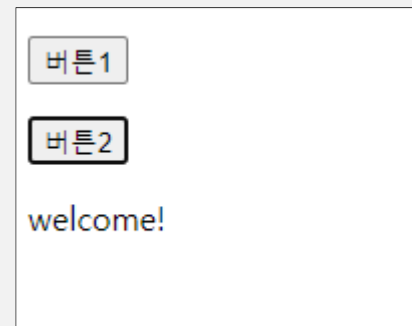
    $(function(){

      $("#btn1")
      .data({"text":"javascript"})
      .on({
        "mouseover": overFnc,
        "mouseout": outFnc
      });

      $("#btn2")
      .data({"text":"welcome!"})
      .on({
        "mouseover focus": overFnc,
        "mouseout blur": outFnc
      });
```

키보드 이벤트

```
        function overFnc() {  
            $(".txt").text($(this).data("text"));  
        }  
  
        function outFnc() {  
            $(".txt").text("");  
        }  
  
    });  
</script>  
</head>  
  
<body>  
  
    <p><button id="btn1">버튼1</button></p>  
    <p><button id="btn2">버튼2</button></p>  
    <p class="txt"></p>  
</body>  
  
</html>
```



키보드 이벤트

■ keydown() 이벤트 메서드

- keydown 이벤트가 발생되었을 때 실행되는 이벤트 메서드
 - 이벤트가 발생되면 인자로 지정한 콜백 함수를 실행함

```
$("#요소").keydown( 이벤트핸들러 )
```

- Keydown() 이벤트 메서드의 특징
 - Ctrl, Alt, Shift 키를 포함한 모든 키의 키 코드를 추출할 수 있음
 - 알파벳의 경우 대소문자 구분하지 않고 같은 키 코드를 출력

키보드 이벤트

■ Keypress() 이벤트 메서드

- keypress 이벤트가 발생되었을 때 실행되는 이벤트 메서드
 - 이벤트가 발생되면 인자로 지정한 콜백 함수를 실행함

```
$("#요소").keypress( 이벤트핸들러 )
```

- Keypress() 이벤트 메서드의 특징
 - Ctrl, Alt, Shift 키의 키 코드를 추출하지 못함
 - 알파벳의 경우 대소문자 구분하여 서로 다른 키 코드를 출력
 - 한글을 인식하지 못함

키보드 이벤트

7-event-keydown-press.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#user_id").keydown(function(e){
        $("#txt1").text(e.keyCode);
      });

      $("#user_id").keypress(function(e){
        $("#txt2").text(e.keyCode);
      });

    });
  </script>
</head>
<body>
  <p><input type="text" name="user_id" id="user_id"></p>
  <p>keydown : <span id="txt1"></span></p>
  <p>keypress : <span id="txt2"></span></p>
</body>
</html>
```

a

keydown : 65

keypress : 97

A

keydown : 65

keypress : 65

keydown : 17

keypress : ctrl키 입력

ㅁ

keydown : 229

keypress :

키보드 이벤트

■ keyup() 메서드

- Keyup 이벤트가 발생되었을 때 수행되는 이벤트 메서드

```
$(“요소”).keyup( 이벤트핸들러 )
```

- 한글을 처리할 때 주로 사용되는 메서드

- 키보드 관련 이벤트의 발생 순서

- 1. 사용자가 키보드를 누름
- 2. keydown 이벤트가 발생
- 3. 글자가 입력됨
- 4. keypress 이벤트가 발생
- 5. 사용자가 키보드에서 손을 뗌
- 6. keyup 이벤트가 발생

- keypress() 메서드는 한글을 처리할 수 없음

- keydown() 메서드도 사용 불가

- keydown 이벤트는 글자가 입력되기 전에 발생하기 때문

- 주로 keyup() 메서드를 사용

키보드 이벤트

7-event-keyup.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $('textarea').keyup(function () {

        var inputLength = $(this).val().length;
        var remain = 150 - inputLength;

        $('h1').html(remain);
        if (remain >= 0) {
          $('h1').css('color', 'black');
        } else {
          $('h1').css('color', 'red');
        }
      });

    });
  </script>
</head>
```

키보드 이벤트

```
<body>
  <div>
    <p>글을 입력하세요(최대 150글자 입니다)</p>
    <h1>150</h1>
    <textarea cols="70"rows="5"></textarea>
  </div>
```

```
</body>
</html>
```

글을 입력하세요(최대 150글자 입니다)

133

keyPress 이벤트가 아니라

글을 입력하세요(최대 150글자 입니다)

-16

keyPress 이벤트가 아니라 KeyDown 이벤트입니다. keyPress 이벤트가 아니라 KeyDown 이벤트입니다. keyPress 이벤트가 아니라 KeyDown 이벤트입니다. keyPress 이벤트가 아니라 KeyDown 이벤트입니다. keyPress 이벤트가 아니라 KeyDown 이벤트입니다.

키보드 이벤트

7-event-keyboard.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $(document).on("keydown", keyEventFnc);
      function keyEventFnc(e) {
        var direct = "";


        switch(e.keyCode){
          case 37: direct = "LEFT"; break;
          case 38: direct = "TOP"; break;
          case 39: direct = "RIGHT"; break;
          case 40: direct = "BOTTOM"; break;
        }

        if(direct) $("#user_id").val(direct);
      }

    });
  </script>
</head>
```

키보드 이벤트

```
<body>  
  <div>  
    <p><input type="text" name="user_id" id="user_id"></p>  
  </div>  
</body>  
</html>
```



RIGHT

키보드의 방향키 입력

키보드 이벤트

■ change() 이벤트 메서드

- 선택한 <form> 요소의 값이 새로운 값으로 변경될 때 수행되는 메서드
 - 포커스가 선택한 요소에서 다른 요소로 이동할 때 이벤트를 발생시킴

```
$("#요소").change( 이벤트핸들러 )
```

키보드 이벤트

7-event-change-1.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#rel_site").change(function(){
        $("#txt").text($(this).val());
      });

    });
  </script>
</head>
<body>
  <select id="rel_site">
    <option value="">사이트 선택</option>
    <option value="www.naver.com">네이버</option>
    <option value="www.nate.com">네이트</option>
  </select>
  <p id="txt"></p>
</body>
</html>
```



A screenshot of a web browser window. At the top, there is a dropdown menu with the text '네이버' (Naver) and a downward arrow. Below the dropdown, the text 'www.naver.com' is displayed.

키보드 이벤트

7-event-change-2.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $('#all-check').change(function() {

        if (this.checked) {
          $('#check-item').children().prop('checked', true);
        } else {
          $('#check-item').children().prop('checked', false);
        }

      });

    });
  </script>
</head>
```

키보드 이벤트

```
<body>

  <input type="checkbox" id="all-check" />

  <label>All</label>

  <div id="check-item">

    <input type="checkbox" />
    <label>A Option</label>

    <input type="checkbox" />
    <label>B Option</label>

    <input type="checkbox" />
    <label>C Option</label>
  </div>

</body>
</html>
```

☐ All
☐ A Option ☐ B Option ☐ C Option

☒ All
☒ A Option ☒ B Option ☒ C Option

이벤트가 발생한 요소 추적

■ \$(this) 선택자와 index() 메서드

- \$(this) 선택자
 - 선택된 요소 자신을 나타내는 선택자
 - this 선택자를 사용하면 어느 요소에서 이벤트가 발생하는지 추적할 수 있음
- index() 메서드
 - 선택된 요소 중 이벤트가 발생한 요소의 인덱스를 반환

```
$("#요소").index( )
```

이벤트가 발생한 요소 추적

7-event-this.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .line { border: 2px solid red }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#btn").mouseenter(function(){
        $(this).addClass('line');
        $(this).text("MouseOver");
      })
      .mouseleave(function(){
        $(this).removeClass('line');
        $(this).text("마우스 오버");
      })
    });
  </script>
</head>
```

이벤트가 발생한 요소 추적

```
<body>  
  <button id="btn">마우스 오버</button>  
</body>  
</html>
```

마우스 오버

MouseOver

이벤트가 발생한 요소 추적

7-event-index.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#menu a").mouseover(function(){
        $("#menu a").css("background","none");
        $(this).css("background","yellow");
        var num=$("#menu a").index(this);
        $(".txt").text(num);

      });

    });
  </script>
</head>
<body>
  <ul id="menu">
    <li><a href="#">내용1</a></li>
    <li><a href="#">내용2</a></li>
    <li><a href="#">내용3</a></li>
  </ul>
  <p class="txt"></p>
</body>
</html>
```

- [내용1](#)
- [내용2](#)
- [내용3](#)

2

수고하셨습니다