





Events

jQuery - 이벤트

Loading

jQuery - 로딩

.ready() 이벤트 핸들러 함수

ready() 함수는 이미 배웠던 것처럼 HTML 문서의 내용을 읽어들이고 후 실행됩니다.



.ready() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    //jQuery code  
});
```

```
$().ready(function() {  
    //jQuery code  
});
```

```
$(function() {  
    //jQuery code  
});
```

.load() 이벤트 핸들러 함수

load() 함수는 이미 배웠던 것처럼 HTML 문서와 연결된 내용(이미지)을 읽어들이고 후 실행됩니다.

.load() 이벤트 핸들러 함수

```
$(window).load(function() {  
    //문서의 내용을 모두 가져온 후 실행됩니다.  
});
```




.unload() 이벤트 핸들러 함수

unload() 함수는 윈도우 창을 닫은 직후, 실행됩니다.



.unload() 이벤트 핸들러 함수

```
$(window).unload(function() {  
    //윈도우 창을 닫으면 실행합니다.  
});
```

Browser

jQuery - 브라우저 이벤트

.resize() 이벤트 핸들러 함수

resize() 함수는 윈도우 창 크기를 변경할 경우, 실행됩니다.



.resize() 이벤트 핸들러 함수

```
$(window).resize(function() {  
    //윈도우 창의 사이즈를 변경하면 실행합니다.  
});
```



.scroll() 이벤트 핸들러 함수

scroll() 함수는 윈도우 창을 스크롤 할 경우, 실행됩니다.



.scroll() 이벤트 핸들러 함수

```
$(window).scroll(function() {  
    //윈도우 창을 움직이면 실행합니다.  
});
```

.error() 이벤트 핸들러 함수

error() 함수는 에러가 발생한 경우 실행됩니다.

.error() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('img').error(function(){  
        $(this).hide();  
    });  
});
```

Mouse

jQuery - 마우스 이벤트



.click() 이벤트 핸들러 함수

click() 함수는 요소를 클릭할 경우 발생합니다.



.click() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').click(function() {  
        $(this).after('<p>단락 생성</p>');  
    });  
});
```



.click() 실행 함수

```
$(document).ready(function() {  
    $('p').click();  
});
```

.dblclick() 이벤트 핸들러 함수

dblclick() 함수는 요소를 더블클릭한 경우 발생합니다.



.dblclick() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').dblclick(function() {  
        $(this).after('<p>단락 생성</p>');  
    });  
});
```

.dblclick() 실행 함수

```
$(document).ready(function() {  
    $('p').dblclick();  
});
```


.mouseover() 이벤트 핸들러 함수

mouseover() 함수는 요소에 마우스가 올라간 경우 발생합니다.

.mouseover() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mouseover(function() {  
        $(this).addClass('selected');  
    });  
});
```

.mouseout() 이벤트 핸들러 함수

`mouseout()` 함수는 요소에서 마우스가 벗어난 후 발생합니다.

.mouseout() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mouseout(function() {  
        $(this).removeClass('selected');  
    });  
});
```

.mousemove() 이벤트 핸들러 함수

mousemove() 함수는 요소에서 마우스가 움직일 경우 발생합니다.

.mousemove() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mousemove(function(e) {  
        var page_coords = '(' + e.pageX + ',' + e.pageY + ')';  
        var client_coords = '(' + e.clientX + ',' + e.clientY + ')';  
        $('span:first', this).text('page_coords - ' + page_coords);  
        $('span:last', this).text('client_coords - ' + client_coords);  
    });  
});
```

.mousedown() 이벤트 핸들러 함수

mousedown() 함수는 요소를 마우스로 누를 때 발생합니다.

.mousedown() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mousedown(function() {  
        $(this).addClass('selected');  
    });  
});
```


.mouseup() 이벤트 핸들러 함수

mouseup() 함수는 요소를 마우스로 눌렀다 떼 후 발생합니다.

.mouseup() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mouseup(function() {  
        $(this).removeClass('selected');  
    });  
});
```

.mouseenter() 이벤트 핸들러 함수

mouseenter() 함수는 요소에 마우스가 들어간 경우 발생합니다.
이벤트 버블링의 영향을 받지 않습니다.

.mouseenter() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mouseenter(function() {  
        $(this).find('span').text('mouseenter');  
    });  
});
```

.mouseleave() 이벤트 핸들러 함수

mouseleave() 함수는 요소에서 마우스가 나올때 경우 발생합니다.
이벤트 버블링의 영향을 받지 않습니다.

.mouseleave() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').mouseleave(function() {  
        $(this).find('span').text('mouseleave');  
    });  
});
```

.hover() 이벤트 핸들러 함수

hover() 함수는 요소에서 마우스가 올라갈 때, 벗어날 때 두가지 상황에서 발생합니다.

.hover(inFn, outFn) 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').hover(function() {  
        $(this).addClass('selected'); // 마우스 오버  
    }, function() {  
        $(this).removeClass('selected'); // 마우스 아웃  
    });  
});
```


.hover(inOutFn) 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').hover(function() {  
        $(this).toggleClass('selected'); // 마우스 오버, 아웃  
    });  
});
```

.toggle() 이벤트 핸들러 함수

toggle() 함수는 요소를 마우스로 누를때마다 발생합니다.

.toggle(fn1, fn2, ...) 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').toggle(function() {  
        $(this).addClass('selected1');  
    }, function() {  
        $(this).addClass('selected2');  
    }, function() {  
        $(this).addClass('selected3');  
    });  
});
```

.toggle([duration]) 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p:first').toggle();  
    $('p:eq(2)').toggle('slow');  
    $('p:last').toggle(450);  
});
```

Keyboard

jQuery - 키보드 이벤트

.keydown() 이벤트 핸들러 함수

keydown() 함수는 키보드로 눌렀을 때 발생합니다.

.keydown() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').keydown(function() {  
        // 키 누를 때 발생  
    });  
});
```

.keypress() 이벤트 핸들러 함수

keypress() 함수는 키보드 눌렀을 때 발생합니다.

.keypress() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').keypress(function() {  
        // 키 다운 후, 발생  
    });  
});
```



.keyup() 이벤트 핸들러 함수

keyup() 함수는 키보드를 눌렀다 뗐을 때 발생합니다.

.keyup() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').keyup(function() {  
        // 키 누르고 땔 때 발생  
    });  
});
```

Form

jQuery - 폼(양식) 이벤트

.focus() 이벤트 핸들러 함수

focus() 함수는 폼 요소에 포커스된 상황에서 발생합니다.

.focus() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('#input#test').focus(function() {  
        $(this).val('포커스가 되었습니다.');    });  
});
```

.focus() 실행 함수

```
$(document).ready(function() {  
    $('input#test').focus(function() {  
        $(this).val('포커스가 되었습니다.');    });  
    $('#test').focus();  
});
```

.blur() 이벤트 핸들러 함수

blur() 함수는 폼 요소에 블러(포커스를 벗어난)된 상황에서 발생합니다.

.blur() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('input#test').blur(function() {  
        $(this).val('블러 되었습니다.');    });  
});
```

.blur() 실행 함수

```
$(document).ready(function() {  
    $('input#test').blur(function() {  
        $(this).val('블러 되었습니다.');    });  
    $('#test').blur();  
});
```

.focusin() 이벤트 핸들러 함수

focusin() 함수는 요소에서 포커스-인 된 상황에서 발생(버블링)합니다.

.focusin() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').focusin(function() {  
        $(this).find('span').css('display','inline').fadeOut(800);  
    });  
});
```

.focusout() 이벤트 핸들러 함수

focusout() 함수는 요소에서 포커스-아웃 된 상황에서 발생(버블링)합니다.

.focusout() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').focusout(function() {  
        $(this).find('span').text('focus out').css('display','inline').fadeOut(800);  
    });  
});
```



.change() 이벤트 핸들러 함수

change() 함수는 select의 option이 변경된 상황에서 발생합니다.

.change() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('select.selection').change(function() {  
        var msg = "";  
        $('option:selected', this).each({  
            msg += $(this).text() + ' ';  
        });  
        $('<div/>').text(msg);  
    });  
});
```


.change() 실행 함수

```
$(document).ready(function() {  
    $('select.selection').change(function() {  
        var msg = "";  
        $('option:selected', this).each({  
            msg += $(this).text() + ' ';  
        });  
        $('<div/>').text(msg);  
    }).change();  
});
```



.select() 이벤트 핸들러 함수

select() 함수는 input, textfield를 마우스로 드래그 선택할 경우 발생합니다.

.select() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $(document).select(function() {  
        $('<div />').text('select 이벤트 발생');  
    });  
});
```



.select() 실행 함수

```
$(document).ready(function() {  
    $('input').select();  
});
```



.submit() 이벤트 핸들러 함수

submit() 함수는 form 요소의 내용을 제출할 경우 발생합니다.

.submit() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('#forms').submit(function() {  
        alert('폼 양식을 제출했습니다.');        return false;  
    });  
});
```

.submit() 실행 함수

```
$(document).ready(function() {  
    $('#forms').submit(function() {  
        alert('폼 양식을 제출했습니다.');        return false;  
    }).submit();  
});
```

E-Handling

jQuery - 이벤트 핸들링

.bind() 이벤트 핸들러 함수

bind() 함수는 자바스크립트의 이벤트 핸들러를 연결하는 것이라고 이해하시면 됩니다.

.bind() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').bind('click', function() {  
        $(this).after('<p>클릭</p>');  
    });  
    $('#footer address').bind('mouseover', function() {  
        $(this).after('<p>마우스 오버</p>');  
    });  
});
```

.unbind() 이벤트 핸들러 함수

unbind() 함수는 자바스크립트의 이벤트 핸들러를 연결한 것을 해제하는 것이라고 이해하시면 됩니다.

.unbind() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p:last').bind('click', function() {  
        $('p').unbind('click');  
    });  
});
```



.one() 이벤트 핸들러 함수

one() 함수는 단 한번! 자바스크립트의 이벤트 핸들러를 실행하는 것이라고 이해하시면 됩니다.



.one() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p:last').one('click', function() {  
        $('p').addClass('one');  
    });  
});
```

.trigger() 이벤트 핸들러 함수

trigger() 함수는 ‘방아쇠’라는 의미 그대로 대상에 이벤트를 엮어주는 것
이라고 이해하시면 됩니다.

.trigger() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p:last').one('click', function() {  
        $('div').trigger('dblclick');  
    });  
});
```


.triggerHandler() 이벤트 핸들러 함수

trigger() 함수와 비슷해보이지만 실제 이벤트를 걸어준다기 보다는
핸들링된 함수만 실행하는 역할을 수행합니다.



.trigger() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p:last').one('click', function() {  
        $('input').triggerHandler('focus');  
    });  
});
```

Live

jQuery - 라이브 이벤트

.live() 이벤트 핸들러 함수

live() 함수는 bind() 이벤트 함수와 유사하나, 한가지 다른 점은 현재 문서 요소 뿐만 아니라, 동적으로 생성된 요소에도 이벤트가 살아서 적용된다는 것입니다.

.live() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('p').live('click', function() {  
        $(this).after('<p>클릭</p>');  
    });  
    $('#footer address').live('mouseover', function() {  
        $(this).after('<p>마우스 오버</p>');  
    });  
});
```

.die() 이벤트 핸들러 함수

die() 함수는 unbind() 이벤트 함수와 비슷하며, lie() 함수를 제거할 때 사용합니다.
인수(전달인자)가 없으면 모든 lie() 함수에 엮인 이벤트를 제거합니다.



.die() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('a.die').live('click', function() {  
        $('p').die('click');  
    });  
});
```

delegate

jQuery - 델리게이트 이벤트

.delegate() 이벤트 핸들러 함수

delegate() 함수는 live() 이벤트를 대체하는 새로운 메소드로,
동적으로 생성된 요소에도 이벤트가 살아서 적용될 뿐, 아니라 개별(each)적으로도
코드 적용이 가능하여 매우 강력한 이벤트 핸들러입니다.

.delegate() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('table').each(function() {  
        $('td').live('click', function() {  
            $(this).toggleClass('live');  
        });  
    });  
  
    $('table').delegate('td', 'click', function() {  
        $(this).toggleClass('delegate');  
    });  
});
```

.undelegate() 이벤트 핸들러 함수

undelegate() 함수는 die() 이벤트를 대체하며, delegate()가 엮인 이벤트를 제거하는 역할을 수행합니다.

.undelegate() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('table').delegate('td', 'click', function() {  
        $(this).dblclick(function(){  
            $(this).undelegate('td', 'click');  
        })  
    });  
});
```

on

jQuery - on 이벤트

.on() 이벤트 핸들러 함수

on() 함수는 bind(), live(), delegate() 이벤트를 대체하는 v1.7+의 새로운 메소드로, 하나 이상의 이벤트를 대상에 연결하는 매우 강력한 이벤트 핸들러입니다.

.on() vs .delegate()

```
$(document).ready(function() {  
    $('table').delegate('td', 'click', function() {  
        $(this).toggleClass('delegate');  
    });  
});
```

```
$('table').on('click', 'td', function() {  
    $(this).toggleClass('delegate');  
});  
});
```

.on() 전달인자에 따라 변경

```
$(document).ready(function() {  
    $('table td').on('click', function() {  
        $(this).toggleClass('delegate');  
    });  
  
    $('table').on('click', 'td', function() {  
        $(this).toggleClass('delegate');  
    });  
});
```


.on() 이벤트 데이터 전달

```
$(document).ready(function() {  
    $('table td').on('click', { key: 'value' }, function() {  
        $(this).toggleClass('delegate');  
    });  
});
```

.on() 이벤트 맵 처리 가능

```
$(document).ready(function() {  
  $('table td').on({  
    'click': function() {  
      $(this).toggleClass('click');  
    },  
    'hover': function() {  
      $(this).toggleClass('hover');  
    }  
  });  
});
```

.undelegate() 이벤트 핸들러 함수

undelegate() 함수는 die() 이벤트를 대체하며, delegate()가 엮인 이벤트를 제거하는 역할을 수행합니다.

.undelegate() 이벤트 핸들러 함수

```
$(document).ready(function() {  
    $('table').delegate('td', 'click', function() {  
        $(this).dblclick(function(){  
            $(this).undelegate('td', 'click');  
        })  
    });  
});
```