



16 이벤트

모던 웹을 위한 Javascript jQuery 입문

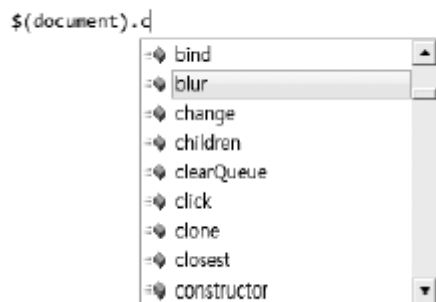
❖ jQuery로 이벤트 다루기

- 보조 기능을 사용해 어떠한 메서드가 이벤트와 관련되는지 살펴보기

코드 16-1 \$(document).ready() 메서드

```
$(document).ready(function (event) {  
  
});
```

그림 16-1 jQuery의 이벤트 메서드



❖ jQuery 사용해 이벤트 연결

- 가장 기본적인 방법은 bind() 메서드 사용
- bind() 메서드는 다음과 같은 두 가지 형태로 사용
 - 1 \$(selector).bind(eventName, function(event) { })
 - 2 \$(selector).bind(object)

코드 16-2 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```



❖ \$(selector).bind(eventName, function(event) { })

- bind() 메서드의 첫 번째 매개 변수에 이벤트 이름 입력
- 두 번째 매개 변수에 이벤트 핸들러 입력
- 이벤트 핸들러 안에서 this 키워드는 이벤트 발생한 객체 의미
- 예제 코드 16-3]
 - h1 태그를 click 이벤트에 연결
 - 이벤트 발생 시 이벤트 발생 객체에 '+' 글자 추가

코드 16-3 bind() 메서드(1)

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').bind('click', function () {
        $(this).html(function (index, html) {
            return html + '+';
        });
    });
});
</script>
```



❖ 2 \$(selector).bind(object)

- bind() 메서드의 매개 변수에 객체 넣어줌
- 속성 이름과 속성값에 이벤트 이름과 이벤트 핸들러 넣음
- 예제 코드 16-4]
 - 마우스가 진입하면 reverse 클래스 속성 추가
 - 마우스가 빠져나가면 reverse 클래스 속성 제거

코드 16-4 bind() 메서드(2)

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('h1').bind('click', function () {
            $(this).html(function (index, html) {
                return html + '+';
            });
        });

        // 이벤트를 연결합니다.
        $('h1').bind({
            mouseenter: function () { $(this).addClass('reverse'); },
            mouseleave: function () { $(this).removeClass('reverse'); }
        });
    });
</script>
```



❖ Style 태그 입력

코드 16-5 style 태그 구성

```
<style>
  .reverse {
    background: Black;
    color: White;
  }
</style>
```

그림 16-2 이벤트 연결

Header-0+++

Header-1+++++

Header-2+



16.2 간단한 이벤트 연결

❖ jQuery 단축 이벤트 메서드

■ 사용법

- `$(selector).method(function(event) { });`

표 16-2 jQuery 단축 이벤트 메서드

blur	focus	focusin	focusout	load
resize	scroll	unload	click	dblclick
mousedown	mouseup	mousemove	mouseover	mouseout
mouseenter	mouseleave	change	select	submit
keydown	keypress	keyup	error	ready

표 16-3 jQuery의 메서드

메서드 이름	설명
<code>toggle()</code>	click 이벤트를 여러 이벤트 핸들러를 번갈아가며 실행할 수 있게 연결합니다.
<code>hover()</code>	mouseenter 이벤트와 mouseleave 이벤트를 동시에 연결합니다.



16.2 간단한 이벤트 연결

❖ jQuery 단축 이벤트 메서드 구현

■ HTML 페이지 구성

코드 16-6 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .reverse {
      background: Black;
      color: White;
    }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js" />
  <script>
    $(document).ready(function () {

      });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```



❖ hover() 메서드

- 두 개의 함수를 매개 변수로 넣어줌
 - 첫 번째 매개 변수에 넣는 함수는 mouseenter 이벤트 핸들러
 - 두 번째 매개 변수에 넣는 함수는 mouseleave 이벤트 핸들러

코드 16-7 hover() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').hover(function () {
        $(this).addClass('reverse');
    }, function () {
        $(this).removeClass('reverse');
    });
});
</script>
```



❖ toggle() 메서드

- 매개 변수에 여러 개의 함수 입력 가능 (click 이벤트 핸들러와 관련)

코드 16-8 toggle() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').hover(function () {
        $(this).addClass('reverse');
    }, function () {
        $(this).removeClass('reverse');
    });

    // 이벤트를 연결합니다.
    $('h1').toggle(function () {
        // 문자열의 앞에 별을 추가합니다.
        $(this).html(function (index, html) {
            return '★' + html;
        });
    }, function () {
        // 문자열의 뒤에 별을 추가합니다.
        $(this).html(function (index, html) {
            return html + '★';
        });
    }, function () {
        // 문자열의 별을 제거합니다.
        $(this).html(function (index, html) {
            return html.substring(1, html.length - 1);
        });
    });
});
</script>
```

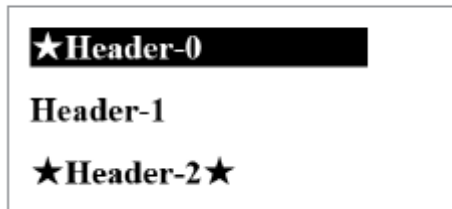


❖ 이벤트 발생 순서

- 1. 문자열의 앞에 별 추가
- 2. 문자열의 뒤에 별 추가
- 3. 문자열의 별 제거

❖ 실행 결과

그림 16-3 실행 결과



❖ 이벤트 제거 시 unbind() 메서드 사용

■ unbind() 메서드의 사용 형태

- 1 \$(selector).unbind() - 해당 문서 객체와 관련된 모든 이벤트 제거
- 2 \$(selector).unbind(eventName) - 특정 이벤트와 관련된 모든 이벤트 제거
- 3 \$(selector).unbind(eventName, function) - 특정 이벤트 핸들러 제거

표 16-4 jQuery의 메서드

메서드 이름	설명
unbind()	이벤트를 제거합니다.

■ body 태그 구성

코드 16-9 body 태그 구성

```
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
```



16.3 이벤트 연결 제거

❖ 예제 코드 16-10]

- h1 태그에 click 이벤트 연결하고 이벤트 발생 알람
- 이후에 unbind() 메서드 사용해 이벤트 제거

코드 16-10 unbind() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').click(function () {
        // 출력합니다.
        $(this).html('CLICK');
        alert('이벤트가 발생했습니다.');
```



```
        // 이벤트를 제거합니다.
        $(this).unbind();

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

- 같은 기능 하는 one() 메서드 존재

표 16-5 jQuery의 메서드

메서드 이름	설명
one()	이벤트를 한번만 연결합니다.



16.4 매개 변수 context

❖ 매개 변수 context

- jQuery 메서드는 매개 변수 두 개 입력 가능
- selector 매개 변수 이외에 context 매개 변수 가능
- 매개 변수 context = selector 적용하는 범위 한정

그림 16-4 jQuery 메서드의 형태

```
$(  
  $(selector, context)
```

그림 16-5 예제

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Header 2

Vestibulum porttitor pellentesque est, ut sollicitudin turpis blandit ac.

Header 3

Etiam consequat est vitae felis blandit tincidunt.



❖ 매개 변수 context 예제

- 특정 부분에 선택자 적용하고 싶을 때 사용하는 매개 변수
- Body 태그 구성

코드 16-11 body 태그 구성

```
<body>
  <div>
    <h1>Header 1</h1>
    <p>Paragraph</p>
  </div>
  <div>
    <h1>Header 2</h1>
    <p>Paragraph</p>
  </div>
  <div>
    <h1>Header 3</h1>
    <p>Paragraph</p>
  </div>
</body>
```



❖ 매개 변수 context 예제

■ Style 태그 구성

코드 16-12 style 태그 구성

```
<style>
  * { margin:0px; padding:0px }
  div
  {
    margin:5px; padding:3px;

    border-width:3px;
    border-style:solid;
    border-color:Black;
    border-radius:10px;
  }
</style>
```



❖ 매개 변수 context 예제

■ Context 객체 사용

코드 16-13 context 객체 사용

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('div').click(function () {
        // 변수를 선언합니다.
        var header = $('h1', this).text();
        var paragraph = $('p', this).text();

        // 출력합니다.
        alert(header + '\n' + paragraph);
    });
});
</script>
```



❖ jQuery가 이벤트 객체 정형화

- jQuery의 이벤트 객체는 모든 브라우저가 같은 방법으로 사용
- 같은 속성을 가짐

❖ 자주 사용하는 jQuery 이벤트 객체의 속성

표 16-6 jQuery 이벤트 객체의 속성

이벤트 객체 속성	설명
event.pageX	브라우저의 화면을 기준으로 한 마우스의 X 좌표 위치
event.pageY	브라우저의 화면을 기준으로 한 마우스의 Y 좌표 위치
event.preventDefault()	기본 이벤트를 제거합니다.
event.stopPropagation()	이벤트 전달을 제거합니다.



❖ 이벤트 객체 활용 예제

- HTML5 Canvas 사용하는 예제 작성
 - canvas 태그에 id 속성 입력
 - width 속성 과 height 속성 지정
- canvas 태그는 HTML5부터 지원하는 태그
 - 인터넷 익스플로러 8 이하에서는 지원 불가
- body 태그 구성

코드 16-14 body 태그 구성

```
<body>  
  <canvas id="canvas" width="500" height="300">  
  
  </canvas>  
</body>
```



❖ 이벤트 객체 활용 예제

- Document 객체의 getElementById() 메서드 사용
 - canvas 태그에 id 속성 부여

코드 16-15 2d context 객체 추출과 이벤트 연결

```
<script>
$(document).ready(function () {
    // 변수를 선언합니다.
    var canvas = document.getElementById('canvas');
    var context = canvas.getContext('2d');

    // 이벤트를 연결합니다.
    $(canvas).toggle(function (event) {

    }, function (event) {

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



❖ 예제 코드 16-16]

- 문서 객체와 마우스 커서의 상대 위치 구하는 방법
 - 이벤트 객체에서 pageX 속성과 pageY 속성 사용

코드 16-16 jQuery 이벤트 객체

```
// 이벤트를 연결합니다.
$(canvas).toggle(function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;
}, function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;
});
```



❖ context 객체의 beginPath(), moveTo(),.lineTo(), stroke() 메서드 사용

- canvas 객체 에 선 그릴 수 있음

코드 16-17 Canvas를 사용한 선 그리기

```
// 이벤트를 연결합니다.
$(canvas).toggle(function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;

    // 선을 그립니다.
    context.beginPath();
    context.moveTo(x, y);

}, function (event) {
    // 위치를 얻어냅니다.
    var position = $(this).offset();
    var x = event.pageX - position.left;
    var y = event.pageY - position.top;

    // 선을 그립니다.
    context.lineTo(x, y);
    context.stroke();
});
```

그림 16-6 실행 결과



❖ 이벤트 강제 발생

■ trigger() 메서드 사용 예

- 1 \$(selector).trigger(eventName)
- 2 \$(selector).trigger(eventName, data)

– 2번 형태의 매개 변수 data는 일반적으로 배열

표 16-7 jQuery의 메서드

메서드 이름	설명
trigger()	이벤트를 강제로 발생시킵니다.

■ 1번 형태 예제 Body 태그 구성

코드 16-18 body 태그 구성

```
<body>
  <h1>Start: </h1>
  <h1>Start: </h1>
</body>
```



❖ trigger() 메서드

- h1 태그에 click 이벤트 연결
 - 클릭하면 뒤에 별 추가
 - setInterval() 함수 사용

– 마지막에 위치한 h1 태그는 자동으로 1초마다 클릭 이벤트 실행

코드 16-19 trigger() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').click(function () {
        $(this).html(function (index, html) {
            return html + '★';
        });
    });

    // 1초마다 함수를 실행합니다.
    setInterval(function () {
        $('h1').last().trigger('click');
    }, 1000);
});
</script>
```

그림 16-7 실행 결과

Start: ★★★★★★★★★★
Start: ★★★★★



16.6 이벤트 강제 발생

❖ 간단한 방식으로 이벤트 강제 실행

코드 16-20 간단한 이벤트 강제 실행

```
// 1초마다 함수를 실행합니다.  
setInterval(function () {  
    $('h1').last().click();  
}, 1000);
```



❖ trigger() 메서드의 2번 형태 실행

- 2 \$(selector).trigger(eventName, data)

코드 16-21 trigger() 메서드를 사용한 매개 변수 전달

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').click(function (event, data1, data2) {
        alert(data1 + ' : ' + data2);
    });

    // 이벤트를 강제로 발생시킵니다.
    $('h1').eq(1).trigger('click', [273, 52]);
});
</script>
```



❖ 기본 이벤트 제거, 이벤트 전달을 막는 메서드

표 16-8 이벤트 객체의 메서드

메서드 이름	설명
preventDefault()	기본 이벤트를 제거합니다.
stopPropagation()	이벤트 전달을 제거합니다.

코드 16-22 body 태그 구성

```
<body>
  <h1>
    <a href="http://hanb.co.kr">Hanb Media</a>
  </h1>
</body>
```

코드 16-23 style 태그 구성

```
<style>
  * {
    margin:5px; padding:5px;
    border:3px solid Black;
  }
</style>
```



❖ 예제 코드 16-24]

- preventDefault() 메서드
 - H1 태그 눌렀을 때 자동 실행되는 a 태그 기본 이벤트 제거

코드 16-24 preventDefault() 메서드

```
<script>
$(document).ready(function () {
    $('a').click(function (event) {
        $(this).css('background-color', 'Blue');
        event.preventDefault();
    });

    $('h1').click(function () {
        $(this).css('background-color', 'Red');
    });
});
</script>
```

그림 16-8 anchor 태그를 클릭했을 경우의 실행 결과



❖ 예제 코드 16-25]

- 이벤트 객체의 stopPropagation() 메서드 사용

코드 16-25 stopPropagation() 메서드

```
$('#a').click(function (event) {  
    $(this).css('background-color', 'Blue');  
    event.stopPropagation();  
    event.preventDefault();  
});
```

그림 16-9 anchor 태그를 클릭했을 경우의 실행 결과



❖ 예제 코드 16-26]

- stopPropagation() 메서드와 preventDefault() 메서드 함께 사용
- jQuery는 return false 사용
 - 이 두 가지 함께 적용하는 것으로 인식

코드 16-26 return false

```
$('#a').click(function (event) {  
    $(this).css('background-color', 'Blue');  
    return false;  
});
```



16.8 이벤트 연결의 종류

❖ jQuery의 이벤트 연결 메서드

표 16-9 jQuery의 이벤트 연결 메서드

메서드 이름	설명
bind()	현재 존재하는 문서 객체에만 이벤트를 연결합니다.
delegate()	현재 또는 미래에 존재하는 문서 객체에 이벤트를 연결합니다.
live()	현재 또는 미래에 존재하는 문서 객체에 이벤트를 연결합니다.
one()	한번만 이벤트에 연결합니다.

❖ jQuery의 이벤트 제거 메서드

표 16-10 이벤트 연결 제거 관련 메서드

메서드 이름	설명
unbind()	bind() 메서드를 사용해 연결한 이벤트를 제거합니다.
die()	live() 메서드를 사용해 연결한 이벤트를 제거합니다.
undelegate()	delegate() 메서드를 사용해 연결한 이벤트를 제거합니다.



❖ 이벤트 제거 메서드 비교 예제

■ Body 태그 작성

코드 16-27 body 태그 구성

```
<body>
  <div id="wrap">
    <h1>Header</h1>
  </div>
</body>
```

■ bind() 메서드를 사용한 이벤트 연결

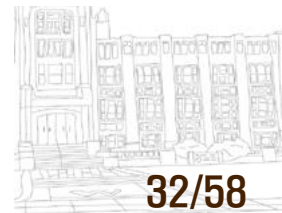
코드 16-28 bind() 메서드를 사용한 이벤트 연결

```
<script>
  $(document).ready(function () {
    $('h1').bind('click', function () {
      var length = $('h1').length;
      var targetHTML = $(this).html();
      $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
    });
  });
</script>
```

그림 16-10 Header를 클릭하면 생성됩니다.

Header

- 1 - Header**
- 2 - Header**
- 3 - Header**



❖ 이벤트 제거 메서드 비교 예제

■ live() 메서드

- 현재 존재하는 h1 태그 + 앞으로 계속 생성하는 h1 태그에도 이벤트 연결

코드 16-29 live() 메서드를 사용한 이벤트 연결

```
<script>
    $(document).ready(function () {
        $('h1').live('click', function () {
            var length = $('h1').length;
            var targetHTML = $(this).html();
            $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
        });
    });
</script>
```

그림 16-11 어떠한 h1 태그를 선택해도 요소가 추가됨

Header

1 - Header

2 - 1 - Header

3 - 2 - 1 - Header

4 - 1 - Header



❖ 이벤트 제거 메서드 비교 예제

■ Die() 메서드

• 사용 방식

- 1 **\$(selector).die()**
- 2 **\$(selector).die(eventName)**
- 3 **\$(selector).die(eventName, function)**

- 매개 변수 지정하지 않으면 해당 선택자와 관련된 모든 이벤트 제거

코드 16-30 die() 메서드

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').live('click', function () {
        var length = $('h1').length;
        var targetHTML = $(this).html();
        $('#wrap').append('<h1>' + length + ' - ' + targetHTML + '</h1>');
    });

    // 5초 이후에는 모든 이벤트가 제거됩니다.
    setTimeout(function () {
        $('h1').die();
    }, 5000);
});
</script>
```



❖ jQuery 1.7의 이벤트 연결 메서드

표 16-11 jQuery 1.7의 이벤트 연결 메서드

메서드 이름	설명
on()	이벤트를 연결합니다.
off()	이벤트를 제거합니다.

- on() 메서드의 기존 이벤트 연결 메서드 대체
 - 1 bind(): \$(selector).on(eventName, eventHandler)
 - 2 live(): \$(document).on(eventName, selector, eventHandler)
 - 3 delegate(): \$(selector).on(eventName, selector, eventHandler)
- off() 메서드의 기존 이벤트 연결 메서드 대체
 - 이벤트의 모든 이벤트 핸들러 제거하고 싶을 때
 - 매개 변수 **eventHandler** 생략
 - 1 unbind(): \$(selector).off(eventName[, eventHandler])
 - 2 die(): \$(document).off(eventName, selector[, eventHandler])
 - 3 undelegate(): \$(selector).off(eventName, selector[, eventHandler])

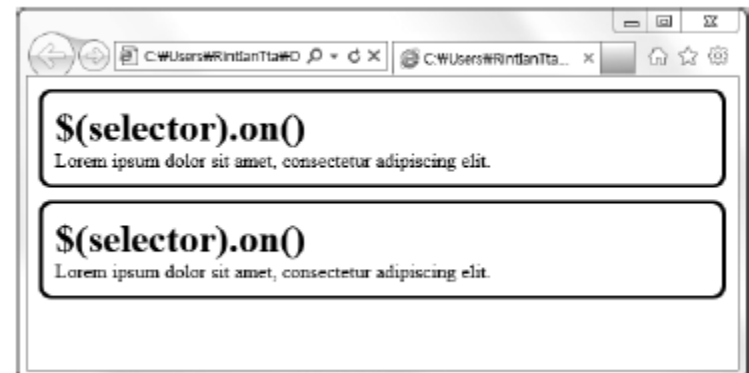


❖ on() 메서드의 bind() 메서드 대체 형태

코드 16-31 on() 메서드의 bind() 메서드 대체 형태

```
<!DOCTYPE html>
<html>
<head>
  <style>
    * { margin:0px; padding:0px;}
    div {
      margin:10px; padding:10px;
      border:3px Solid Black;
      border-radius:10px;
    }
  </style>
  <script src="http://code.jquery.com/jquery-1.7b1.js"×/script>
  <script>
    $(document).ready(function () {
      $('div').on('click', function () {
        $(this).clone().appendTo('body');
      });
    });
  </script>
</head>
<body>
<div>
  <h1>$(selector).on()</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</div>
</body>
</html>
```

그림 16-12 bind() 메서드의 대체 형태



❖ on() 메서드의 delegate() 메서드 대체 형태

- 선택자 부분에서 이벤트 지속 범위 한정
 - 첫 번째 매개 변수에 이벤트 이름
 - 두 번째 매개 변수에 이벤트가 지정될 선택자 입력
 - 세 번째 매개 변수에 이벤트 핸들러 지정

코드 16-32 on() 메서드의 delegate() 메서드 대체 형태

```
<script>
  $(document).ready(function () {
    $('body').on('click', 'div', function () {
      $(this).clone().appendTo('body');
    });
  });
</script>
```



❖ 이벤트 지속 범위 한정

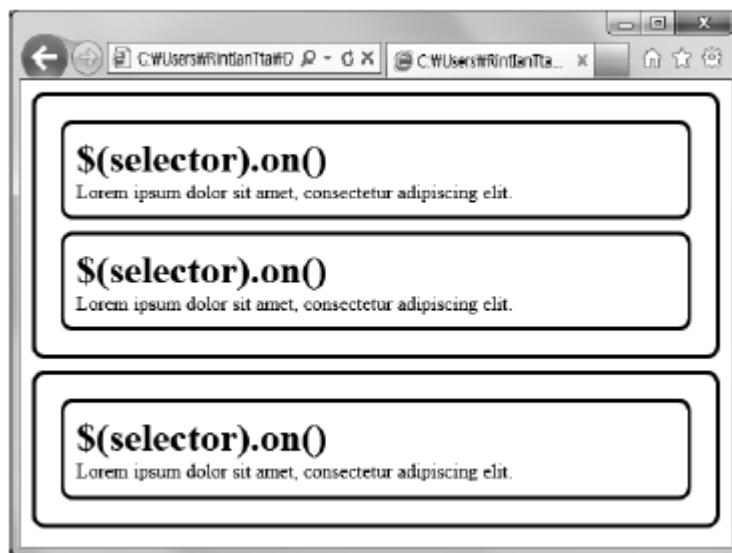
■ 예제 코드 16-33

- id 속성을 upper_div로 지정한 div 태그, lower_div로 지정한 div 태그
- on() 메서드를 delegate() 메서드의 대체 형태로 사용
- 선택자 부분에서 이벤트 지속 범위를 id 속성을 upper_div로 지정한 태그로 한정

코드 16-33 이벤트 범위 한정

```
<script>
$(document).ready(function () {
    $('#upper_div').on('click', 'div', function () {
        $(this).clone().appendTo('#upper_div');
        $(this).clone().appendTo('#lower_div');
    });
});
</script>
```

그림 16-13 delegate() 메서드의 대체 형태



❖ delegate() 메서드

- 이벤트의 지속 범위 한정

❖ live() 메서드

- 문서 전체가 이벤트의 지속 범위
- 예제 코드 16-34
 - 선택자에 document 객체 지정

코드 16-34 on() 메서드의 live() 메서드 대체 형태

```
<script>
    $(document).ready(function () {
        $(document).on('click', 'div', function () {
            $(this).clone().appendTo( '#upper_div' );
            $(this).clone().appendTo( '#lower_div' );
        });
    });
</script>
```



❖ 마우스 이벤트

표 16-12 마우스 이벤트

이벤트 이름	설명
click	마우스를 클릭할 때 발생합니다.
dblclick	마우스를 더블클릭할 때 발생합니다.
mousedown	마우스 버튼을 누를 때 발생합니다.
mouseup	마우스 버튼을 떼를 때 발생합니다.
mouseenter	마우스가 요소의 경계 외부에서 내부로 이동할 때 발생합니다.
mouseleave	마우스가 요소의 경계 내부에서 외부로 이동할 때 발생합니다.
mousemove	마우스를 움직일 때 발생합니다.
mouseout	마우스가 요소를 벗어날 때 발생합니다.
mouseover	마우스를 요소 안에 들어올 때 발생합니다.



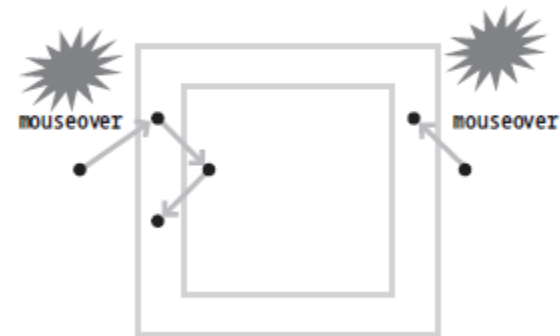
❖ 예제 코드 16-35]

- div 태그를 두 개 중첩해서 만든 후
- 외부 div 태그에 mouseover 이벤트와 mouseenter 이벤트 모두 연결

코드 16-35 mouseover와 mouseenter 이벤트의 차이

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .outer { width:200px; height:200px; background:Orange; padding:50px; margin:10px;}
    .inner { width:100%; height:100%; background:Pink; }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js" />
  <script>
    $(document).ready(function(){
      $('.outer').mouseover(function(){
        $('body').append('<h1>MOUSEOVER</h1>');
      }).mouseenter(function(){
        $('body').append('<h1>MOUSEENTER</h1>');
      });
    });
  </script>
</head>
<body>
  <div class="outer">
    <div class="inner" />
  </div>
</body>
</html>
```

그림 16-14 mouseover와 mouseenter 이벤트의 차이



❖ 키보드 이벤트

- keypress 이벤트는 한글로 사용할 수 없음

표 16-13 키보드 이벤트

이벤트 이름	설명
keydown	키보드가 눌려질 때 발생합니다..
keypress	글자가 입력될 때 발생합니다.
keyup	키보드가 떼어질 때 발생합니다.

■ 예시

그림 16-15 마투데이의 글쓰기

지금 내 생각을

마투데이에

150

😊

✓ 올리기



❖ 동적으로 글자 수 세기

- 키보드 이벤트와 키보드 이벤트의 주의할 점 숙지
- HTML 페이지 구성

코드 16-36 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script>
  </script>
</head>
<body>
  <div>
    <p>지금 내 생각을</p>
    <h1>150</h1>
    <textarea cols="70" rows="5">/textarea>
  </div>
</body>
</html>
```



❖ textarea 태그에 keyup 이벤트 연결

- keyup 이벤트 발생하면 글자 개수 받아 출력

코드 16-37 keyup 이벤트 연결

```
<script>
    $(document).ready(function (event) {
        $('textarea').keyup(function () {
            // 남은 글자 수를 구합니다.
            var inputLength = $(this).val().length;
            var remain = 150 - inputLength;

            // 문서 객체에 입력합니다.
            $('h1').html(remain);
        });
    });
</script>
```



❖ 키보드 이벤트의 실행 순서

- 1. 사용자가 키보드 누름
- 2. keydown 이벤트 발생
- 3. 글자 입력
- 4. keypress 이벤트 발생
- 5. 사용자가 키보드에서 손을 땀
- 6. keyup 이벤트 발생
- keydown 이벤트가 발생한 순간에는 글자가 입력돼 있지 않음
- 입력한 글자 수를 표시해야 하므로 keyup 이벤트 사용



❖ 글자 개수에 따른 스타일 변경

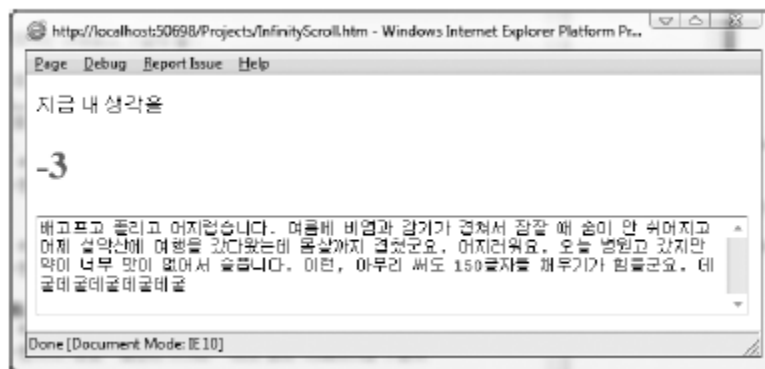
코드 16-38 글자 개수에 따른 스타일 변경

```
<script>
$(document).ready(function (event) {
    $('textarea').keyup(function () {
        // 남은 글자 수를 구합니다.
        var inputLength = $(this).val().length;
        var remain = 150 - inputLength;

        // 문서 객체에 입력합니다.
        $('h1').html(remain);

        // 문서 객체의 색상을 변경합니다.
        if (remain >= 0) {
            $('h1').css('color', 'Black');
        } else {
            $('h1').css('color', 'Red');
        }
    });
});
</script>
```

그림 16-16 실행 결과



❖ 윈도우 이벤트란?

- window 객체와 document 객체 이외에 img 태그 등이 사용할 수 있는 이벤트

표 16-14 윈도우 이벤트

이벤트	설명
ready	문서 객체가 준비를 완료합니다.
load	윈도우(문서 객체)를 불러들일 때 발생합니다.
unload	윈도우(문서 객체)를 닫을 때 발생합니다.
resize	윈도우의 크기를 변화시킬 때 발생합니다.
scroll	윈도우를 스크롤할 때 발생합니다.
error	에러가 있을 때 발생합니다.



❖ 윈도우 이벤트 사용한 무한 스크롤 구현

- 무한 스크롤은 최근 SNS와 같은 서비스에서 자주 볼 수 있는 효과
- Mixsh 사이트 참고 (<http://mixsh.com>)

그림 16-17 <http://mixsh.com>의 무한 스크롤



❖ jQuery 이벤트 사용해 간단하게 무한 스크롤 구현

■ 1) 기본적인 HTML 페이지 구성

코드 16-39 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script>
    $(document).ready(function () {

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

</body>
</html>
```



❖ jQuery 이벤트 사용해 간단하게 무한 스크롤 구현

- 2) window 객체에 scroll 이벤트 연결
 - window 객체는 별도의 선택자 사용하지 않음

코드 16-40 scroll 이벤트 연결

```
<script>
    $(document).ready(function () {
        $(window).scroll(function () {

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



❖ jQuery 이벤트 사용해 간단하게 무한 스크롤 구현

- window 객체에 scroll 이벤트 연결 원리
 - 사용자가 마우스 스크롤 움직이면 무조건 scroll 이벤트 발생
 - 무한 스크롤을 만들려면 화면 끝까지 스크롤이 도달했다는 사실 인식 필요
 - document 객체의 height 속성은 문서 전체의 높이 의미
 - 스크롤이 끝까지 내려가면

» window 객체 scrollTop + height = document 객체 높이

그림 16-18 무한 스크롤 이벤트의 구성



❖ jQuery 이벤트 사용해 간단하게 무한 스크롤 구현

- 스크롤이 끝까지 닿으면 자동으로 h1 태그 10개 추가

코드 16-41 무한 스크롤

```
<script>
$(document).ready(function () {
    $(window).scroll(function () {
        var scrollHeight = $(window).scrollTop() + $(window).height();
        var documentHeight = $(document).height();

        if (scrollHeight == documentHeight) {
            for (var i = 0; i < 10; i++) {
                $('<h1>Infinity Scroll</h1>').appendTo('body');
            }
        }
    });
});
</script>
```



❖ jQuery 이벤트 사용해 간단하게 무한 스크롤 구현

■ 전체 스크립트 코드

코드 16-42 전체 script 태그

```
<script>
  // 무한 스크롤 부분
  $(document).ready(function () {
    // 스크롤 이벤트 발생 시
    $(window).scroll(function () {
      // 필요한 변수를 구합니다.
      var scrollHeight = $(window).scrollTop() + $(window).height();
      var documentHeight = $(document).height();

      // 스크롤의 높이와 문서의 높이가 같을 때
      if (scrollHeight == documentHeight) {
        for (var i = 0; i < 10; i++) {
          $('<h1>Infinity Scroll</h1>').appendTo('body');
        }
      }
    });
  });

  // 테스트를 위해 내부에 공간을 채워둡니다.
  $(document).ready(function () {
    for (var i = 0; i < 20; i++) {
      $('<h1>Infinity Scroll</h1>').appendTo('body');
    }
  });
</script>
```



❖ 스크롤 구현 결과

- 두 번째 `$(document).ready()` 이벤트에서 h1 태그 20개 생성
 - 현재 문서에 아무 내용도 없으니 스크롤이 생기지 않기 때문
- 무한 스크롤을 바로 모바일에 적용하기 어려움
 - 모바일 디바이스에 따라 문서 객체가 60픽셀에서 200픽셀 정도 크게 나옴
 - 직접 장치를 확인하고 처리해야 함

그림 16-19 끝없이 계속되는 스크롤



❖ 입력 양식 이벤트

표 16-15 입력 양식 이벤트

이벤트 이름	설명
change	입력 양식의 내용을 변경할 때 발생합니다.
focus	입력 양식에 초점을 맞추면 발생합니다.
focusin	입력 양식에 초점이 맞춰지기 바로 전에 발생합니다.
focusout	입력 양식에 초점이 사라지기 바로 전에 발생합니다.
blur	입력 양식에 초점이 사라지면 발생합니다.
select	입력 양식을 선택할 때 발생합니다(input[type=text]태그와 textarea 태그 제외).
submit	submit 버튼을 누르면 발생합니다.
reset	reset 버튼을 누르면 발생합니다.



❖ submit 이벤트

- 입력 양식이 있어야 하므로 body 태그 구성

코드 16-43 body 태그 구성

```
<body>
  <form id="my_form">
    <table>
      <tr>
        <td>이름: </td>
        <td><input type="text" name="name" id="name"/></td>
      </tr>
      <tr>
        <td>비밀번호: </td>
        <td><input type="password" name="password" id="password"/></td>
      </tr>
    </table>
    <input type="submit" value="제출"/>
  </form>
</body>
```



❖ submit 이벤트

- form 태그에서 발생하는 이벤트
 - form 객체에 submit() 메서드 연결
 - 유효성 검사를 할 때는 기본 이벤트 제거해야 함

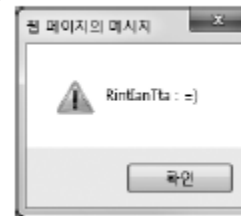
코드 16-44 submit 이벤트와 기본 이벤트 제거

```
<script>
    $(document).ready(function () {
        $('#my_form').submit(function (event) {
            // 입력 양식의 value를 가져옵니다.
            var name = $('#name').val();
            var password = $('#password').val();

            // 출력합니다.
            alert(name + ' : ' + password);

            // 기본 이벤트를 제거합니다.
            event.preventDefault();
        });
    });
</script>
```

그림 16-20 실행 결과



❖ change 이벤트

- type 속성이 checkbox와 radio인 input 태그의 상태 변경하는 이벤트
- click 이벤트가 아님

코드 16-45 check 속성 변경

```
<script>
$(document).ready(function () {
    $('#all_check').change(function () {
        if (this.checked) {
            $('#check_item').children().attr('checked', true);
        } else {
            $('#check_item').children().attr('checked', false);
        }
    });
});
</script>
<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>
```





Thank You !

모던 웹을 위한 Javascript jQuery 입문