



JQuery 2강- 탐색메서드, 이벤트, 효과

양 명 속

[now4ever7@gmail.com]



목차

- 탐색(Traversing) 메서드
 - 추가 필터링 관련 메서드
 - 찾기와 관계된 API들
- 이벤트
- 효과(Effects)



탐색(Traversing) 메서드

- 탐색(Traverse) 관련 메서드
 - Traverse – 어떤 것을 횡단하거나, 탐색하거나, 검토하거나, 여기 저기 왔다갔다 하는 등의 의미
 - 개체들의 집합에서 다시금 **특정 개체를 찾거나**, 필터하거나, 추가 하는등의 작업을 위해 지원되는 API 메서드들의 그룹
- 개체를 찾거나 필터하는 작업은 대부분 셀렉터를 통해서 이루어지지만, **1차적으로 셀렉터를 통해서 필터링을 하고 난 다음에, 그 결과 집합 내에서 추가적으로 필터링하거나, 추가 탐색하거나, 다른 결과 집합과 합치거나 하는 등의 작업을 하기 위해서는 트래버스 메서드가 유용하게 사용됨**

[1] 추가 필터링 관련 메서드

■ 추가 필터링 관련 메서드

- 셀렉터를 통해서 1차적으로 일치되는 집합을 얻어낸 후, 그 집합에 대해서 사용할 수 있는 메서드들

eq(index)	일치된 요소들 중에서 인덱스와 일치하는 단일 요소를 가져옴
filter(expr)	지정된 표현식과 매치되지 않는 모든 요소들을 제거함. 즉, 매치되는 요소들만을 가져옴
filter(func)	지정된 함수와 매치되지 않는 모든 요소들을 제거함
is(expr)	현재 개체가 표현식에 해당한다면 true, 표현식에 여러 개의 조건이 있다면 그 중 한 개만 맞아도 true 가 됨
not(expr)	지정된 표현식과 매치되는 모든 요소들을 제거함 즉, 매치되지 않는 요소들만을 가져옴
first()	첫 번째에 위치하는 문서 객체를 선택
last()	마지막에 위치하는 문서 객체를 선택
map(callback)	jQuery 개체 안에 있는 요소들의 집합을 다른 집합으로 변경해서 옮김

추가 필터링 관련 메서드

- filter() 메서드와 is(), not() 메서드
 - 셀렉터에 의해서 1차적으로 걸러낸 데이터를 가지고 작업을 하다가 2차적으로 추가 필터링을 한다거나, 특정 요소가 어떤 조건에 부합하는지를 검사한다거나 하는 작업에 활용.
- eq(index)
 - :eq(index) 라는 셀렉터 필터와 거의 동일
 - 하나는 메서드이고 하나는 셀렉터 구문이라는 차이

```
예)  
$("div:eq(1)").addClass("blue");  
$("div").eq(1).addClass("blue");
```

- 첫 번째 구문 - 현재 문서에 있는 div 중 2번째로 등장하는 div를 얻어내는 셀렉터, 그 요소에 대해 blue 라는 css 클래스를 적용
- 두 번째 구문 - 모든 div들을 선택한 다음, 그 중 2번째 개체를 얻어내기 위해서 eq(1)이라는 메서드를 사용
- 이 둘은 결과적으로 동일

end() 메서드

- eq() 메서드를 사용한 경우에는 end()라는 메서드의 도움을 받아 이전 상태로 돌아갈 수 있다는 장점이 있다
- end() 메서드
 - 현재 일치된 개체 집합을 변경하여, **방금 일어난 작업 직전의 상태로 되돌림**
- 대부분의 조작 메서드, 트래버스 메서드에 end()라는 메서드를 적용할 수 있음
 - 자기가 되돌릴 과거 상태가 존재한다면, 현재의 개체집합을 그 상태로 되돌리고, 되돌릴 과거가 없는 상태라면 단순히 빈 값을 반환
- 예) "모든 홀수 번째(index 1,3,5) div를 구한 다음, 그 중 첫 번째 div는 orange라는 배경색을 적용하고, 두 번째 div는 blue 배경색을 적용하라"
 - end() 메서드를 사용하지 않는다면

```
$("#div:odd").eq(0).css("background", "orange");  
$("#div:odd").eq(1).css("background", "blue");
```

- 각각에 대해서 동일한 검색을 두 번이나 반복하게 됨



end() 메서드

```
$("#div:odd")  
  .eq(0).css("background", "orange")  
  .eq(1).css("background", "blue");
```

- 이는 올바르게 동작하지 않음
- 이미 \$("#div:odd").eq(0)으로 인해서 매치되는 집합이 전체 div:odd가 아니라 단일 요소로 줄어든 상황에서는 eq(1)을 해봐야 아무런 대상도 찾을 수 없기 때문
- 최종 요소에 대해 작업을 하고 **방금 이전의 상태로 돌아가서 다시 검색할 수 있도록 end()** 사용

```
$("#div:odd")  
  .eq(0).css("background", "orange")  
  .end()    // $("#div:odd")와 동일  
  .eq(1).css("background", "blue");
```

- end()는 eq(0)을 하기 직전의 상태로 되돌리는 역할을 함
- end()를 사용하면 단 한 줄의 메서드 체인으로 원하는 작업을 마무리할 수 있게 됨

filter() 메서드

■ filter(*expression*)

- 인자인 expression – 여기에는 주로 셀렉터를 사용함
- 결과적으로는 일반 셀렉터를 통해 검색하는 것과 동일
- 셀렉터는 문서상에 존재하는 개체들을 대상으로 검색한다면, 이는 메모리 상에 존재하는 개체 집합을 대상으로 필터링을 한다는 부분에서 차이
- 즉, 셀렉터에 의한 1차적 검색 집합에 대해서 다시 추가적으로 필터링을 하고 싶은 경우에 사용하는 API 메서드
- 그 표현식과 매치되지 않는 모든 요소들을 개체 집합(메모리)에서 제거하여 필터링된 결과를 얻음

```
예)  
$("div:odd")  
$("div").filter(":odd")
```

■ filter(*func*)

- 인자로 표현식이 아닌 함수를 사용
- 필터링을 함수를 통해서 보다 세밀하게 제어하고 싶은 경우에 사용
- 필터링 조건이 일반적인 표현식으로 설명하기 어려운 경우에 별도의 함수로 작성을 해서 그 안에서 어떻게 필터를 수행할 것인지를 지정



filter() 메서드

```
$("#div").filter(":odd")  
  .eq(0).css("background", "orange")  
  .end()    // filter(":odd")  
  .eq(1).css("background", "blue")  
  .end()    // filter(":odd")  
  .css("color", "red"); //odd인 div들의 폰트색상을 red로 적용
```

- end() 메서드는 한번만 가능한 것이 아니라 메서드 체인 내에서 계속적으로 사용하여 방금 전 상태로 계속 올라갈 수 있음

```
<script>
  $(document).ready(function () {
    $('h3').filter(function (index) {
      return index % 3 == 0;
    }).css({
      backgroundColor: 'black',
      color: 'white'
    });
  });
</script>
</head>
<body>
  <h3>Header-0</h3>
  <h3>Header-1</h3>
  <h3>Header-2</h3>
  <h3>Header-3</h3>
  <h3>Header-4</h3>
  <h3>Header-5</h3>
</body>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5



not(expr) 메서드

- not(expr) 메서드 – filter(expr) 메서드와 반대되는 메서드
 - filter(expr) 메서드가 현재의 집합에서 expr 표현식과 매치되는 것만을 가져온다면, not(expr) 메서드는 현재의 집합에서 expr 표현식과 매치되지 않는 것들만을 가져옴

```
$("#div").filter(":odd")  
$("#div").not(":even")
```

- 결과는 동일

is(expr) 메서드

■ is(expr) 메서드

- 개체를 비교하여 true/false를 알려주는 메서드

예)

```
var $myDiv = $("div").eq(5);  
if ($myDiv.is("div"))  
{  
    $myDiv.css("border", "4px solid yellow");  
}
```

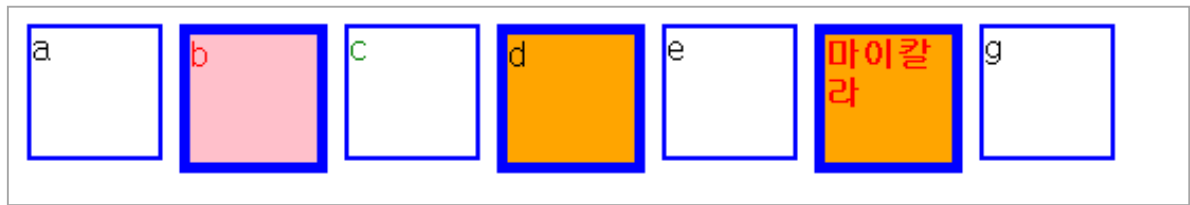
- div 중 6번째 요소를 가져와서 \$myDiv라는 변수로 참조
 - jQuery 개체를 일시적으로 참조하는 변수명에는 \$로 시작하는 변수명을 쓰도록 권유
 - 일반 변수들과 jQuery 개체변수가 쉽게 구분되기 때문
- \$myDiv 변수에 6번째 div를 참조한 뒤, is(expr) 메서드를 사용해서 그 개체가 "div" 개체인지를 비교하여, 만일 "div"가 맞다면, 그 div에 대해서는 노란색 테두리를 지정



is(expr) 메서드

- expr인 자에는 일반적으로 "div"처럼 단일 구문을 작성하곤 하지만, ","를 구분자로 사용하여 여러 개의 표현식을 나열할 수도 있음
- 예)
 - `$myDiv.is(".orange, .blue, .lightblue")`
 - `$myDiv`가 orange나 blue, lightblue css 클래스 중 하나라도 가지고 있으면 true

예제



```
<script type="text/javascript">
    $(document).ready(function() {
        $("div").eq(1).css("color", "red");

        $("div").filter(":odd")
            .eq(0).addClass("pink")
            .end() //filter(":odd")
            .eq(1).addClass("orange")
            .end() //filter(":odd")
            .css("border-width", "5px");

        $("div").not(":odd").eq(1).css("color", "green");

        var $myDiv = $("div").eq(5);
        if ($myDiv.is("div")) {
            $myDiv.css("color", "red")
                .css("font-weight", "bold");
        }

        if ($myDiv.is(".blue, .orange, .pink")) {
            $myDiv.text("마이칼라");
        }
    });
</script>
```



예제

```
<style type="text/css">
  div{width:60px;height:60px;float:left;
    border:2px solid blue;margin:4px}
  .orange{background:orange}
  .blue{background:blue}
  .pink{background:pink}
</style>
</head>

<body>
  <div>a</div>
  <div>b</div>
  <div>c</div>
  <div>d</div>
  <div>e</div>
  <div class="orange">f</div>
  <div>g</div>
</body>
```



[2] 찾기와 관계된 API들

find(expr)	자식을 대상으로 지정된 표현식과 일치하는 요소를 검색
add(expr)	현재 요소에 expr와 일치하는 요소를 추가함 즉, 일치되는 요소가 추가 확장되는 의미
next(expr)	현재 요소 바로 다음에 나오는 형제 요소 를 선택함 expr 을 지정하면 그것으로 필터링을 수행함
nextAll(expr)	현재 요소 바로 다음에 나오는 모든 형제 요소들 을 선택 expr 을 지정하면 그로 필터링을 수행
parent(expr)	현재 요소의 부모를 선택 expr 을 지정하면 그로 필터링을 수행
prev(expr)	현재 요소보다 앞서 나오는 형제 요소 를 선택 expr 을 지정하면 그로 필터링을 수행
prevAll(expr)	현재 요소보다 앞서 나오는 모든 형제 요소들을 선택 expr 을 지정하면 그로 필터링을 수행
siblings(expr)	현재 요소의 모든 형제 요소들(자신은 제외) 을 선택 expr 을 지정하면 그로 필터링을 수행

add(expr)

■ add(expr)

- 셀렉터에 의해 1차적으로 검색된 결과 집합에 추가적인 검색 집합을 합치는(add) 역할을 함

■ 예) `$("div").add("p").css("background", "yellow");`

- 검색 결과가 합쳐지는 것
- 즉, `$("div").add("p")` 라는 것은 `$("div, p")`와 같다고 볼 수 있음

```
$("div")  
  .css("border", "3px solid red")  
  .add("p")  
  .css("background", "yellow");
```

- 모든 **div**에게는 붉은색 테두리를 주고, **div**에 더하여 모든 **p** 요소들에게 노란 배경색을 주는 표현
- 즉, 모든 **div**와 **p**는 노란 배경색을 갖지만, 붉은 테두리는 **div**만 갖게 됨



find(*expr*)

■ find(expr)

- 현재 검색된 요소의 자식을 대상으로 추가적인 검색을 수행
- filter()와 find()의 차이
- filter()
 - 검색된 결과 집합에서 그 개체들에 대해서 다시 필터링을 하는 것
- find()
 - 검색된 결과 집합에서 찾는 것이 아니라, 결과 집합 내의 각 요소의 "자식"에게서 찾는 것

예)

```
$("#div:odd").filter("p")
```

```
$("#div:odd").find("p")
```

- 첫번째 구문 - filter 구문의 결과는 "없음"
 - 셀렉터에 의해서 홀수 div 들만 검색한 다음, 그 div들 중에서 p라는 태그를 갖는 것을 가져오려고 하면 없음
- filter() 메서드에서 인자로 요소명을 주는 경우는 거의 없음.
- find()에 의한 결과 - (div 요소의 하위로 p요소가 있다고 가정했을 경우) Div들 중에 p 태그를 갖는 것이 있다면, 그 p 요소들은 find() 명령에 의해서 검색됨

```
<style type="text/css">
```

```
div{width:60px;height:60px;float:left;
margin:7px;border:1px solid blue;
padding:7px}
```

```
</style>
```

```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
```

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

```
$(document).ready(function() {
    $("div").eq(1)
        .siblings().css("border-width", "3px") //자신을 제외한 형제요소들
        .end() // eq(1)
        .next().css("background", "lightBlue") //바로 다음 형제요소
        .end() //eq(1)
        .nextAll().css("color", "red"); //바로 다음의 모든형제요소들
```

```
//div의 자식에서 p를 찾아 테두리 주고, span도 검색결과에 추가하여 노랑배경색 주기
$("div").find("p").css("border", "1px solid red")
.add("span").css("background", "yellow");
```

```
//$("div p, span").css("background", "yellow");
```

```
});
```

```
</script>
```

```
<body>
```

```
<div>A<p>p a</p></div>
```

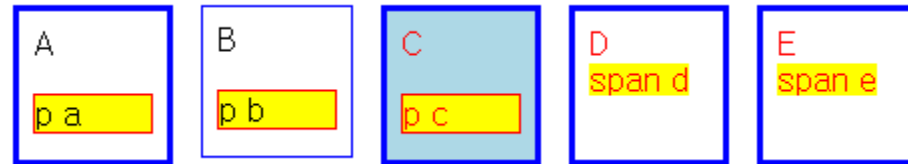
```
<div>B<p>p b</p></div>
```

```
<div>C<p>p c</p></div>
```

```
<div>D <br /><span>span d</span></div>
```

```
<div>E <br /><span>span e</span></div>
```

```
</body>
```



```

<script type="text/javascript">
$(document).ready(function() {
    $('.error').hide();
    $('.submit').click(function(event){
        var count=0;
        $('#signup').find('input[name=lang]').each(function(){
            if($(this).is(':checked')){
                count++;
            }
        });
        if(count==0){ //error
            $('.error').show();
            event.preventDefault(); //이벤트 진행을 막는다
        }else{
            $('.error').hide();
        }
    });
});
</script>
<style type="text/css">
    .error{ color:red;font-size:0.8em;}
</style></head><body>
<form id="signup" method="post" action="a.jsp">
<label><input type="checkbox" name="lang" value="1">java</label>
<label><input type="checkbox" name="lang" value="2">jsp</label>
<label><input type="checkbox" name="lang" value="3">oracle</label>
<span class="error">하나라도 체크하셔야 합니다</span>
<br><br>
<input class="submit" type="submit" value="전송">
</form>

```

☐ java
 ☐ jsp
 ☐ oracle
 하나라도 체크하셔야 합니다

전송

```
$(document).ready(function() {  
    $('.error').hide();  
    $('.submit').click(function(event){  
        var count=$('input[name=lang]:checked').length;  
        if(count==0){  
            $('.error').show();  
            event.preventDefault();  
        }else{  
            $('.error').hide();  
        }  
    });  
});
```

```

<script type="text/javascript">
$(document).ready(function() {
    $('.error').hide();
    $('.submit').click(function(event){
        var count=$('# #lang option:selected').val();
        if(count==0){
            $('.error').show();
            event.preventDefault();

        }else{
            $('.error').hide();
        }
    });
});
</script>
<style type="text/css">
    .error{ color:red;font-size:0.8em;}
</style></head>
<body>
<form id="signup" method="post" action="a.jsp">
<label for="lang">사용가능언어</label>
<select id="lang" class="infobox">
    <option value="0">선택하세요</option>
    <option value="1">java</option>
    <option value="2">jsp</option>
    <option value="3">c</option>
</select>

<span class="error">언어를 선택하셔야 합니다</span>
<br><br>
<input class="submit" type="submit" value="전송">
</form>

```

사용가능언어

선택하세요 ▼

언어를 선택하셔야 합니다

전송

```

<script type="text/javascript">
$(document).ready(function() {
    $('.error').hide();
    $('.infobox').blur(function(){
        var data=$(this).val();
        var len=data.length;
        if(len<1){
            $(this).parent().find('.error').show();
        }else{
            $(this).parent().find('.error').hide();
        }
    });

    $("#submit1").click(function(event){
        if($('#userid').val()==""){
            alert('아이디를 입력하세요');
            $('#userid').focus();
            event.preventDefault();
        }else if($('#pwd').val().length<1){
            alert('비밀번호를 입력하세요');
            $('#pwd').focus();
            event.preventDefault();
        }else if(!$('#name').val()){
            alert('이름을 입력하세요');
            $('#name').focus();
            event.preventDefault();
        }
    });
});
});

```

User Id *	<input type="text"/>	필수 입력항목입니다
Password *	<input type="password"/>	필수 입력항목입니다
Name *	<input type="text"/>	
<input type="button" value="전송"/>		

```

</script>
<style type="text/css">
    .error{ color:red;font-size:0.8em;}
</style>
</head>
<body>

<form id="signup" method="post" action="a.jsp">
<div>
    <label class="label" for="userid">User Id *</label>
    <input type="text" class="infobox" id="userid" />
    <span class="error">필수 입력항목입니다</span>
</div>
<div>
    <label class="label" for="pwd">Password *</label>
    <input type="password" class="infobox" id="pwd" />
    <span class="error">필수 입력항목입니다</span>
</div>
<div>
    <label class="label" for="name">Name *</label>
    <input type="text" class="infobox" id="name" />
    <span class="error">필수 입력항목입니다</span>
</div>
<input id="submit1" type="submit" value="전송">
</form>

```




이벤트 관련 메서드



Event 다루기

- 웹 프로그래밍에서의 이벤트
 - 사용자의 어떤 액션(개체를 클릭하거나, 마우스를 움직이는 행동)에 의해서 야기되는 사건을 의미
 - 이벤트 - 클릭, 마우스 오버, 키 입력 등
 - 이벤트가 발생하는 경우 특정 로직을 수행하기 위해서 해당 이벤트와 특정 메서드를 연결해 놓는데, 이러한 연결을 이벤트 처리기 매핑(event handler mapping)이라고 함
 - 매핑에 의해서 연결된 메서드들 - 이벤트 처리기 (event handler)
 - Click, MouseOver, KeyPress 등
- 자바스크립트를 사용해서 이벤트 처리기를 매핑할 수 있음
 - `Window.onload = pageLoad;`
 - 이벤트는 load
 - 이벤트 처리기는 pageLoad
 - 그 둘을 연결하는 위의 코드를 이벤트 처리기 매핑 코드라고 부름
 - 즉, load 이벤트가 발생하는 경우에는 pageLoad라는 이벤트 처리기를 실행하라는 의미



jQuery가 제공하는 이벤트 관련 메서드

on(type, fn)	매치된 요소에 이벤트 처리기를 바인딩한다. type에는 이벤트 명칭을, fn 에는 이벤트 처리기 함수를 작성하면 된다.
off(type, fn)	매치된 요소에서 지정된 이벤트와 매핑된 모든 처리기들을 제거한다. on 메서드와 상반되는 메서드이다
one(type, fn)	매치된 요소에 오직 한번만 실행되는 이벤트 처리기를 바인딩한다
trigger(event, data)	매치된 요소의 특정 이벤트를 트리거한다.



이벤트 연결

- 이벤트 연결 메서드
 - `on()` : 이벤트를 연결함

```
$(selector).on(이벤트이름, function(event){});  
$(selector).on(object);
```

- `$(selector).on (이벤트 이름, 이벤트 리스너);`
 - 이벤트 리스너 안에서 **this** 키워드는 이벤트가 발생한 객체를 의미함

```
<script type="text/javascript">
    $(document).ready(function() {
        $('button').on('click', function(event){
            $('#div1').html($(this).text()+" 클릭됨!!");

        });

        $('button').on({
            mouseover: function () { $(this).css('background', 'pink'); },
            mouseout: function () { $(this).css('background', ''); }
        });
    });
</script>
```

```
<body>
    <button>버튼1</button><br>
    <button>버튼2</button><br>
    <button>버튼3</button><br>

    <div id="div1"></div>
</body>
```

간단한 이벤트 연결

- 많이 사용하는 이벤트를 간단한 방식으로 연결할 수 있게 함

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

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

- hover() 메서드 - 두 개의 함수를 매개변수로 넣어줌
 - 첫 번째 매개변수는 mouseenter 이벤트 리스너, 두 번째 매개변수는 mouseleave 이벤트 리스너

hover()	mouseenter 이벤트와 mouseleave 이벤트를 동시에 연결함
---------	---

```
$(selector).hover(function(event){}, function(event){});
```



예제

```
$('#button').hover(function () {  
    $(this).css('background', 'skyblue');  
},  
function () {  
    $(this).css('background', '');  
});
```

```
$('#button').mouseover(function () {  
    $(this).css('background', 'cyan');  
}).mouseout(function () {  
    $(this).css('background', '');  
}); //메서드 체이닝 이용
```



이벤트 연결 제거

- 이벤트를 제거할 때는 off()메서드 사용

```
[1] $(selector).off();
```

```
[2] $(selector).off(eventName);
```

```
[3] $(selector).off(eventName, function);
```

- [1] 해당 문서 객체와 관련된 모든 이벤트를 제거
- [2] 해당 문서 객체의 특정 이벤트와 관련된 모든 이벤트를 제거
- [3] 특정 이벤트 리스너를 제거



예제 - 이벤트를 한번만 실행

```
<script>
    $(document).ready(function() {
        $('button').click(function() {
            $(this).html('클릭됨!');
            alert('클릭 이벤트 발생!');

            $(this).off();
        });
    });
</script>

<body>
    <button>버튼1</button><br>
    <button>버튼2</button><br>
    <button>버튼3</button><br>
</body>
```



이벤트 객체

- 자바스크립트의 이벤트 객체는 브라우저마다 사용방법이나 이벤트 객체의 속성이 다름
- JQuery의 이벤트 객체는 모든 브라우저가 같은 방법으로 사용하고, 같은 속성을 가짐
- JQuery 이벤트 객체의 속성

이벤트 객체 속성	설명
event.pageX	브라우저의 화면을 기준으로 한 마우스의 X좌표 위치
event.pageY	브라우저의 화면을 기준으로 한 마우스의 Y좌표 위치
event.preventDefault()	기본 이벤트를 제거함
event.stopPropagation()	이벤트 전달을 제거함

예제

```
<script>
    $(document).ready(function () {
        $('img').on('mousedown', function (event) {
            var x = event.pageX;
            var y = event.pageY;

            $("#div1").html("x="+x+", y="+y);
        });
    });
</script>
</head>
<body>
    <br><br>

    <div id="div1"></div>
</body>
```



x=12, y=10

기본 이벤트 제거/이벤트 전달 제거

`preventDefault()` - 기본 이벤트를 제거함

`stopPropagation()` - 이벤트 전달을 제거함

- `stopPropagation()`, `preventDefault()` 메서드를 함께 사용하는 경우가 많으므로,
- **jQuery**는 **return false** 를 사용하면 이 두 가지를 함께 적용한 것으로 인식함

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

자바 스크립트는 **return false**를 사용하면 기본 이벤트만 제거됨

```
<style>
```

```
div, a {  
    padding: 10px;  
    border: 1px solid gray;  
}
```

```
</style>
```

```
<script>
```

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

event.preventDefault()

- a 태그를 클릭하면 무조건 기본 이벤트가 실행되어 페이지가 이동하므로 a 태그의 기본 이벤트 제거

```
    $('div').click(function() {  
        $(this).css('background-color', 'skyblue');  
    });
```

```
});
```

```
</script>
```

- a 태그를 클릭하면 a 태그 이외에도 div 태그에 이벤트 전달이 일어남 (이벤트 버블링)
- 이벤트를 전달을 막으려면 **stopPropagation()** 메서드 사용

```
<body>
```

```
    <div>
```

```
        <a href="http://www.naver.com">naver</a>
```

```
    </div>
```

```
</body>
```



이벤트 버블링

■ 이벤트 버블링

- 겹쳐진 개체에서 가장 위쪽에 있는 개체를 선택했을 경우 아래의 개체들도 모두 이벤트가 일어나는 현상을 말한다.
- 하위 요소(자식)로부터 상위요소(부모)로 이벤트가 전파 되는 것
- 이벤트를 발생시킨 컴포넌트에서 이벤트를 부모 컴포넌트로 전달하는 것



이벤트의 bubbling : Button -> FDetail -> FoodLst => EComm

- Button을 클릭하면 아래의 모든 개체도 같이 이벤트가 일어나게 된다. 원하지 않는 이벤트까지 발생할 수 있는 우려가 있다.
- 버블링-자식 노드에서 부모 노드 순으로 이벤트를 실행하는 것
- 캡처링 - 이벤트가 부모 노드에서 자식 노드 순으로 실행되는 것

```

<style>
  div{width:200px; height:200px}
  .outer {background: skyblue;padding: 50px;}
  .inner {background: pink;}
</style>
<script type="text/javascript">
  function in_onclick() {
    alert("in_click");
    //이벤트 버블링 제거-다른 이벤트는 제거된다.
    event.cancelBubble = true; //자바스크립트 코드
  }
  function out_onclick() {
    alert("out_click");
  }
</script>
</head>

<body>
<h3>이벤트 버블링</h3>
<div class="outer" onclick="return out_onclick()">
  <div class="inner" onclick="return in_onclick()">
    </div>
  </div>
</body>

```

버블링이 일어나는 것의 **event.cancelBubble** 속성값을 **true**로 해주면 버블링이 일어나지 않는다

```

$(document).ready(function () {
  $('.outer').click(function () {
    alert("outer_click");
  });
  $('.inner').click(function (event) {
    alert("inner_click");
    //이벤트 버블링 제거
    event.stopPropagation();
  });
});

```



마우스 이벤트

이벤트 이름	설명
click	마우스를 클릭할 때 발생합니다.
dblclick	마우스를 더블 클릭할 때 발생합니다.
mousedown	마우스 버튼을 누를 때 발생합니다.
mouseup	마우스 버튼을 뗄 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.


```

<style>
  div{width:200px; height:200px}
  .outer {
    background: skyblue;
    padding: 50px;
  }
  .inner {
    background: pink;
  }

```

```

</style>

```

```

<script>

```

```

$(document).ready(function () {
  $('div.outer').mouseover(function () {
    $('body').append('<h3>mouseover</h3>');
  }).mouseenter(function () {
    $('body').append('<h3>MOUSEENTER</h3>');
  });
});

```

```

</script>

```

```

<body>
  <div class="outer">
    <div class="inner"></div>
  </div>
</body>

```



- **mouseover** 이벤트는 이벤트 버블링을 적용함
- 내부의 **div** 태그 안에 들어가도 이벤트를 발생시킴
- **mouseenter** 이벤트는 문서 객체의 안에 있는지 외부에 있는지 체크

키보드 이벤트/윈도우 이벤트

이벤트 이름	설명
keydown	키보드를 누를 때 발생합니다.
keypress	글자가 입력될 때 발생합니다.
keyup	키보드를 뗄 때 발생합니다.

한글은 keypress 이벤트를 지원하지 않음

이벤트	설명
ready	문서 객체가 준비 완료되면
load	윈도(문서 객체)를 불러들일 때 발생합니다.
unload	윈도(문서 객체)를 닫을 때 발생합니다.
resize	윈도의 크기를 변화시킬 때 발생합니다.
scroll	윈도를 스크롤할 때 발생합니다.
error	에러가 있을 때 발생합니다.

```

<script type="text/javascript">
    $(document).ready(function() {
        //[1] lblError 레이어 클리어
        $('#txtPwd').keyup(function() {
            $('#lblError').text(''); // 클리어
        });

        //[2] 암호 확인 기능 구현
        $('#txtPwdConfirm').keyup(function() {
            if ($('#txtPwd').val() != $('#txtPwdConfirm').val()) {
                $('#lblError').html("<b>비밀번호가 틀립니다.</b>"); // 레이어에 HTML 출력
            }
            else {
                $('#lblError').html("<b>비밀번호가 맞습니다.</b>");
            }
        });
    });
</script>

```

```

<body>
    비밀번호:<input type="password" id="txtPwd" size="20"><br />
    비밀번호 확인:<input type="password" id="txtPwdConfirm" size="20"><br />
    <input type="button" id="btnOK" value="확인" />
    <div id="lblError" style="color:red">암호를 입력하십시오.</div>
</body>

```



입력 양식 이벤트

이벤트 이름	설명
change	입력 양식의 내용을 변경할 때 발생합니다.
focus	입력 양식에 초점을 맞추면 발생합니다.
focusin	입력 양식에 초점이 맞추어지기 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"] 태그 및 textarea 태그 제외).
submit	submit 버튼을 누르면 발생합니다.
reset	reset 버튼을 누르면 발생합니다.

```
<style type="text/css">
```

```
.label {float: left; width: 100px; }
```

```
.error { color: red; padding-left: 10px; }
```

```
.submit { margin-left: 125px; margin-top: 10px;}
```

```
</style>
```

```
<script>
```

```
$(document).ready(function() {
```

```
    $(' .error').hide();
```

```
    $('#frm1').submit(function(event) {
```

```
        //$('# .submit').click(function(event) {
```

```
            var data = $('#userid').val();
```

```
            var len = data.length;
```

```
            if (len < 1) {
```

```
                $(' .error').show();
```

```
                event.preventDefault(); //기본 이벤트 제거
```

```
            } else {
```

```
                $(' .error').hide();
```

```
            }
```

```
        });
```

```
    });
```

```
</script></head><body>
```

```
<form id="frm1" method="post" action="event5.jsp">
```

```
    <div> <label class="label" for="userid">아이디 *</label>
```

```
        <input type="text" name="userid" id="userid"/>
```

```
        <span class="error">반드시 값을 입력하셔야 합니다!</span> </div>
```

```
    <input class="submit" type="submit" value="전송">
```

```
</form></body>
```

아이디 *

반드시 값을 입력하셔야 합니다!

전송

```

<script>
    $(document).ready(function() {
        $('#chkAll').change(function() {
            if (this.checked) {
                $('#chkItem').children().prop('checked', true);
            } else {
                $('#chkItem input[type=checkbox]').prop('checked', false);
            }
        });
    });
</script>
<body>
    <input type="checkbox" id="chkAll" /><label for="chkAll">All</label>

    <div id="chkItem">
        <input type="checkbox" id="chk1" /><label for="chk1">참외</label>
        <input type="checkbox" id="chk2" /><label for="chk2">자두</label>
        <input type="checkbox" id="chk3" /><label for="chk3">수박</label>
    </div>
</body>

```

☐ All
☐ 참외 ☐ 자두 ☐ 수박

체크 상태를 바꿀 때는 **prop()** 메서드 사용

```

$( document ).ready( function() {
    $( '#chkAll' ).click( function() {
        $('#chkItem').children().prop( 'checked', this.checked );
    } );
} );

```



.prop()

- .prop() – 지정한 선택자를 가진 첫 번째 요소의 속성값을 가져오거나 속성값을 추가함
 - 주의할 점 – HTML 입장에서의 속성(attribute)이 아닌 JavaScript 입장에서의 속성(property)이라는 것
- 문법
 - 속성값을 가져올 때
 - .prop(propertyName)
 - 속성값을 추가할 때
 - .prop(propertyName, value)



.prop()

- prop는 dom객체에서 속성(Property)을 Object 리터럴로 접근함
 - 즉, 자바스크립트 객체 접근방식으로 접근함

```
예)  
o.value = 'text';  
jQuery.prop('value', 'text');
```

- attr는 dom객체에서 특성(Attribute)을 마크업 특성으로 접근함
 - 즉, html tag의 특성을 참조하거나 적용함

```
예)  
  
jQuery.attr('src', 'imgUrl');
```

- attr은 기본 태그에서 설정할수 있는 요소들을 접근할 때 용이하고,
prop는 dom객체로 활성화된 객체의 열거된 속성을 접근할 때 용이함

.prop()

- jquery가 1.6 버전부터 .attr() 메서드를 .attr() 과 .prop() 로 세분화함
 - HTML 속성을 취급하는 attributes 와 JavaScript 프로퍼티를 취급하는 properties를 분리
- 비교

```
.attr( "checked" ) - checked  
.prop( "checked" ) - true  
.is( ":checked" ) - true
```

- 1.6 이전에 attr("checked") 의 값은 프로퍼티값인 true 를 반환
- 1.6 이후에 prop("checked") 가 true 반환, attr("checked")는 문자열 값인 "checked" 를 반환함



.prop()

at <http://api.jquery.com/prop/>

<code>elem.checked</code>	true (Boolean) Will change with checkbox state
<code>\$(elem).prop("checked")</code>	true (Boolean) Will change with checkbox state
<code>elem.getAttribute("checked")</code>	"checked" (String) Initial state of the checkbox; does not change
<code>\$(elem).attr("checked") (1.6)</code>	"checked" (String) Initial state of the checkbox; does not change
<code>\$(elem).attr("checked") (1.6.1+)</code>	"checked" (String) Will change with checkbox state
<code>\$(elem).attr("checked") (pre-1.6)</code>	true (Boolean) Changed with checkbox state



이벤트 관련 메서드

- one() 메서드

- 일회성 이벤트 처리를 지원하기 위한 메서드

```
예) $("#MyBtn").one("click", function() {  
    alert("버튼 클릭");  
});
```



이벤트 관련 메서드

- trigger() 메서드

- 코드를 사용해서 특정 이벤트를 일으키기 위한 메서드
- 즉, 실제로 사용자가 클릭 액션을 하지 않았음에도, 클릭을 한 것처럼 이벤트를 일으키는 코드

예)

```
$("#MyBtn").trigger("click");
```

- 사용자가 버튼을 클릭한 것처럼 클릭 이벤트를 일으키게 함
- 코드를 사용해서 강제로 이벤트를 발생시키는 방법

예제

```
<script type="text/javascript">
    $(document).ready(function() {
        $('#btn').on("click", function() {
            alert('버튼 클릭됨');
        });
        // 페이지 로드시 사용자의 반응이 아닌 코드에 의해서 click 이벤트를 실행
        $('#btn').trigger("click");

        $('#img1').one('click', function() {
            alert('이미지는 한번만 클릭하세요');
        });
    });
</script>
</head><body>
<div id="my">
    <input type="button" id="btn" value="버튼" >
    <img id="img1" src='images/dog.jpg'>
</div>
</body>
```



예제

Hello JQuery

안녕

```
<script type="text/javascript">
    $(document).ready(function() {
        $("p").hover(function() {
            $(this).css("background", "pink");
        }, function() {
            $(this).css("background", "");
        });
    });
</script>

<body>
    <p>Hello JQuery</p>
    <p>안녕</p>
</body>
```

```

<script type="text/javascript">
    $(document).ready(function() {
        $("#btn1").click(function() {
            $('#div1').toggleClass("hidden");
        });
        $('table tr:gt(0)').hover( // 0번째 인덱스보다 큰 경우만
            function() { $(this).addClass('hover'); },
            function() { $(this).removeClass('hover'); }
        );
    });
</script>
<style type="text/css">
    .hidden { display:none; }
    #div1{background-color:pink; width:200px; height:70px;padding:7px}
    .hover { cursor:pointer; background-color:lightBlue; }
</style>
</head>
<body>
    <div id="div1">메뉴 레이어</div>
    <input id="btn1" type="button" value="버튼" /> <br /><br />
    <table border="1">
        <tr> <th>번호</th> <th>프로그래밍</th> </tr>
        <tr> <td>1</td><td>C#</td></tr>
        <tr> <td>2</td><td>ASP.NET</td></tr>
        <tr> <td>3</td><td>JQuery</td> </tr>
    </table></body></html>

```

`$('#div1').toggle();`



```

<script type="text/javascript">
$(document).ready(function() {
    $('.error').hide();
    $('#submit').click(function(event){
        var data=$('#infobox').val(); //12a3
        var len=data.length;
        var c;
        for(var i=0;i<len;i++) {
            //c=data.charAt(i).charCodeAt(0); //1,2,a
            c=data.charCodeAt(i);
            if(c <48 || c >57) {
                $('.error').show();
                event.preventDefault();
                break;
            } else {
                $('.error').hide();
            }
        }
    });
});
</script></head>

```

```

<style type="text/css">
.error{
    color:red;
    font-size:0.8em;
}
</style>

```

```

//음수 허용 (- 기호 허용)
for(var i=0;i<len;i++) {
    c=data.charAt(i).charCodeAt(0);
    if(c==45 && i==0){
        continue;
    }
}
}

```

```

</script></head>
<body>
    <form id="signup" method="post" action="">
        <div>
            <label class="label" for="age">나이 </label>
            <input type="text" class="infobox" name="age" id="age"/>
            <span class="error"> 숫자만 입력하셔야 합니다!</span>
        </div>
        <input class="submit" type="submit" value="Submit">
    </form>
</body>

```



```

<script type="text/javascript">
$(document).ready(function() {
    $('.error').hide();
    $('#submit').click(function(event){
        var data=$('#infobox').val();
        if(validate_phoneno(data)){
            $('.error').hide();
        }else{
            $('.error').show();
            event.preventDefault();
        }
    });
});

function validate_phoneno(ph){
    var pattern=new RegExp(/^([0-9-+]+)$/g);
    return pattern.test(ph); //정규식과 일치하면 true
}
</script>
<style type="text/css">
    .error{
        color:red;
        font-size:0.8em;
    }
</style>
</head>
<body>
<form id="signup" method="post" action="a.jsp">
<div>
<label class="label" for="phone">전화번호</label>
<input type="text" class="infobox" name="phone" id="phone"/>

```

전화번호

전화번호는 숫자,+,-만을 포함할 수 있습니다.

- 정규식 `/^([0-9-+]+)$/g`
0 에서 9사이의 숫자나 - 또는 + 기호로 시작하거나 끝나야 한다는 의미
(^는 시작, \$는 끝을 의미)
닫기 대괄호(`]`) 뒤의 + 기호는 이 패턴이 한 번 또는 그 이상 반복된다는 의미
즉, 숫자 또는 - 또는 + 기호를 전화번호 입력란에 여러 번 입력할 수 있음

```

<span class="error">전화번호는 숫자,+,-만을 포함할 수 있습니다.</span>
</div>
<input class="submit" type="submit" value="전송 ">
</form>

```

```

<script type="text/javascript">
$(document).ready(function() {
    $('.error').hide();
    $('#submit').click(function(event){
        var data=$('#infobox').val();
        if(validate_userid(data)){
            $('.error').hide();
        }else{
            $('.error').show();
            event.preventDefault();
        }
    });
});

function validate_userid(uid){
    var pattern= new RegExp(/^[a-z0-9_]+$ /g);
    return pattern.test(uid);
}
</script>

<style type="text/css">
    .error{        color:red;font-size:0.8em;}
</style>
</head>
<body>
<form id="signup" method="post" action="a.jsp">
<div>
<label class="label" for="userid">아이디</label>
<input type="text"  class="infobox" name="userid" id="userid"/>
<span class="error">아이디는 문자,숫자,(밑줄문자)만을 포함할 수 있습니다.</span>
</div>
<input class="submit" type="submit" value="전송 ">
</form>

```

아이디

ab12_*@

전송

아이디는 문자,숫자,(밑줄문자)만을 포함할 수 있습니다.

- 정규식 /^[a-z0-9_]+\$ /g

a에서 z 사이의 문자, 0 에서 9사이의 숫자나 _로 시작하
거나 끝나야 한다는 의미



정규표현식

- 정규표현식을 사용하는 방법
 - 객체초기화(Object initializer) 방법
 - `var regExp = /정규표현식/[Flag];`
- Flag의 종류
 - 자주 사용하는 Flag
 - [1] g - Global - 문자열 내의 모든 패턴을 찾는다
 - [2] i - Ignore case - 문자열의 대소문자를 구별하지 않는다
 - [3] m - Multi line - 문자열이 행이 바뀌어도 찾는다.
- Flag를 사용을 하지 않을 수도 있다.
 - 만약 Flag를 설정 하지 않을 경우에는 문자열 내에서 검색대상이 많더라도 한번만 찾고 끝나게 된다

```

<script type="text/javascript">
    $(function(){
        $(' .error').hide();
        //$('#userid').focus();
        $('#submit').click(function(event){
            var count=0;
            /* $('#signup').find('input[name=lang]').each(function(idx, item){
                if($(this).is(':checked')){
                    count++;
                }
            }); */
            var data= $('#signup input[name=lang]:checked');
            count = data.length;
            if(count==0){ //error
                $('#signup input[name=lang]').parent()
                .parent().find(' .error').show();
                event.preventDefault(); //기본이벤트의 진행을 막는다
            }else{ //정상
                $('#signup input[name=lang]').parent()
                .parent().find(' .error').hide();
            }
            var lang2=$('#lang2 option:selected').val();
            if(lang2==0){ //error
                $('#lang2').parent().find(' .error').show();
                event.preventDefault();
            }else{ //정상
                $('#lang2').parent().find(' .error').hide();
            }
        }
    });

```

☐ java
 ☐ jsp
 ☐ oracle
 하나라도 체크하셔야 합니다

사용가능언어

 언어를 선택하셔야 합니다

User Id * 필수 입력항목입니다

Password * 필수 입력항목입니다

Name * 필수 입력항목입니다

//console.log(\$('.infobox').val()); //첫번째 입력상자의 value만 가져온다

```
$('.infobox').each(function(idx, item){  
    var data = $(this).val();  
    if(data.length<1){//error  
        $(this).parent().find('.error').show();  
        event.preventDefault();  
    }else{ //정상  
        $(this).parent().find('.error').hide();  
    }  
});
```

});

});

```
$('.infobox').blur(function(event){  
    var data = $(this).val();  
    if(data.length<1){//error  
        $(this).parent().find('.error').show();  
        event.preventDefault();  
    }else{ //정상  
        $(this).parent().find('.error').hide();  
    }  
});
```

});

});

</script>

<style type="text/css">

.error{ color:red;font-size:0.8em;}

</style>

</head><body>

<form id="signup" method="post" action="a.jsp">

```
<div>
    <label><input type="checkbox" name="lang" value="1">java</label>
    <label><input type="checkbox" name="lang" value="2">jsp</label>
    <label><input type="checkbox" name="lang" value="3">oracle</label>
    <span class="error">하나라도 체크하셔야 합니다</span>
</div>

<div>
    <label for="lang2">사용가능언어</label>
    <select id="lang2">
        <option value="0">선택하세요</option>
        <option value="1">java</option>
        <option value="2">jsp</option>
        <option value="3">c</option>
    </select>
    <span class="error">언어를 선택하셔야 합니다</span>
</div>

<div>
    <label class="label" for="userid">User Id *</label>
    <input type="text" class="infobox" id="userid" />
    <span class="error">필수 입력항목입니다</span>
</div>

<div>
    <label class="label" for="pwd">Password *</label>
    <input type="password" class="infobox" id="pwd" />
    <span class="error">필수 입력항목입니다</span>
</div>

<div>
    <label class="label" for="name">Name *</label>
    <input type="text" class="infobox" id="name" />
    <span class="error">필수 입력항목입니다</span>
</div>

<input id="submit" type="submit" value="전송">
</form>
```

```

<script type="text/javascript">
    $(function(){
        $('#submit').click(function(event){
            if($('#userid').val().length<1){
                alert('아이디를 입력하세요');
                $('#userid').focus();
                return false;
            }else if($('#pwd').val().length<1){
                alert('비밀번호를 입력하세요');
                $('#pwd').focus();
                return false;
            }else if($('#name').val().length<1){
                alert('이름을 입력하세요');
                $('#name').focus();
                return false;
            }
        });
    });
</script>
<style type="text/css">
    .error{ color:red;font-size:0.8em;}
</style>
</head>

```

User Id *

Password *

Name *

자바스크립트 알림

아이디를 입력하세요

```
<body>
<form id="signup" method="post" action="a.jsp">
<div>
    <label class="label" for="userid">User Id *</label>
    <input type="text" class="infobox" id="userid" />
</div>
<div>
    <label class="label" for="pwd">Password *</label>
    <input type="password" class="infobox" id="pwd" />
</div>
<div>
    <label class="label" for="name">Name *</label>
    <input type="text" class="infobox" id="name" />
</div>

<input id="submit" type="submit" value="전송">
</form>
</body>
```




실습

```
<!DOCTYPE html>
<html><head><meta charset="EUC-KR">
<title>Insert title here</title>
<style type="text/css">
    .error{ color:red;font-size:0.8em;}
</style></head>
<body>
<form id="signup" method="post" action="a.jsp">
<div>    <label><input type="checkbox" name="lang" value="1">java</label>
        <label><input type="checkbox" name="lang" value="2">jsp</label>
        <label><input type="checkbox" name="lang" value="3">oracle</label>
        <span class="error">하나라도 체크하셔야 합니다</span>
</div>
<div>    <label for="lang2">사용가능언어</label>
        <select id="lang2">
            <option value="0">선택하세요</option>
            <option value="1">java</option>
            <option value="2">jsp</option>
            <option value="3">c</option>
        </select>
        <span class="error">언어를 선택하셔야 합니다</span>
</div>
```



실습

```
<div>    <label class="label" for="userid">User Id *</label>
          <input type="text" class="infobox" id="userid" />
          <span class="error">필수 입력항목입니다</span>
</div>
<div>    <label class="label" for="pwd">Password *</label>
          <input type="password" class="infobox" id="pwd" />
          <span class="error">필수 입력항목입니다</span>
</div>
<div>    <label class="label" for="name">Name *</label>
          <input type="text" class="infobox" id="name" />
          <span class="error">필수 입력항목입니다</span>
</div>
<input id="submit" type="submit" value="전송">
</form>

</body>
</html>
```



실습2

```
<!DOCTYPE html><html><head><meta charset="EUC-KR">
<title>Insert title here</title>
<style type="text/css">
    .lbError{ color:red; }
</style></head>
<body>
<form method="post" action="a.jsp">
<div>
    <input type="checkbox" id="chkAll" /><label for="chkAll">All</label>
    <div id="chkItem">
        <input type="checkbox" id="chk1"/><label for="chk1">참외</label>
        <input type="checkbox" id="chk2"/><label for="chk2">자두</label>
        <input type="checkbox" id="chk3"/><label for="chk3">수박</label>
    </div>
</div><hr>
<div>    <label class="label" for="userid">아이디</label>
        <input type="text" class="infobox" name="userid" id="userid"/>
        <span class="lbError">아이디는 문자,숫자,*,@만을 포함할 수 있습니다.</span>
</div>
```



실습2

```
<div>
```

```
    비밀번호:<input type="password" id="txPwd" size="20"><br />
```

```
    비밀번호 확인:<input type="password" id="txPwdConfirm" size="20">
```

```
    <span class="lbError">암호를 입력하시오.</span></div>
```

```
    <input type="button" id="btOK" value="확인" />
```

```
</form>
```

```
<hr>
```

```
    <table border="1">
```

```
        <tr> <th>프로그래밍 1</th><th>프로그래밍 2</th> </tr>
```

```
        <tr> <td>C#</td><td>C</td></tr>
```

```
        <tr> <td>ASP.NET</td><td>C++</td></tr>
```

```
        <tr> <td>jQuery</td><td>java</td> </tr>
```

```
    </table></
```

```
</body>
```

```
</html>
```



효과(Effects) 관련 메서드

jQuery가 제공하는 Effects 관련 메서드

■ 기본 시각 효과

show()	요소 보이기
hide()	요소 숨기기
toggle()	show()와 hide() 메서드를 번갈아 실행
fadeIn()	서서히 나타나기(불투명도 표시) 문서 객체를 선명하게 보여줌
fadeOut()	문서객체를 흐리게 사라지게 함
fadeToggle()	fadeIn(),fadeOut() 메서드를 번갈아 실행
slideDown()	위에서 아래로 슬라이드 다운됨(보여짐)
slideUp()	아래서 위로 슬라이드 업됨(사라짐)
slideToggle()	슬라이드 업/다운에 대한 토글링

```
$(selector).method()  
$(selector).method(speed)  
$(selector).method(speed, callback)
```



Effects 관련 메서드

```
$(selector).method()  
$(selector).method(speed)  
$(selector).method(speed, callback)
```

■ 각 매개변수의 의미

■ speed

- 효과를 진행할 속도를 지정
- 밀리 초 단위의 숫자 또는 문자열 slow, normal, fast 를 입력

■ callback

- 효과를 모두 완료하고 실행할 함수

```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
```

```
<script>
    $(document).ready(function () {
        $('button').first().click(function () {
            $('.list1').toggle();
        });
        $('button').eq(1).click(function () {
            $('.list2').slideToggle(2000);
        });
        $('button').last().click(function () {
            $('.list3').fadeToggle('slow');
        });
    });
</script></head><body>
```

```
<button>show Toggle</button>
```

```
<button>slide Toggle</button>
```

```
<button>fade Toggle</button>
```

```
<div class="list1">
```

```
    <h1>제목1</h1>
```

```
    <p>내용1입니다</p>
```

```
</div><br>
```

```
<div class="list2">
```

```
    <h1>제목2</h1>
```

```
    <p>내용2입니다</p>
```

```
</div><br>
```

```
<div class="list3">
```

```
    <h1>제목3</h1>
```

```
    <p>내용3입니다</p>
```

```
</div></body>
```





show()

- show()
 - 대상이 되는 모든 엘리먼트를 보이도록 함
- show(*speed*[, *fn*])
 - speed – speed 수치 동안 보이도록 함
 - fast (200), normal (400), slow (600), 또는 0 이상의 숫자 (millisecond 단위)
 - fn – speed 의 수치가 끝났을 때, callback fn을 실행함

```
$("#img:first").show(3000);  
  
$("#img:eq(1)").show(4000, function() {  
    alert("show animation complete");  
});
```



hide()

- hide()
 - 대상이 되는 모든 엘리먼트를 보이지 않도록 함
- hide(*speed* [, *fn*])

예제



```
<script type="text/javascript">
    $(document).ready(function(e) {
        /*$("#div")
            .css({ width: 120, height: 32, border: "1px solid gray" })
            .css('text-align','center');*/

        $("#img:first").show(3000);

        $("#btn1").click(function() {
            $("#img:eq(1)").show(2000, function() {
                alert("show animation complete");
            });
        });

        $("#btn2").click(function() {
            $("#img:eq(1)").hide(2000, function() {
                alert("hide event occurred!");
            });
        });
    });
</script>
```

```
<style>
div{
    width: 120;
    height: 32;
    border: 1px solid gray;
    text-align:center;
}
</style>
```



예제

```
</head>
<body>
  <div>
    
  </div>
  <br />
  <input id="btn1" type="button" value="보이기" />
  <input id="btn2" type="button" value="숨기기" /><br /><br />
  
</body>
```



toggle()

- toggle()

- 대상 엘리먼트가 화면에 보이는 경우 보이지 않게 만들고, 보이지 않는 경우 보여지도록 변경

```
$("#img").toggle();
```

예제



```
<script type="text/javascript">
    $(document).ready(function() {
        $("button").click(function() {
            $("img").toggle();
        });
    });
</script>
```

```
<body>
    <button>Toggle</button>
    <p></p>
    <p></p>
</body>
```



slideDown(*speed* [, *fn*])

- slideDown(*speed* [, *fn*])
 - 대상 엘리먼트의 style 속성값이 "display:none" 인 경우 상단에서부터 하단으로 speed 수치의 시간안에 서서히 보여지도록 함

```
$("#div1 img").slideDown(2000);
```



`slideUp(speed [, fn])`

- `slideUp(speed [, fn])`
 - 대상 엘리먼트의 `style` 속성값이 `"display:block"` 인 경우 상단방향으로 `speed` 수치의 시간안에 서서히 사라지게 함

```
$("#div1 img").slideUp(2000);
```




slideToggle(*speed* [, *fn*])

- slideToggle(*speed* [, *fn*])
 - 대상 엘리먼트의 style 속성값이 "display:none" 인 경우 slideDown을 실행시키고, "display:block" 일 경우 slideUp을 실행시킴

```
$("#img").slideToggle(2000);
```

```

<script type="text/javascript">
$(document).ready(function() {
    var bool=true;
    $('.message').hide();
    $('span.readmore').click(function() {
        $('.message').slideToggle('slow');
        if(bool){
            $(this).text('Read Less...');
            //bool=false;
        }else{
            $(this).text('Read More...');
            //bool=true;
        }
        bool=!bool;
    });
}); </script></head>

```

Styles make the formatting job much easier and efficient.
Read More...

Styles make the formatting job much easier and efficient.
Read Less...

To give an attractive look to web sites, styles are heavily used. JQuery is a powerful JavaScript library that allows us to add dynamic elements to our web sites. Not only it is easy to learn but easy to implement too. A person must have a good knowledge of HTML and CSS and a bit of Javascript. JQuery is an open source project that provides a wide range of features with cross platform compatibility.

```

<body>
<div>Styles make the formatting job much easier and efficient.</div>
<span class="readmore">Read More...</span>
<div class="message">

```

To give an attractive look to web sites, styles are heavily used.

JQuery is a powerful JavaScript library that allows us to add dynamic elements to our web sites. Not only it is easy to learn but easy to implement too. A person must have a good knowledge of HTML and CSS and a bit of Javascript. JQuery is an open source project that provides a wide range of features with cross platform compatibility.

```

</div>
</body>

```



`fadeIn(speed [, fn])`

- `fadeIn(speed [, fn])`
 - 대상 엘리먼트의 `style` 속성값이 `"display:none"` 인 경우 불투명도를 점차 높여서 보이도록 함

```
$("#div1 img").fadeIn(3000);
```



fadeOut(*speed* [, *fn*])

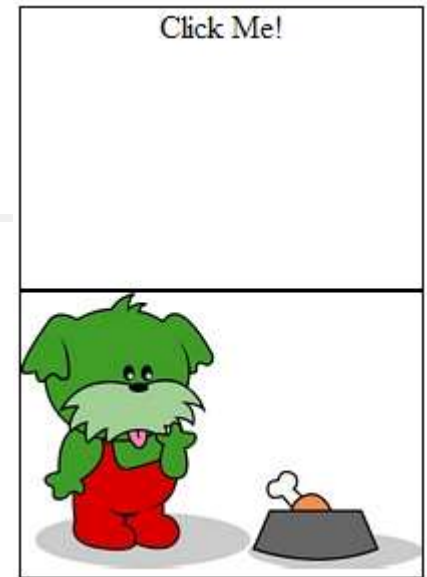
- fadeOut(*speed* [, *fn*])
 - 대상 엘리먼트의 style 속성값이 "display:block" 인 경우 불투명도를 점차 낮춰서 보이지 않도록 함

```
$("#div2 img").fadeOut(3000, function() {  
    $("#div2 span").show();  
});
```

예제

```
<script type="text/javascript">
    $(document).ready(function(e) {
        $("#div1 img").css("display","none");
        $("#div1").click(function() {
            $("#div1 span").hide();
            $("#div1 img").fadeIn(3000);
        });

        $("#div2 span").css("display", "none");
        $("#div2").click(function() {
            $("#div2 img").fadeOut(3000, function() {
                $("#div2 span").show();
            });
        });
    });
</script></head>
<body>
    <div id="div1">
        <span>Click Me!</span>
        
    </div>
    <div id="div2">
        <span>Click Me!</span>
        
    </div></body>
```



```
<style>
div{
    width: 200px;
    height: 140px;
    border: 1px solid gray;
    text-align:center;
}
</style>
```



`fadeTo(speed , opacity [, fn])`

- `fadeTo(speed [, fn])`
 - 대상 엘리먼트의 불투명도를 정해진 시간 동안 변화도록 설정함
 - 불투명도(`opacity`) : 0 ~ 1
 - 불투명도(`opacity`)가 0 : 보이지 않게 됨

```
$("#img").fadeTo("slow", 0.2);
```

예제



```
<script type="text/javascript">
    $(document).ready(function(e) {
        $("img").mouseenter(function() {
            $("img").fadeTo("slow", 0.2);
        });

        $("img").mouseleave(function() {
            $("img").fadeTo("slow", 1);
        });
    });
</script>

<body>
    <div>
        
    </div>
</body>
```

```
<style>
div{
    width: 200px;
    height: 140px;
    border: 1px solid gray;
    text-align:center;
}
</style>
```

jQuery가 제공하는 Effects 관련 메서드

■ 사용자 정의 효과

animate()

사용자 지정 효과를 생성
여러 개의 애니메이션 효과를 동시 부여

```
$(selector).animate(object)  
$(selector).animate(object, speed)  
$(selector).animate(object, speed, callback)
```

■ 첫 번째 매개변수인 객체에 입력할 수 있는 속성

- opacity
- width, height
- margin, padding
- left, right, top, bottom

현재 상태에서 지정한 속성의
형태로 애니메이션이 동작함


```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
```

```
<style>
```

```
div {
```

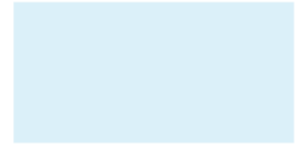
```
width: 70px; height: 70px;
```

```
background-color: skyblue;
```

```
position: relative;
```

```
}
```

```
</style>
```



```
<script>
```

```
$(document).ready(function () {
```

```
  $('div').hover(function () {
```

```
    $(this).animate({
```

```
      left:300, width: '140px', opacity:0.3
```

```
    }, 'slow');
```

```
  }, function () {
```

```
    $(this).animate({
```

```
      left:0, width: '70px', opacity:1
```

```
    }, 'slow');
```

```
  });
```

```
});
```

```
</script>
```

```
<body>
```

```
  <div></div>
```

```
</body>
```

```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
```

```
<style>
```

```
div {
```

```
width: 50px; height: 50px;
```

```
background-color: pink;
```

```
}
```

```
</style>
```

```
<script>
```

```
$(document).ready(function () {
```

```
  $('div').click(function () {
```

```
    $(this).animate({
```

```
      width: '+=50',
```

```
      height: '+=50'
```

```
    }, 2000);
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
  <div></div>
```

```
</body>
```



현재상태를 기준으로 한 상대적 애니메이션
+=, -= 연산자 : 상대적으로 속성이 변함

사각형을 클릭하면 계속 사각형이 커짐

```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
```

```
<style>
```

```
div {
```

```
width: 50px; height: 50px;
```

```
background-color: green;
```

```
position: relative;
```

```
}
```

```
</style>
```

```
<script>
```

```
$(document).ready(function () {
```

```
  $('div').click(function () {
```

```
    $(this).animate({
```

```
      left: '+=100'
```

```
    }, 2000).animate({
```

```
      width: '+=50'
```

```
    }, 2000).animate({
```

```
      height: '+=50'
```

```
    }, 2000);
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
  <div></div>
```

```
</body>
```

jQuery의 효과 메서드를 사용하면 명령이 차례로 큐에 들어가고, 들어간 순서대로 실행됨

animate() 메서드를 메서드 체이닝으로 연결
- 동시에 애니메이션이 실행되는 것이 아니라, **차례대로 실행됨**



animate(param[, duration] [,fn])

- animate(param[, duration] [, fn])
 - 대상 엘리먼트에 animate를 적용하게 되면 엘리먼트의 크기와 위치, 색상 등을 원하는 시간 동안 변경할 수 있으며, 변경이 완료된 순간 event를 callback fn 의 형태로 확인할 수 있음

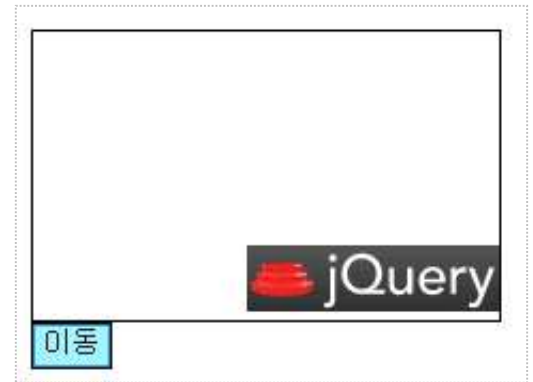
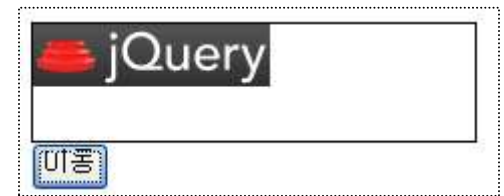
```
$(this).animate({ width: "100%",opacity: 0.4, marginLeft: "20"}  
                , 2000);
```

예제 1

```
<script type="text/javascript">
    $(document).ready(function(e) {
        /*$("#div").css({
            width: 220, height: 58,
            border: "1px solid black"
        }).attr("align", "left");*/

        $("#img").click(function() {
            $(this).animate({
                width: "100%",
                opacity: 0.4,
                marginLeft: "20"
            }, 2000);
        });

        $("#btn1").click(function() {
            $("#img").animate({ marginLeft: 100 }, 1000)
                .animate({ marginTop: 100 }, 1000);
        });
    });
</script>
</head>
<body>
    <div>
        
    </div>
    <input id="btn1" type="button" value="이동" /></body>
```



```
<style>
div{
    width: 220px;
    height: 58px;
    border: 1px solid gray;
    text-align:left;
}
</style>
```

예제2

```
<script type="text/javascript">
    $(document).ready(function() {
        /*$("#div#outer").css({width:301,height:20,border:"1px solid black"});
        $("#div#gaege").css({width:0,height:20,"backgroundColor":"gray"});*/

        $("#div#outer").hover(
            function(){
                $("#div#gaege").animate({
                    width:"100%"
                }, 3000);
            },
            function(){
                $("#div#gaege").animate({
                    width:"0%"
                }, 3000);
            }
        );
    });
</script>
</head>
<body>
    <div id="outer">
        <div id="gaege"></div>
    </div>
</body>
```

```
<style>
div#outer{
    width: 301px;
    height: 20px;
    border: 1px solid black;
}
div#gaege{
    width: 0;
    height: 20px;
    background:gray;
}
</style>
```

jQuery가 제공하는 Effects 관련 메서드

■ 애니메이션 정지

stop()

현재 진행중인 애니메이션 효과를 멈춤

```
$(selector).stop()  
$(selector).stop(clearQueue)  
$(selector).stop(clearQueue, goToEnd)
```

■ 매개변수

- clearQueue, goToEnd – boolean, 입력하지 않으면 false
- clearQueue – true 로 설정하면 clearQueue() 를 실행하는 것과 같은 효과를 냄
- goToEnd – true 로 설정하면 제자리에서 멈추는 것이 아니라, 지정한 최종 형태에서 멈춤

clearQueue() – 큐의 내용을 제거함



jQuery가 제공하는 Effects 관련 메서드

- 애니메이션 큐 관리 메서드
 - `clearQueue()` - 큐의 내용을 제거함
- 애니메이션 지연
 - `delay()` - 큐에 있는 명령을 잠시 중지함
 - `delay()` 메서드의 매개변수에 정지하고자 하는 시간을 밀리초 단위로 입력함
 - 예) 3초 후 `animate()` 메서드를 실행
 - `$(div).delay(3000).animate({width: 100}, 'slow');`
 - 효과 관련 메서드를 특정한 시간 후에 실행하고 싶을 때 사용



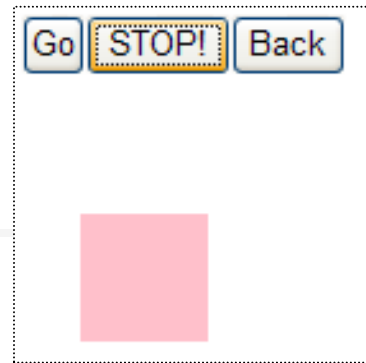
stop()

- stop()

- 대상 엘리먼트에 적용되어 있는 animate 를 중지시킴

```
$(".block").stop();
```

예제



```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
<style type="text/css">
    div { position: absolute; background-color: pink; left: 0px; top: 60px;
        width: 50px; height: 50px; margin: 5px; }
</style>
<script type="text/javascript">
    $(document).ready(function () {
        // Start animation
        $("#go").click(function () {
            $(".block").animate({ left: '+=100px', top: '+=100px' }, 2000);
        });
        // Stop animation when button is clicked
        $("#stop").click(function () {
            $(".block").stop();
        });
        // Start animation in the opposite direction
        $("#back").click(function () {
            $(".block").animate({ left: '-=100px', top: '-=100px' }, 2000);
        });
    });
</script>
</head>
<body>
    <button id="go">Go</button>
    <button id="stop">STOP!</button>
    <button id="back">Back</button>
    <div class="block"></div></body>
```

실습

- more 를 클릭하면 본문 div 태그가 보이도록
- close 버튼을 클릭하면 본문은 사라지고, more는 보이도록
 - slideUp, slideDown, fadeIn, fadeOut

안녕하세요~~~ more...

안녕하세요~~~
여기는 본문입니다..

close



UI 관련 플러그인



jQuery UI 플러그인

■ 플러그인

- 폼 유효성 검사, 슬라이더 등의 추가 기능을 별도의 자바스크립트 파일로 제공하는 jQuery 라이브러리
- [1] 코어 플러그인 - 코어 프로젝트의 일부로 간주되는 jQuery의 확장 부분
 - jQuery UI 를 코어 플러그인이라고 할 수 있다
- [2] 커뮤니티 플러그인
 - 일반 개발자들이 자신이 직접 개발한 것을 공개한 플러그인
 - 대부분의 jQuery 플러그인은 <http://plugins.jquery.com/> 에서 확인 가능

■ jQuery UI 플러그인 - jQuery에서 공식적으로 지정한 플러그인

- 2008년 초까지 많이 사용한 유명한 jQuery 플러그인을 모아 정리한 것
- UI와 관련된 플러그인



jQuery UI 다운로드

- <http://jqueryui.com/>
 - 최신버전의 jQuery UI 관련 플러그인 다운로드
 - 유용한 UI 관련 컨트롤 제공
- 종류
 - 상호작용(Interaction)
 - Draggable
 - Droppable
 - Selectable
 - Sortable
 - 위젯(Widgets)
 - Accordion
 - Autocomplete
 - Datepicker
 - Progressbar
 - Slider
 - Tabs 등
 - 유틸리티(Utility)
 - 효과(Effects)



jQuery UI 플러그인

- jQuery UI의 모든 플러그인은 **플러그인의 이름과 같은 메서드가 해당 플러그인의 핵심 메서드임**
 - Datepicker 플러그인은 datepicker() 메서드가 핵심 메서드, Accordion 플러그인은 accordion() 메서드가 핵심 메서드
 - \$("#startDay").datepicker();
- 모든 jQuery UI 플러그인은 **핵심 메서드의 첫 번째 매개변수에 옵션 객체를 넣어줌**
 - 각 플러그인마다 해당 옵션이 있고, 옵션은 객체로 만들어 첫 번째 매개변수로 넣어줌

```
$("#startDay").datepicker({  
    dateFormat:'yy-mm-dd',  
    changeYear:true,  
    dayNamesMin:['일','월','화','수','목','금','토'],  
    monthNames:['1월','2월','3월','4월','5월','6월','7월','8월','9월','10월','11월','12월']  
});
```



jQuery UI 플러그인의 메서드

- 속성(옵션)을 지정할 때는 옵션 객체를 사용
- 달력을 생성한 이후에 선택한 날짜를 읽어오거나 하는 행위는 메서드를 사용함
- jQuery UI 플러그인의 모든 메서드의 형태

```
$(selector).datepicker( "getDate" )
```

플러그인 이름 메서드 이름

- datepicker 플러그인의 getDate 메서드라고 부름



예제

```
<link rel="stylesheet" href="../../jquery/jquery-ui.css" type="text/css" />
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
<script src="../../jquery/jquery-ui.js" type="text/javascript"></script>
<script>
$(document).ready(function() {
    $("#startDay").datepicker({
        dateFormat:'yy-mm-dd',
        changeYear:true,
        dayNamesMin:['일','월','화','수','목','금','토'],
        monthNames:['1월','2월','3월','4월','5월','6월','7월','8월','9월','10월','11월','12월']
    });
    $('button').click(function () {
        var currentDate = $("#startDay").datepicker( "getDate" );
        alert(currentDate);
    });
});
</script></head><body>
<input type="text" name="startDay" id="startDay">
<br>
<button>버튼</button></body>
```

예제-Accordian

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="euc-kr">
  <link rel="stylesheet" href="../../jquery/jquery-ui.css">
  <script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
  <script src="../../jquery/jquery-ui.js" type="text/javascript"></script>
  <script type="text/javascript">
    $(document).ready(function () {
      $('#accordion').accordion();
      //$('#accordion').accordion({ event: "mouseenter" });
    });
  </script>
</head>
<body>
  <div id="accordion">
    <h3>제목1</h3>
    <div><p>내용1</p></div>
    <h3>제목2</h3>
    <div><p>내용2</p></div>
    <h3>제목3</h3>
    <div><p>내용3</p></div>
  </div>
</body>
</html>
```

제목1
내용1
제목2
제목3

예제-tabs

탭 제목1

탭 제목2

탭 제목3

첫번째 영역의 내용

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="../../jquery/jquery-ui.css">
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
<script src="../../jquery/jquery-ui.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $('#tabs').tabs(); // 탭
    });
</script>
</head>
<body>
    <div id="tabs">
        <ul>
            <li><a id="tab1" href="#tabContent_1">탭 제목1</a></li>
            <li><a id="tab2" href="#tabContent_2">탭 제목2</a></li>
            <li><a id="tab3" href="#tabContent_3">탭 제목3</a></li>
        </ul>
        <div id="tabContent_1"><p>첫번째 영역의 내용</p></div>
        <div id="tabContent_2"><p>두번째 영역의 내용</p></div>
        <div id="tabContent_3"><p>세번째 영역의 내용</p></div>
    </div>
</body>
</html>
```

예제-autocomplete

```
<!doctype html><html lang="en"><head> <meta charset="utf-8">
<title>jQuery UI Autocomplete – Default functionality</title>
<link rel="stylesheet" href="../../jquery/jquery-ui.css">
<script src="../../jquery/jquery-1.11.3.js"></script>
<script src="../../jquery/jquery-ui.js"></script>
<script>
$(function() {
  var availableTags = [
    "ActionScript",  "AppleScript",  "Asp",    "BASIC",    "C",    "C++",    "Clojure",    "COBOL",
    "ColdFusion",    "Erlang",    "Fortran", "Groovy",    "Haskell",    "Java",    "JavaScript",
    "Lisp",    "Perl",    "PHP",    "Python",    "Ruby",    "Scala",    "Scheme"
  ];
  $( "#tags" ).autocomplete({
    source: availableTags
  });
});
</script>
</head>
<body>
<div class="ui-widget">
  <label for="tags">Tags: </label>
  <input id="tags">
</div></body></html>
```

Tags: a

- ActionScript
- AppleScript
- Asp
- BASIC
- Erlang
- Fortran
- Haskell
- Java
- JavaScript
- Scala



trim

<code>\$.trim(string 문자열)</code>	입력 값으로 받은 문자열의 앞, 뒤 공백을 제거함
----------------------------------	-----------------------------



예제

```
<script src="../../jquery/jquery-1.11.3.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function(){
        var str = " hello, jQuery!! ";
        var result = "before[" + str + "<br>";
        result+="길이:" + str.length + "<br><br>";
        result+="after[" + $.trim(str) + "<br>"; // 양쪽 공백 제거
        result+="trim 후 길이:" + $.trim(str).length;

        $('#result').html(result);
    });
</script>
<body>
<div id="result"></div>
</body>
```

before[hello, jQuery!!]
길이:17

after[hello, jQuery!!]
trim 후 길이:15