

ver 1.6.1



<http://www.jobtc.kr>
훈련교사 : 박원기

1 jQuery 개요

1.1 jQuery 의 특징

- CSS 셀렉터
html 의 구조를 명료하고 읽기 쉬운 형태로 사용 및 표현 가능
- 플러그인 아키텍처
중복되는 기능을 최대한 피하고 이미 개발된 수많은 플러그인들을 사용할 수 있음
- 메서드 체인
여러 개의 동작을 하나로 묶어 사용할 수 있어 중복되는 코드를 최대한 줄일 수 있음.
- 크로스 브라우저
브라우저별로 처리 방법이 다른 이벤트 처리나 함수들의 문장을 하나의 표현으로 사용 가능.

1.2 문장 구조

- 따옴표는 작은따옴표, 큰따옴표 를 구분하지 않는다.
- 요소명 > 요소명 과 같이 특정 지역을 한정하여 Selector 를 사용하는 것이 속도 향상에 도움이 된다.

1.3 jQuery 메서드

jQuery() 메서드는 jQuery 라이브러리의 근간을 이루는 메서드이다. 이는 \$() 이란 별칭으로 사용되고 있다.

- \$(selector[, context])
- \$(element)
- \$(elementArray)
- \$(object)
- \$(html)

와 같이 여러 형태로 사용되며 특정 DOM 엘리먼트에 해당하는 새로운 jQuery 객체를 생성한다.

1.4 작성 방법

방법 1,2,3 은 모두 같은 의미이다.

1.4.1 방법 1

- jQuery('div').hide();
- jQuery('div').text('hi...');

1.4.2 방법 2

- jQuery('div').hide().text('hi...');

1.4.3 방법 3

- jQuery('div')
.hide()
.text('hi...');

1 jQuery 개요 > 1.5 접근 방법

1.5 접근 방법

- HTML DOM 을 통한 접근
- DOM 계층구조를 통한 접근
- 속성(id 또는 class, 기타)을 통한 접근

1.6 라이브러리 설치

라이브러리 사용 방법은 두가지로 나뉘 수 있다. 라이브러리를 복사하여 서버에 직접 설치해서 사용하는 방법과 원격지의 라이브러리를 가져다 사용하는 방법이 있다. 각각의 장단점이 있기 때문에 사용 환경에 적합한 방법을 사용하도록 하자.

예를 들어 네트워크 상태가 불안정하거나 오프라인 상태에서도 jQuery 를 사용해야 한다면 라이브러리를 복사하여 사용하는 방법이 좋을 것이다.

1.6.1 라이브러리 복사

www.jquery.com 에 접속하여 라이브러리를 적당한 경로에 복사함으로써 설치작업 완료.



1.6.2 CDN 사용 방법

CDN 은 Content Delivery Network 또는 Content Distribution Network 의 약자로 '컨텐츠 전송 네트워크' 라는 명칭으로 불려진다.

Step 1.

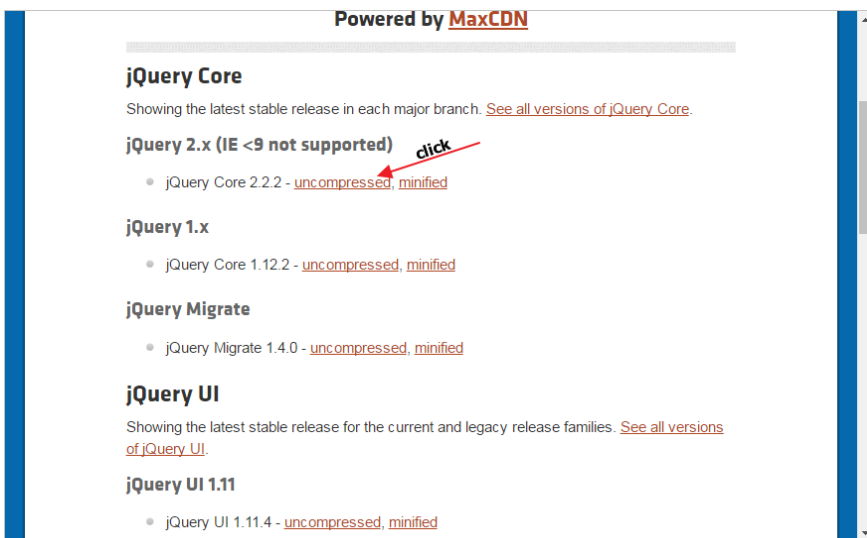
<https://code.jquery.com> 사이트를 방문한다.

1 jQuery 개요 > 1.6 라이브러리 설치



Step 2.

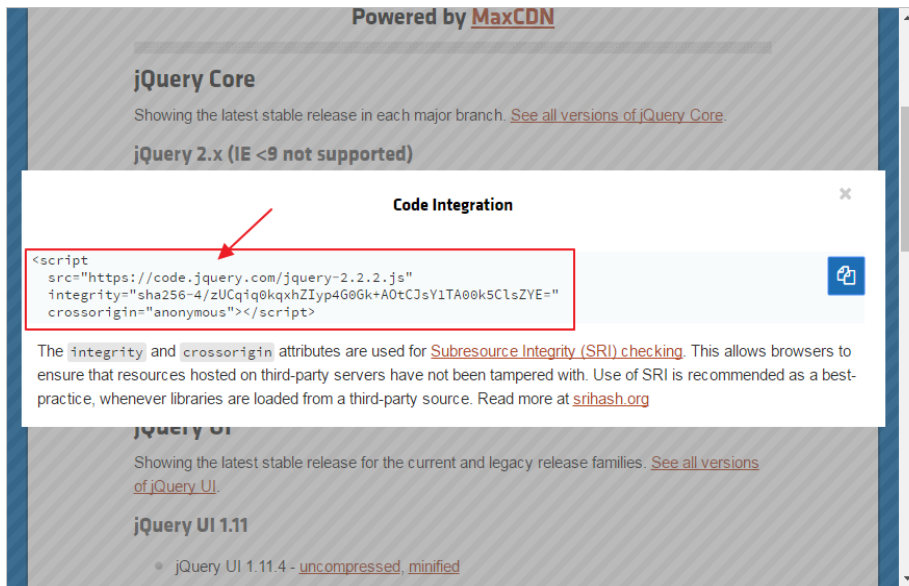
원하는 라이브러리의 버전의 링크를 클릭한다.



Step 3.

표시된 <script/>영역을 복사하여 필요한 웹 페이지에 붙여넣기 한다.

1 jQuery 개요 > 1.6 라이브러리 설치



내용이 모두 필요한 것은 아니다. 대부분 아래와 같은 URL 만을 갖고도 충분히 사용할 수 있다.

```
<script src="https://code.jquery.com/jquery-2.2.2.js"></script>
```

단, CDN 방법은 네트워크가 항상 연결되어 있는 상태에서만 사용할 수 있기 때문에 **네트워크 환경이 원활하지 않다면 라이브러리를 다운로드 받아 사용하는 것을 권장한다.**

1.7 jQuery API 구조

- jQuery CORE
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

1.8 DOM ?

DOM(Document Object Model) 의 약자로 HTML 의 구조를 트리 구조로 조직하여 클라이언트 영역에서 재조합 기능을 제공하고 사용자와 상호 작용을 통해 트리 구조 내에서 개별 객체에 접근이 가능하도록 만든 구조.

2 Selector

2.1 Selector 란

jQuery 를 사용하여 뭔가를 하려면 대상이 필요하다. 그 대상을 찾는 기능을 Selector 라 한다.
Selector 의 종류

- CSS Selector
- Attribute Selector
- Form Selector

2.2 사용 방법

- `$(selector) | jQuery(selector)`
- `jQuery()`보다 표현상 간단한 `$()`를 더 많이 사용.
- selector : 태그 또는 id, class 명을 뜻함.

2.3 CSS 셀렉터의 종류

2.3.1 `$(this)`

- 자기 자신을 나타냄.

2.3.2 `$("*")`

- 모든 엘리먼트를 선택한다.
- `$("*")` : 모든 엘리먼트 선택
- `$(p>*)` : p 엘리먼트안의 모든 자식 엘리먼트를 선택한다.

2.3.3 `$("#id")`

- `$('#center_div')` : id 가 center_div 인 엘리먼트를 선택한다.

2.3.4 `$(".class name")`

- `$('.subject')` : class 가 subject 인 모든 엘리먼트를 선택한다.

2.3.5 `$("element name")`

- `$('div')` : 문서안의 모든 div 엘리먼트를 선택한다.
- `$('font')` : 문서안의 모든 font 엘리먼트를 선택한다.

2.3.6 `$("element#id")`

- `$(p#body)` : p 엘리먼트중에 아이디가 body 인 엘리먼트를 선택한다.

2.3.7 `$("element.class")`

- `$(p.body)` : p 엘리먼트중에 클래스명이 body 인 엘리먼트를 선택한다.

2.3.8 `$("s1, s2, s3, ...")`

- `$(div, p, span)` : 엘리먼트 종류중 div, p, span 인 모든 엘리먼트를 선택한다.

2.3.9 `$("#container element")`

- `$('#center div')` : 아이디가 center 인 엘리먼트안에 있는 자식 엘리먼트 중 div 인 엘리먼트를

선택한다.

2.3.10 \$('<container element>')

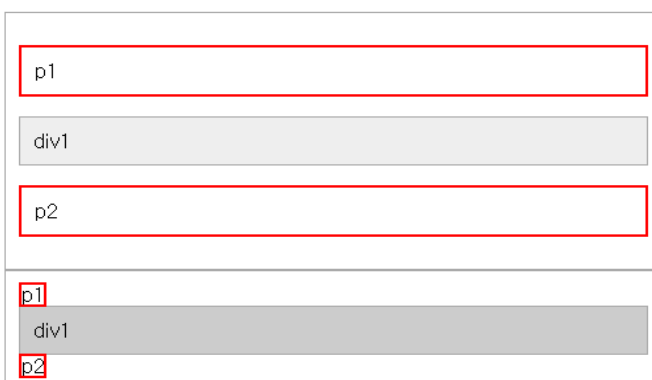
- \$('<center div>'): 클래스명이 center 인 엘리먼트안에 있는 자식 엘리먼트 중 div 인 엘리먼트를 선택한다.

[기본예]

/selector/ef_1.html

```
<!DOCTYPE html >
<html lang='ko'>
<head>
<meta charset="UTF-8">
<title>E F child selector</title>
<style>
  div,p{border:1px solid #aaaaaa;padding:10px}
</style>
<script src='../lib/jquery-2.0.1.js'></script>
<script>
  $(document).ready(function(){
    $('#center div').css('background-color', '#eeeeee');
    $('.center div').css('background-color', '#cccccc');
    $('p, span').css('border', '2px solid #ff0000');
  });
</script>

</head>
<body>
<div id='center'>
  <p>p1</p>
  <div>div1</div>
  <p>p2</p>
</div>
<div class='center'>
  <span>p1</span>
  <div>div1</div>
  <span>p2</span>
</div>
</body>
</html>
```



2.4 HTML DOM 을 이용한 접근

\$("#태그명") 처럼 사용하여 HTML 요소를 직접 접근한다.

[기본예]

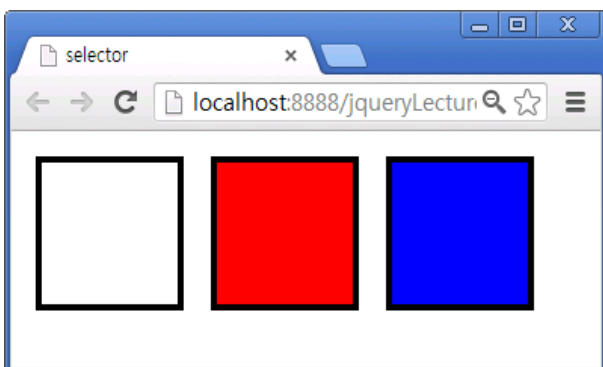
- jQuery 라이브러리는 /lib/폴더에 설치되어 있다고 가정함.
- 파일의 유형은 html 타입이여도 상관 없음.

selector.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>selector</title>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        $("div").css("border", "5px solid #000");
        $("div").css("width", "100px");
        $("div").css("height", "100px");
        $("div").css("margin", "10px");
        $("div").css("float", "left");

        $("#selector_div1").css("background-color", "#ff0000");
        $(".selector_class1").css("background-color", "#0000ff");
    }
)

</script>
</head>
<body>
    <div></div>
    <div id='selector_div1'></div>
    <div class='selector_class1'></div>
</body>
</html>
```



2.5 속성값을 사용한 Selector

- 요소안에 정의된 특정 속성값을 사용하여 요소를 선택할 수 있다.
- Selector 가 생략되면 모든 엘리먼트가 선택 대상이 된다.

2.5.1 \$(Selector[attr])

- attr 속성을 갖는 엘리먼트를 선택한다.
- \$('[rel]') : rel 속성을 갖는 모든 엘리먼트를 선택한다.
- \$('[myclass.style]') : 클래스명이 myclass 이고 style 속성이 있는 모든 엘리먼트를 선택한다.

2.5.2 \$(Selector[attr='value'])

- attr 의 값이 value 인 엘리먼트(요소)를 선택한다.
- \$('[name=myname]') : name 속성의 값이 myname 인 모든 엘리먼트를 선택한다.

2.5.3 \$(Selector[attr != 'value'])

- attr 속성이 없거나 attr 의 값이 value 가 아닌 요소를 선택한다.
- \$('a[rel !=jobtc]') : a 엘리먼트중에 rel 속성이 없거나 rel 값이 jobtc 가 아닌 모든 a 엘리먼트를 선택한다.
- \$('input[name != 'myname']') : input 엘리먼트중에 name 속성이 없거나 name 값이 myname 이 아닌 모든 input 엘리먼트를 선택한다.

2.5.4 \$(Selector[attr ^='value'])

- attr 의 값이 value 로 시작하는 요소
- \$('id^='hello') : id 속성값이 hello 로 시작하는 모든 엘리먼트를 선택한다.

2.5.5 \$(Selector[attr \$='value'])

- attr 의 값이 value 로 끝나는 요소
- \$('input[name \$='kr']') : input 엘리먼트의 name 속성값이 kr 로 끝나는 모든 엘리먼트를 선택한다.

2.5.6 \$(Selector[attr *='value'])

- attr 의 값에 value 를 포함하는 요소
- \$('[style *='background']') : style 속성에 background 속성값이 포함된 모든 엘리먼트를 선택한다.
- \$('a[href *='.com']') : a 엘리먼트들에서 href 속성 값에 .com 값이 포함된 모든 엘리먼트를 선택한다.

2.5.7 \$(Selector[attr ~= 'value'])

- attr 의 값에 공백과 value 를 포함하는 요소. 검색될 value 값은 공백으로 분리된 단어이어야 된다.
- \$('a[rel ~= 'jobtc']') : a 엘리먼트들 중에 rel 속성값이 jobtc 단어가 포함된 모든 엘리먼트를 선택한다.

2.5.8 \$(Selector[attr |= 'value'])

- attr 의 값이 value 이거나 value- 로 시작되는 요소
- 이 셀렉터는 다국어 지원을 위해 CSS 규격에 도입됨.
- \$('div[id |= 'bar']') : div 엘리먼트들 중에 id 속성이 bar 이거나 bar- 로 시작하는 엘리먼트를 선택한다.

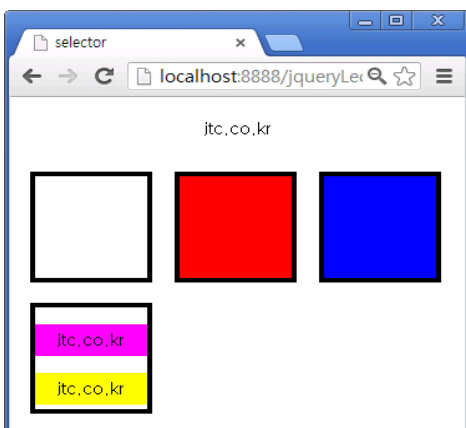
[기본예]

/selector/selector2.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
```

2 Selector > 2.5 속성값을 사용한 Selector

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>selector</title>
<style>
div{
    border:5px solid #000;
    width:100px;height:100px;
    margin:10px;
    float:left;
}
p{
    padding:7px;
    text-align:center;
}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        $("div[a='abd']").css("background-color","#ff0000");
        $("div[a$='c']").css("background-color","#0000ff");
        $("div > p").css("background-color","#ff00ff");
        $("div > p[s='abc']").css("background-color","#ffff00");
    }
)
</script>
</head>
<body>
    <p s="abc">jtc.co.kr</p>
    <div></div>
    <div id='selector_div1' a='abd'></div>
    <div class='selector_class1' a='bc'></div>
    <div>
        <p> jtc.co.kr</p>
        <p s="abc">jtc.co.kr</p>
    </div>
</body>
</html>
```



2.6 DOM 구조를 이용한 Selector

2.6.1 \$('parent > child')

- 부모 요소 바로 아래 자식인(1 단계까지만) 요소만을 반환함.

2.6.2 \$('ancestor descendant')

- 조상/후손의 관계. 단계와 상관없이 모든 후손 요소를 반환함.

2.6.3 \$('prev + next')

- prev 다음에 next 요소가 오는 동일한 패턴의 항목을 반환함.

2.6.4 \$('prev~siblings')

- prev 다음 오는 요소중 siblings와 동일한 형제 요소를 모두 반환함.

[예제] DOM 구조를 사용한 Selector

element 다음에 오는 element(el + el)

```
<!DOCTYPE html >
<html>
<head>
<meta charset="UTF-8">
<title>이웃 selector</title>
<script src='../lib/jquery-2.0.1.js'></script>
<script>
  $(document).ready(function(){
    $('div+p').css('background-color', '#eeeeff');
    $('ul+div').css('background-color', '#ffaana');
  });
</script>
</head>
<body>
<div>
  <p>div > div</p>
</div>
<p>div+p</p>
<ul>
  <li>ul</li>
</ul>
<div>ul + div</div>
<div>ul + div</div>
<div>ul + div</div>
</body>
</html>
```

3 Filter

- 선택된 요소중 필요한 요소만을 걸러내는 방법
- 접두문자 ":" 붙여서 사용함.
- 필터와 필터를 연결하여 사용할 수 있음.

3.1 필터의 종류

- 기본필터
- 폼필터
- 자식필터
- traverse

3.2 기본 필터

3.2.1 :animated

- 애니메이션이 동작중인 요소들과 일치하는 요소들을 반환

3.2.2 :eq(index)

- index 에 해당하는 요소 반환. (zero base)

3.2.3 :even

- 짝수 요소들만 반환(0 base)

3.2.4 :odd

- 홀수 요소들만 반환

3.2.5 :first

- 첫번째 요소만 반환

3.2.6 :last

- 마지막 요소만 반환

3.2.7 :gt(index)

- index 보다 큰 index 에 해당하는 요소들만 반환

3.2.8 :lt(index)

- index 보다 작은 index 에 해당하는 요소들을 반환

3.2.9 :header

- 모든 헤더(h1~h6)들을 반환

3.2.10 :not(selector)

- selector 요소가 아닌 요소들만 반환

3.2.11 :focus

- 현재 포커스가 위치한 요소를 반환(1.6 이상에서 지원)

3.2.12 :parent

- 텍스트 노드를 비롯해 다른 엘리먼트의 부모에 해당하는 엘리먼트 선택.

3.2.13 :empty

- 자식 노드가 없는 엘리먼트 선택.

3.2.14 :contains(text)

- 지정된 text 를 포함하는 엘리먼트 선택.

3.2.15 :has(E)

- E 에 해당하는 엘리먼트가 포함된 엘리먼트를 선택.

3.2.16 :visible

- 현재 보이는 모든 엘리먼트를 선택.
- jquery 내부에서는 넓이와 높이를 검사해서 엘리먼트가 보이는지를 판단함.
- 감춤 상태인 경우
 - display = 'none'
 - 폼 엘리먼트 속성이 type='hidden'인 경우
 - 높이와 넓이가 0 인 경우
 - 부모 엘리먼트가 감추인 경우 등.

3.2.17 :hidden

- 현재 감추어진 모든 엘리먼트를 선택

기본 필터 예제

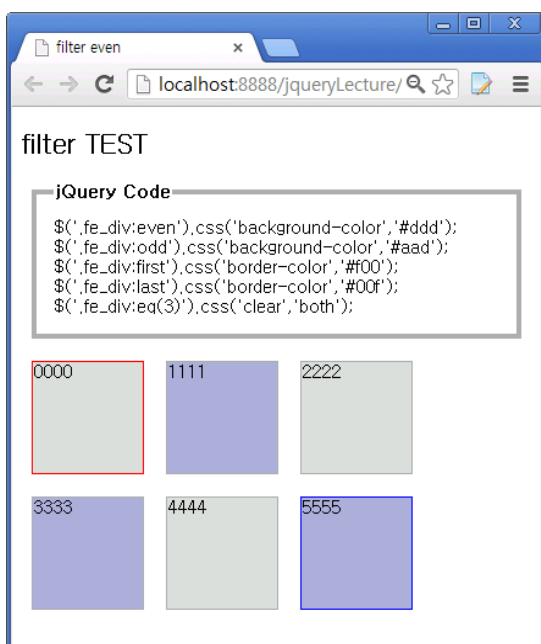
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>filter even</title>
<style>
#fe_fieldset{border:5px solid #aaa;
padding:15px;margin:10px;}
#fe_Legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
display:inline-block;float:Left;margin:10px;}
</style>
<script src = '../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
function(){
var str= "$('.fe_div:even').css('background-color','#ddd');";
str+="$('.fe_div:odd').css('background-color','#aad');";
str+="$('.fe_div:first').css('border-color','#f00');";
str+="$('.fe_div:last').css('border-color','#00f');";
str+="$('.fe_div:eq(3)').css('clear','both');";

eval(str);
document.getElementById('fe_result').innerHTML = str.split(';').join("<br/>");
}
)
</script>

</head>
```

3 Filter > 3.2 기본 필터

```
<body>
<h2>filter TEST</h2>
<fieldset id='fe_fieldset'>
  <legend id='fe_legend'>jQuery Code</legend>
  <div id='fe_result'></div>
</fieldset>
<div class='fe_div'>0000</div>
<div class='fe_div'>1111</div>
<div class='fe_div'>2222</div>
<div class='fe_div'>3333</div>
<div class='fe_div'>4444</div>
<div class='fe_div'>5555</div>
</body>
</html>
```



3.3 폼 필터

폼 요소를 선택할 때 사용되는 필터. 기본 필터 사용 방법과 동일함.

:button	:focus	:selected
:checkbox	:image	:submit
:checked	:input	:text
:disabled	:password	:hidden
:enabled	:radio	
:file	:reset	

폼 필터 예제

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

```

    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>filter form</title>
<style>
#fe_fieldset{border:5px solid #aaa;
padding:15px;margin:10px;}
#fe_Legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
display:inline-block;float:left;margin:10px;}
</style>
<script src = '../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
function(){
var str= "$(':text').css('background-color','#ddd');";
str+="$(':password').css('background-color','#faa');";
str+="$(':checked').css('background-color','#00f');";
str+="$(':selected').css('background-color','#ddf');";

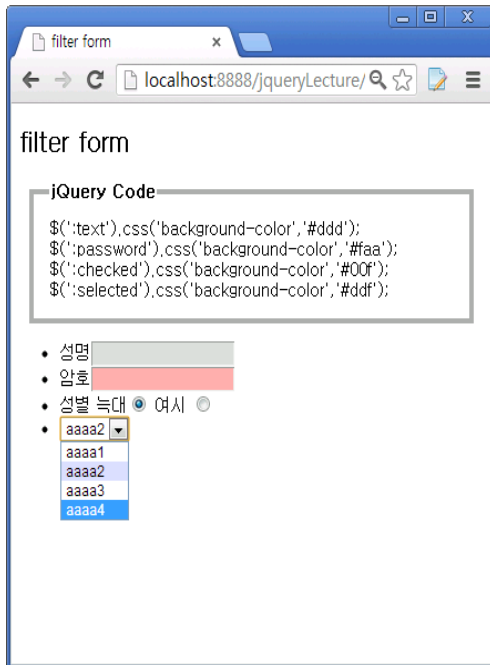
eval(str);
document.getElementById('fe_result').innerHTML = str.split(';').join("<br/>");
}
)
</script>

</head>
<body>
<h2>filter form</h2>
<fieldset id='fe_fieldset'>
<legend id='fe_Legend'>jQuery Code</legend>
<div id='fe_result'></div>
</fieldset>
<ul>
<li><label>성명</label><input type='text'>
<li><label>암호</label><input type='password'>
<li><label>성별</label>
<label>남<input type='radio' name='gender' checked='checked' /></label>
<label>여<input type='radio' name='gender' /></label>

<li><select>
<option>aaaa1</option>
<option selected='selected'>aaaa2</option>
<option>aaaa3</option>
<option>aaaa4</option>
</select>
</li>
</ul>
</body>
</html>

```

3 Filter > 3.3 폼 필터



*크롬에서는 radio/checkbox 부분 표현 안됨(파이어폭스, 오페라가능)

3.4 자식 필터

- 대부분의 자식 필터는 지정된 자식 요소들을 모두 반환한다.
- 헤더태그(h1~h6)을 사용한 이후에는 <div/> 등을 사용하여 영역을 구분해야 됨.

3.4.1 :first-child

- 엘리먼트에서 첫번째 자식에 해당하는 엘리먼트를 모두 선택 한다.
- \$('li:first-child') : 모든 첫 번째 자식 li 엘리먼트를 선택한다.
- \$('.myclass:first-child') : 모든 첫번째 자식 엘리먼트 가운데 클래스명이 myclass 인 엘리먼트를 선택한다.

3.4.2 :last-child

- 엘리먼트에서 마지막 자식 엘리먼트를 선택한다.
- \$('li:last-child') : 부모 엘리먼트에서 마지막으로 위치하는 li 엘리먼트를 모두 선택한다.
- \$('.myclass:last-child') : 부모 엘리먼트에서 마지막 자식 엘리먼트중에 클래스명이 myclass 인 엘리먼트를 선택한다.

3.4.3 :nth-child(index | odd | even | equation)

- 1 base 를 기준으로 하며 자식 엘리먼트의 종류와 상관없이 지정된 위치의 엘리먼트를 선택한다.
- \$('li:nth-child(2)') : 부모 엘리먼트에서 두번째 li 요소를 선택한다.
- \$('p:nth-child(odd)') : p 엘리먼트중에 홀수번째 요소들만 선택한다.
- \$('p:nth-child(even)') : p 엘리먼트들중에 짝수번째 요소들만 선택한다.
- \$('p:nth-child(3n+1)') : 0, 4,7,10... 번째 요소들만 선택한다.

3.4.4 :only-child

- 엘리먼트중에 형제 엘리먼트가 없는 엘리먼트를 선택한다.
- \$(':only-child') : 페이지내에서 형제 엘리먼트가 없는 자식 엘리먼트를 모두 선택한다.
- \$('p:only-child') : 페이지 내에서 형제 엘리먼트가 없는 자식 엘리먼트중 p 엘리먼트를 선택한다.

3.4.5 :not(E) :

- E 셀렉터 표현식에 해당하지 않는 엘리먼트를 모두 반환한다.
- \$(':not(.myclass)') : 클래스명이 myclass 가 아닌 모든 엘리먼트를 선택한다.
- \$('li:not(:last-child)') : li 엘리먼트중에 부모 엘리먼트에 마지막 자식 엘리먼트를 제외한 나머지 엘리먼트를 모두 선택한다.

3.4.6 :empty

- 텍스트 노드를 포함해서 자식 엘리먼트가 없는 엘리먼트를 모두 선택한다.
- \$('p:empty') : p 엘리먼트중에 자식 노드가 없는 엘리먼트를 모두 선택한다.

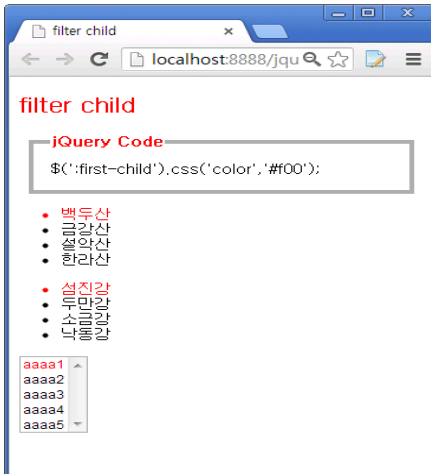
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>filter child</title>
<style>
#fe_fieldset{border:5px solid #aaa;
padding:15px;margin:10px;}
#fe_legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
display:inline-block;float:left;margin:10px;}
</style>
<script src = '../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        var str= "$(':first-child').css('color','#f00');";
        eval(str);
        document.getElementById('fe_result').innerHTML =
            str.split(';').join("<br/>");
    }
)
</script>

</head>
<body>
<h2>filter child</h2>
<fieldset id='fe_fieldset'>
    <legend id='fe_legend'>jQuery Code</legend>
    <div id='fe_result'></div>
</fieldset>
<ul id='fc_ul1'>
    <li>백두산</li>
    <li>금강산</li>
    <li>설악산</li>
    <li>한라산</li>
</ul>
<ul id='fc_ul2'>
    <li>섬진강</li>
    <li>두만강</li>
    <li>소금강</li>
    <li>낙동강</li>
</ul>

<select size=5>
    <option>aaaa1
    <option>aaaa2
    <option>aaaa3
    <option>aaaa4
```

3 Filter > 3.4 자식 필터

```
<option>aaaa5
</select>
</body>
</html>
```



- 모든 첫번째 요소의 속성이 지정되었다.

filter/filter_child2.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>filter child</title>
<style>
#fe_fieldset{border:5px solid #aaa;
padding:15px;margin:10px;}
#fe_legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
display:inline-block;float:left;margin:10px;}
</style>
<script src = '../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
function(){
var str= "$('li:nth-child(2n)').css('color', '#f00');";
eval(str);
document.getElementById('fe_result').innerHTML =
str.split(';').join("<br/>");
}
)
</script>

</head>
<body>
<h2>filter child</h2>
<fieldset id='fe_fieldset'>
<legend id='fe_legend'>jQuery Code</legend>
<div id='fe_result'></div>
</fieldset>
<ul id='fc_ul1'>
<li>백두산</li>
<li>금강산</li>
```

3 Filter > 3.4 자식 필터

```
<li>설악산</li>
<li>한라산</li>
<li>섬진강</li>
<li>두만강</li>
<li>소금강</li>
<li>낙동강</li>
</ul>
</body>
</html>
```



* 순차값은 0 이 아닌 1 부터 시작됨.

4 Traverse

기존 selector 를 사용하여 필요한 요소를 가져올 수 있으나 traverse 를 사용하면 1 차로 선택된 항목들을 대상으로 2,3 차 작업을 더 할 수 있는 장점이 있다. Selector 표현식과 DOM 탐색 메서드 가운데 어느 것을 선택해서 사용할지는 사용자 취향문제이지만 HTML 구조가 Selector 로 접근하기 까다로운 구조라면 DOM 탐색 메서드와 함께 사용하면 좀더 쉽고 빠르게 선택할 수 있을 것이다.

4.1 종류

- Filtering Method
- Miscellaneous traverse
- Tree traverse

4.2 Filtering Method

4.2.1 .eq(index)

- index 위치의 요소 선택(zero base)
- index 값이 음수인 경우는 뒤에서 부터 계산하여 선택
- `$('li').eq(1)` : li 엘리먼트중에 두번째 li 엘리먼트를 선택한다.

4.2.2 .filter(expr)

- 표현식과 일치하는 요소의 집합
- `$('li').filter(':even')` : li 엘리먼트중 짝수번째만 선택한다.(zero base)
- `$('li').filter(function(index){ return index; })` : index 번째를 반환

4.2.3 .first()

- 매개변수는 없으며 첫번째 엘리먼트를 선택한다.

4.2.4 .has(selector)

- 표현식에 해당하는 자손 엘리먼트를 지닌 엘리먼트만 선택
- `$('ul').has('li')` : ul 엘리먼트안에 li 엘리먼트를 선택한다.

4.2.5 .is(expr)

- 어떤 요소의 현재 상태값을 검사하여 boolean 타입으로 반환한다.

4.2.6 .last()

- 매개변수는 없으면 마지막 엘리먼트를 선택한다.

4.2.7 .map(callback)

- 함수를 호출하여 처리하여 그 결과를 반환하여 jQuery 객체를 생성 한다.
- `$('#checkbox').map(function(){ return this.checked=true; });` : 모든 체크박스의 체크한다.

4.2.8 .not(expr)

- 엘리먼트들중에 조건에 맞지 않는 엘리먼트를 선택 반환.
- `$('li').not(':even')` : li 엘리먼트에서 짝수번째가 아닌 엘리먼트를 반환.

4.2.9 .slice(start [, end])

- `start <= N < end` 위치에 해당하는 요소 반환.
- zero base

- end 값이 생략되면 마지막까지 선택함.

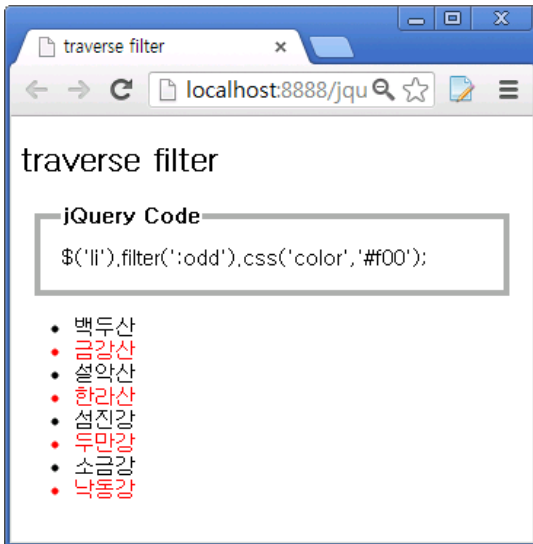
4.2.10 \$('td:eq(0)') == \$('td').eq(0) 과 같은 의미.

[예제] 홀수 번째만 선택

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>traverse filter</title>
<style>
#fe_fieldset{border:5px solid #aaa;
    padding:15px;margin:10px;}
#fe_legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
    display:inline-block;float:left;margin:10px;}
</style>
<script src = '../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        var str= "$('li').filter(':odd').css('color','#f00');";
        eval(str);
        document.getElementById('fe_result').innerHTML =
            str.split(';').join("<br/>");
    }
)
</script>

</head>
<body>
<h2>traverse filter</h2>
<fieldset id='fe_fieldset'>
    <legend id='fe_legend'>jQuery Code</legend>
    <div id='fe_result'></div>
</fieldset>
<ul id='fc_ul1'>
    <li>백두산</li>
    <li>금강산</li>
    <li>설악산</li>
    <li>한라산</li>
    <li>섬진강</li>
    <li>두만강</li>
    <li>소금강</li>
    <li>낙동강</li>
</ul>
</body>
</html>
```

* index 값은 0번 부터 시작



[예제] filter 에 함수 사용하기

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>traverse filter</title>
<style>
#fe_fieldset{border:5px solid #aaa;
padding:15px;margin:10px;}
#fe_legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
display:inline-block;float:left;margin:10px;}
</style>
<script src = '../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        var str= "$('li').filter( function(index) { return index%3==0}).css('color', '#f00');";
        eval(str);
        document.getElementById('fe_result').innerHTML =
            str.split(';').join("<br/>");
    }
)
</script>

</head>
<body>
<h2>traverse filter</h2>
<fieldset id='fe_fieldset'>
    <legend id='fe_legend'>jQuery Code</legend>
    <div id='fe_result'></div>
</fieldset>
<ul id='fc_ul1'>
    <li>백두산</li>
    <li>금강산</li>
    <li>설악산</li>
    <li>한라산</li>
    <li>섬진강</li>

```

4 Traverse > 4.2 Filtering Method

```
<li>두만강</li>
<li>소금강</li>
<li>낙동강</li>
</ul>
</body>
</html>
```



[예제] slice 사용하기

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>traverse filter slice</title>
<style>
#fe_fieldset{border:5px solid #aaa;
padding:15px;margin:10px;}
#fe_legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
display:inline-block;float:left;margin:10px;}
</style>
<script src = '../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
function(){
var str= "$('li').slice(2,4).css('color', '#f00');";
eval(str);
document.getElementById('fe_result').innerHTML =
str.split(';').join("<br/>");
}
)
</script>

</head>
<body>
<h2>traverse filter slice</h2>
<fieldset id='fe_fieldset'>
<legend id='fe_legend'>jQuery Code</legend>
<div id='fe_result'></div>
</fieldset>
<ul id='fc_ul1'>
```

4 Traverse > 4.2 Filtering Method

```
<li>백두산</li>
<li>금강산</li>
<li>설악산</li>
<li>한라산</li>
<li>섬진강</li>
<li>두만강</li>
<li>소금강</li>
<li>낙동강</li>
</ul>
</body>
</html>
```

* slice 의 범위는 $x \leq N < y$



4.3 Tree traverse Method

DOM 의 Tree 구조를 기준으로 요소를 반환함.

4.3.1 .children([selector])

- 선택된 객체의 자식중 selector 와 동일한 요소를 반환

4.3.2 .closest([selector])

- 선택된 객체의 가장 가까운 조상 요소를 가져옴(DOM Tree 구조상)

4.3.3 .find([selector])

- 선택된 객체에서 selector 와 동일한 요소를 반환

4.3.4 .next([selector])

- selector 와 일치하는 형제 요소를 가져옴. 만약 일치하는 요소가 없다면 다음 항목으로 이동하여 요소를 찾음.

4.3.5 .parent([selector])

- selector 와 일치하는 부모 요소를 반환.

4.3.6 .prev([selector])

- selector 와 일치하는 앞 형제 요소를 반환.

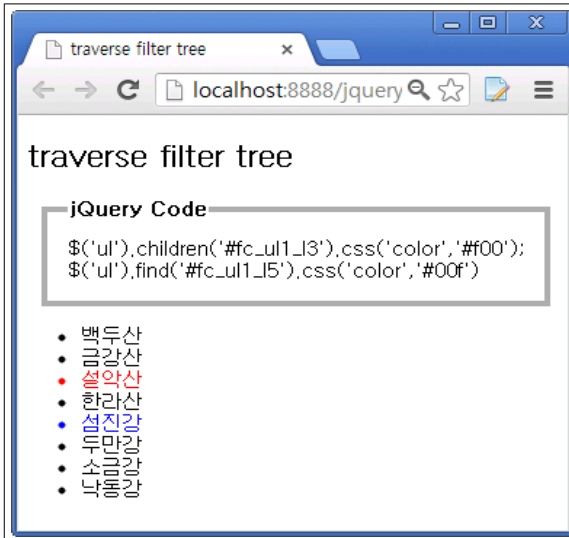
4.3.7 .siblings([selector])

- selector 와 일치하는 형제 요소를 반환.

4 Traverse > 4.3 Tree traverse Method

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>traverse filter tree</title>
<style>
#fe_fieldset{border:5px solid #aaa;
    padding:15px;margin:10px;}
#fe_legend{font-weight:bold;}
.fe_div{ border:1px solid #aaa; width:100px;height:100px;
    display:inline-block;float:left;margin:10px;}
</style>
<script src = '../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        var str= "$('ul').children('#fc_ul1_l3').css('color','#f00');";
        str += "$('ul').find('#fc_ul1_l5').css('color','#00f');";
        eval(str);
        document.getElementById('fe_result').innerHTML =
            str.split(';').join("<br/>");
    }
)
</script>

</head>
<body>
<h2>traverse filter tree</h2>
<fieldset id='fe_fieldset'>
    <legend id='fe_legend'>jQuery Code</legend>
    <div id='fe_result'></div>
</fieldset>
<ul id='fc_ul1'>
    <li id='fc_ul1_l1'>백두산</li>
    <li id='fc_ul1_l2'>금강산</li>
    <li id='fc_ul1_l3'>설악산</li>
    <li id='fc_ul1_l4'>한라산</li>
    <li id='fc_ul1_l5'>섬진강</li>
    <li id='fc_ul1_l6'>두만강</li>
    <li id='fc_ul1_l7'>소금강</li>
    <li id='fc_ul1_l8'>낙동강</li>
</ul>
</body>
</html>
```



4.4 기타 traverse

4.4.1 .add(selector) | .add(element) | .add(html)

- 일치하는 요소의 집합에 요소를 더한다.

4.4.2 .addSelf()

- 현재 설정 스택에 요소의 이전 설정을 추가한다.

4.4.3 .is([selector])

- 엘리먼트들에서 selector 와 일치하는 엘리먼트가 있는지 여부를 반환 한다.
- 반환값 : boolean
- 새로운 객체를 생성하지 않는다.

4.4.4 .contents()

- 텍스트 및 주석 노드를 포함하여 일치하는 요소 집합의 자식 집합을 반환

4.4.5 .end()

- 메서드 체인에서 가장 마지막 DOM 처리 메서드가 실행되기 이전 상태의 jQuery 객체를 반환 한다.

5 Manipulation

지정된 요소의 전후 안팎에 요소를 추가하는 일반적인 기능

- 종류
 - wrap(html) | wrap(elem)
 - append(content) | appendTo(expr)
 - prepend(content) | prependTo(expr)
 - before(content) | after(content)
 - clone()
 - insertBefore(expr) | insertAfter(expr)
 - remove(expr)
 - empty()

5.1 wrap(html) | wrap(elem)

지정된 요소의 html 이나 elem 을 씌운다.

]

Example 1

```
var wrap = "<div class='wrap_class'></div>";
$('div').wrap(wrap);
```

Before

```
<div>Test wrap</div>
```

After

```
<div class='wrap_class'><div>Test wrap</div></div>
```

Example 2

```
$('#p').wrap($('#wrap_cont'));
```

Before

```
<p>Test wrap(elem)</p><div id='wrap_cont'>wrap</div>
```

After

```
<div id='wrap_cont'>wrap<p>Test wrap(elem)</p></div>
```

5.2 append(content) | appendTo(expr)

특정 요소 안의 뒤쪽에 content 를 삽입한다.

- content : 삽입할 요소
- expr : jquery 를 문자열로 표현한 요소

Example 1

```
$('#append_div').append("<font color=red>JOBTC.KR</font>");
```

Before

```
<div id='append_div'>Append</div>
```

After

```
<div id='append_div'>Append<font color=red>JOBTC.KR</font></div>
```

Example 2

5 Manipulation > 5.2 append(content) | appendTo(expr)

```
$('#append_target').append($('#append_p'));
```

Before

```
<p id='append_p'>Append Object</p>
<p id='append_target'>Append!!!</p>
```

After

```
<p id='append_target'>Append!!!<p id='append_p'>Append Object</p></p>
```

5.3 prepend(content) | prependTo(expr)

지정된 요소 **안의 앞쪽**에 content 나 expr 를 추가 한다.

5.4 before(content) | after(content)

지정된 요소 **바깥의 앞쪽**(before())나 **뒤쪽**(after())에 요소를 추가한다.

5.5 clone()

지정된 요소를 복사한다.

5.6 insertBefore(expr) | insertAfter(expr)

- insertBefore() == .before()와 동일
- insertAfter() == .after()와 동일

5.7 remove(expr)

지정된 요소를 제거 한다.

5.8 empty()

지정된 요소를 비운다.

6 Core

6.1 \$(expr, context)

Parameter

- expr : 찾는 표현식
- context : Element | jQuery

Example

- \$('div') : <div> 선택
- \$('div > p') : <div>안에 <p> 선택
- \$('input:radio', document.forms[0]) : 첫번째 폼의 라디오 버튼 선택

[예제] 첫번째 버튼의 바탕색 바꾸기

jQuery

```
$('#input:button', document.forms[0]).css('background-color', '#aaf');
```

HTML

```
<form>
  <label>첫번째폼의 라디오<input type='button' value='first' name='radio'/></label>
</form>
<form>
  <label>두번째폼의 라디오<input type='button' value='second' name='radio'/></label>
</form>
```

6.2 \$(html) | \$(elements)

Html 를 추가하거나 선택할 수 있다.

Html 을 특정 위치에 추가하는예

jQuery

```
$('<h1>hi jobtc</h1>').appendTo('#core_append_div');
```

html

```
<div id='core_append_div'></div>
```

6.3 \$(fn)

\$(document).ready(function(){})을 짧게 표현.

\$(function(){ ... })

6 Core > 6.4 length | size()

6.4 length | size()

지정된 요소의 개수를 구한다.

Html 을 특정 위치에 추가하는예
jQuery
<pre>var len = \$('div').length; var s = \$('div').size(); alert(len + ', ' + s); // 4,4</pre>
html
<pre><div></div><div></div><div></div><div></div></pre>

6.5 get() | get(n)

해당되는 모든 요소들을 배열로 가져오거나 n 번째 요소를 가져옴.

jQuery
<pre>var len = \$('div').get(); \$('#get_result').html(len.join('
'));</pre>
html
<pre><div></div><div></div><div></div><div></div> <div id='get_result'></div></pre>

6.6 each(fn)

지정된 항목을 순환 검색하여 fn 를 수행한다.

- fn : function

jquery
<pre>var bg = ['#f00', '#00f', '#0f0', '#ff0'] \$(document).ready(function(){ \$('.each_div1').each(function(i){ \$(this).css('background-color',bg[i]); }) })</pre>
html
<pre><div class='each_div1'>11</div> <div class='each_div1'>22</div> <div class='each_div1'>33</div> <div class='each_div1'>44</div></pre>

6.7 index(subject) | find()

index() : 지정된 요소를 검색하여 위치값을 반환함. 0 base 이며 찾지 못하면 -1 을 반환.

find() : 저장된 요소를 검색하여 배열 타입으로 반환

- subject : 찾고자 하는 요소. element

```
var bg = [ '#fdd', '#ddf', '#dfd', '#ffd' ]
$(document).ready(function(){
  var index = $('div').index($('.index_div2'));
  $('.index_div' + (index+1)).css('background',bg[index]);
})
```

```
<div class='index_div1'>11</div>
<div class='index_div2'>22</div>
<div class='index_div3'>33</div>
<div class='index_div4'>44</div>
```

6.8 eq(pos) | lt(pos) | gt(pos)

같거나 작거나 큰 요소들을 찾는다.

- pos : Number

```
var bg = [ '#fdd', '#ddf', '#dfd', '#ffd' ]
$(document).ready(function(){
  var eq = $('.sub_div1:eq(2)');
  var lt = $('.sub_div1:lt(2)');
  var gt = $('.sub_div1:gt(2)');

  lt.css('background-color', bg[0]);
  eq.css('background-color', bg[1]);
  gt.css('background-color', bg[2]);
})
```

```
<div class='sub_div1'>11</div>
<div class='sub_div1'>22</div>
<div class='sub_div1'>33</div>
<div class='sub_div1'>44</div>
```

7 Attribute 함수

HTML 요소에서 name 속성을 제외한 나머지 부분을 제어할 수 있다. 정의되지 않은 속성을 임의로 지정할 수는 있으나 브라우저별 호환성이 없으므로 주의해야 한다.

7.1.1 .attr()

- 엘리먼트 집합에 속한 첫 번째 엘리먼트의 속성을 가져온다.
- .attr 메서드의 장점
 - 편리성 : jQuery 객체나 jQuery 메서드 체인 코드에서 바로 활용할 수 있다.
 - 크로스 브라우징 지원 : 브라우저의 종류에 따라 속성 이름이 다르거나 속성값을 다르게 표현할 때가 있다. 이러한 크로스 브라우징의 이슈들을 최소화 할 수 있다.
- 사용 패턴
 - .attr(attributeName)
 - .attr(attributeName, value)
 - .attr(map)
 - .attr(attributeName, function)

7.1.2 .removeAttr()

7.1.3 .val()

- value 값을 가져옴.

7.1.4 .text() | .html()

- 해당 형식으로 값을 가져오거나 대입함.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Attributes function</title>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        $btn = $('#af_btn');
        var v = $btn.attr('value') + "," + $btn.attr('type');
        $btn.click(function(){
            $btn.attr('value', 'atfer jobtc');
        });

        $('#af_html_btn').click(function(){
            var str = '<font color=red>jobtc.kr</font>';
            $('#af_div').html(str);
        });

        $('#af_text_btn').click(function(){
            var str = '<font color=red>jobtc.kr</font>';
            $('#af_div').text(str);
        });
    }
);
```


7 Attribute 함수 > 7 Attribute 함수

```

}
</script>
</head>
<body>
<input type='button' value='jobtc' id='af_btn'>
<input type='button' value='jobtc' id='af_html_btn'>
<input type='button' value='jobtc' id='af_text_btn'>

<div id='af_div'></div>
</body>
</html>

```

8 CSS

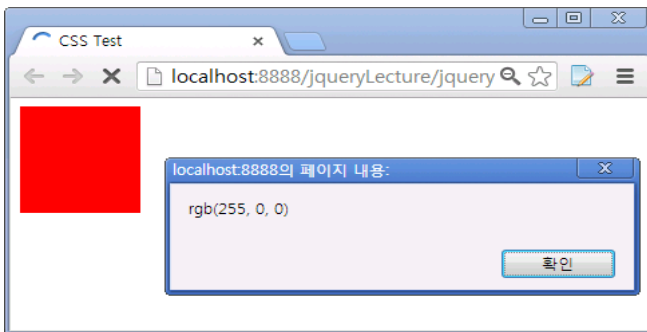
8.1 .css()

사용방법

- .css(프로퍼티명)
 - 프로퍼티의 속성을 가져온다.
- .css(프로퍼티명, 값)
 - 지정된 프로퍼티의 속성을 값으로 변경한다.
- .css(프로퍼티명, function(index, value))
 - index : 현재 선택된 항목의 위치
 - value : 현재 선택된 항목의 object
 - index, value 는 생략할 수 있음.
 - 파라미터를 하나만 사용하면 무조건 index 로 사용됨.
- .css(map)
 - map 형태로 지정.{key:value}
 - 복수 지정을 한번에 할때 사용.

[사용예] <div>태그의 바탕색 가져오기

```
var v = $('div').css('background-color');
alert(v);
```



[사용예] div 태그에 바탕색 지정하기

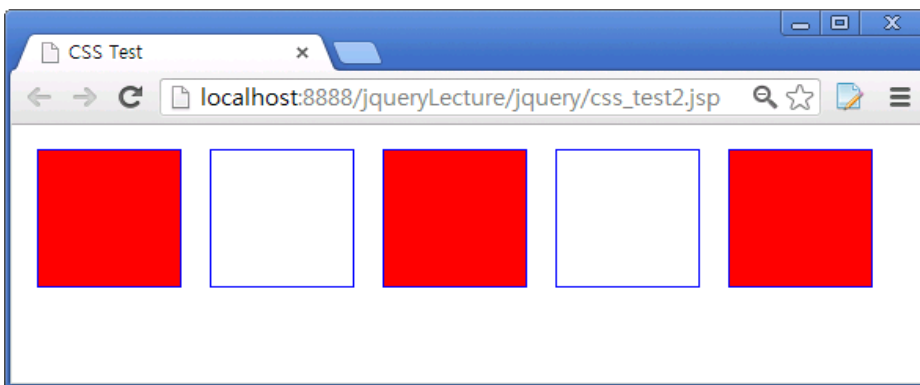
```
$('#div').css('background-color', 'blue');
```



8 CSS > 8.1 .css()

[사용예] index 값이 짝수인 경우 바탕색 변경하기

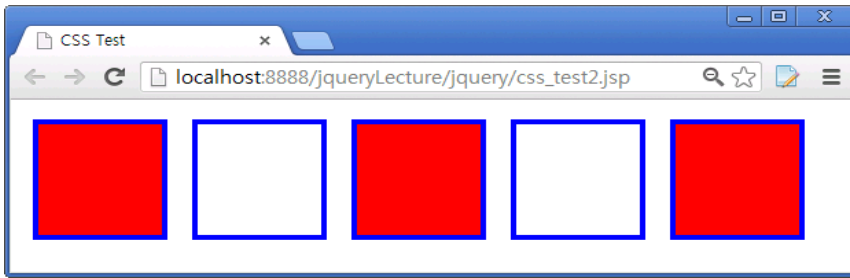
```
$(document).ready(  
  function(){  
    $('div').css({'float':'left', 'margin':'10px'});  
    $('div').css({'width':'100px','height':'100px', 'border':'5px solid #00f' });  
  
    var v = $('div').css('background-color',  
      function(index, v){  
        if(index%2==0 ){  
          return 'red';  
        }  
      }  
    );  
  }  
);
```



[사용예] map 을 사용하여 복수 속성 지정하기

```
$(document).ready(  
  function(){  
    $('div').css({'float':'left', 'margin':'10px'});  
    $('div').css({'width':'100px','height':'100px', 'border':'5px solid #00f' });  
  
    var v = $('div').css('background-color',  
      function(index, v){  
        if(index%2==0 ){  
          return 'red';  
        }  
      }  
    );  
  }  
);
```

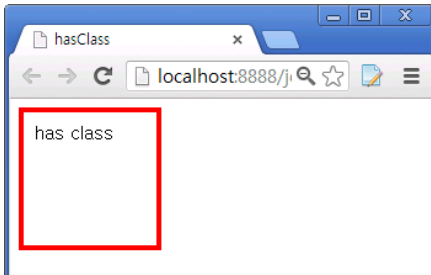
8 CSS > 8.1 .css()



8.2 .hasClass(className)

- className 이 있는지 판단한다.(boolean 타입으로 반환)

```
$(document).ready(  
  function(){  
    var b = $('div').hasClass('hasClass_div1');  
    if(b){  
      $('.hasClass_div1').css('border', '5px solid #f00');  
      $('.hasClass_div1').css('width', '100px');  
      $('.hasClass_div1').css('height', '100px');  
      $('.hasClass_div1').css('padding', '10px');  
    }  
  }  
)
```

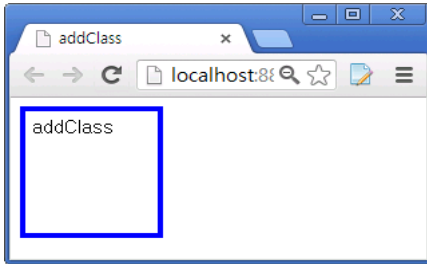


8.3 .addClass()

- .addClass(className)
- .addClass(function(index, currentClass))
- 새로운 클래스 속성을 추가한다.

```
$(document).ready(  
  function(){  
    var str={'width':'100px', 'height':'100px',  
            'border':'5px solid #00f', 'padding':'5px'};  
    $('div').addClass('addClass_div');  
    $('.addClass_div').css(str);  
  }  
)
```

8 CSS > 8.3 .addClass()



8.4 .removeClass()

- 특정 클래스 속성을 삭제한다.
 - .removeClass(className)
 - .removeClass(function(index, class))
 - index : 같은 클래스들의 색인 번호
 - class : 이벤트가 발생한 class name

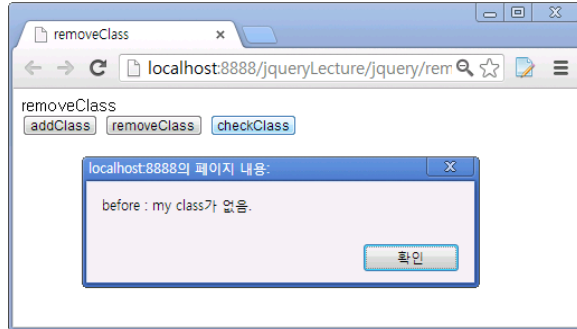
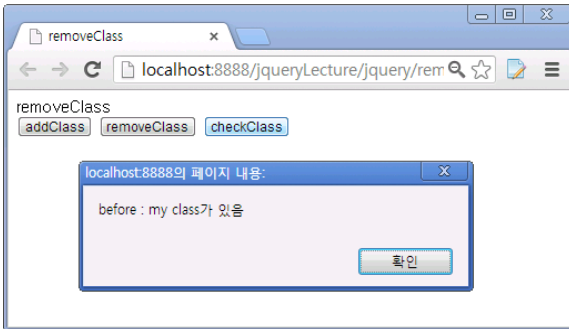
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>removeClass</title>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        $('#addClassBtn').click(addClass);
        $('#removeClassBtn').click(removeClass);
        $('#checkClassBtn').click(checkClass);
    }
);

function addClass(){
    $('div').addClass('my');
}
function removeClass(){
    $('div').removeClass('my');
}
function checkClass(){
    var my = $('div').hasClass('my');
    if(my){
        alert('before : my class가 있음');
    }
    else{
        alert('before : my class가 없음. ');
    }
}
</script>
</head>
<body>
<div class='my'>removeClass</div>

<button id='addClassBtn'>addClass</button>
<button id='removeClassBtn'>removeClass</button>
```

8 CSS > 8.4 .removeClass()

```
<button id='checkClassBtn'>checkClass</button>
</body>
</html>
```



8.5 8.5 .toggleClass()

- 선택된 클래스의 기능을 끄고 켜는 기능.
 - .toggleClass(className)
 - .toggleClass(className, switch)
 - .toggleClass(function(index, class) [, switch])

지정된 class 속성을 모두 지우고 선택된 항목에만 class 속성을 toggle 시킴

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>toggleClass</title>
<style>
.tc_div { width:200px; height:200px;
    display:table-cell; vertical-align: middle;
    text-align: center;
}
.chk{ border:2px solid #aaa;}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        $d = $(''.tc_div');
        $d.click( function(){
            $d.removeClass('chk');
            $(this).toggleClass('chk');
        })
    }
);
</script>
</head>
<body>
<div class='tc_div'>toggle test</div>
<div class='tc_div'>toggle test</div>
<div class='tc_div'>toggle test</div>

</body>
</html>
```

8.6 .position() | .width() | .height()

- 선택된 객체의 위치값이나 넓이, 높이 정보를 가져오거나 지정할 수 있다.
- position()의 속성값 : top | left

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>position | width | height</title>
<style>
.tc_div { width:200px; height:200px;
    display:table-cell; vertical-align: middle;
    text-align: center;
}
</style>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(
    function(){
        $d = $('.tc_div');
        $d.click( function(){
            var str = '';
            var pos = $(this).position();
            var w = $(this).width();
            var h = $(this).height();
            str = 'top='+pos.top;
            str += '<br/>left=' + pos.left;
            str += '<br/>width=' + w;
            str += '<br/>height=' + h;

            $('#ps_div').html(str);
        })
    }
);
</script>
</head>
<body>
<div class='tc_div'>toggle test</div>
<div id='ps_div'></div>
</body>
</html>
```

9 Event

9.1 종류

- load(fn) | unload(fn)
- blur(fn) | blur() | focus(fn) | focus()
- hover(over, out)
- change(fn) | scroll(fn)
- click() | click(fn) | dbclick(fn)
- keydown(fn) | keypress(fn) | keyup(fn)
- mousedown(fn) | mouseenter(fn) | mouseleave(fn) | mouseout(fn) | mouseup(fn) | mousemove(fn) | mouseover(fn)
- ready(fn)
- resize(fn)
- select(fn) | select()
- submit(fn) | submit()
- bind(type, data, fn) | unbind(type, fn)
- one(type, data, fn)
- error(fn)
- trigger(type)
- toggle(even, odd)

9.2 직렬화 관련 메서드

- serialize() : 폼의 값을 get type 으로 반환한다.
- serializeArray() : 폼의 값을 jQuery 배열 객체로 반환 한다.

serialize 예제

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>serialize(직렬화)</title>
<script src='../Lib/jquery-1.8.3.js'></script>
<style>
.sel_label{ display:inline-block; width:80px; text-align:right;}
</style>
<script>
$(document).ready(
    function(){
        $('#se_form').submit( function(){
            var data = $(this).serialize();
            $('#af_div').text(data);
            return false;
        });
    }
)
</script>
```



```

</head>
<body>
<h3>Ajax Serialize Test</h3>
<section>
  <form name='se_form' id='se_form'>
    <label class='sel_Label'>Nic Name</label>
    <input type='text' name='nickname' id='a' value='park' /><br/>

    <label class='sel_Label'>Pass word</label>
    <input type='password' name='pwd' id='c' value='1234' /><br/>
    <input type='hidden' name='mid' id='b' value='a001' /><br/>
    <input type='submit' id='se_submit' value='Submit'>
  </form>
</section>

<section>
<fieldset>
<legend>Result</legend>
<div id='af_div'></div>
</fieldset>
</section>
</body>
</html>

```

* mid 값은 hidden 속성이여서 화면에 표시 되지 않음.



9.3 hover(over, out)

```

<%@ page Language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>hover</title>
<style>
#hover_a{width:100px; height:100px;border:1px solid #fff;}
</style>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(function(){
  $('#hover_a').hover(function(){ //over
    $(this).css({
      'text-decoration':'underline',
      'color':'#f00',
      'background-color':'#ddd'
    });
  });
});

```

```

    });
    $('#af_div').text('mouse over !!!');
}, function() { // out
    $(this).css({
        'text-decoration': 'none',
        'color': '',
        'background-color': ''
    });
    $('#af_div').text('mouse out !!!');
});
})
</script>
</head>
<body>
<section>
    <div id='hover_a'>jobtc</div>
</section>

<section>
<fieldset>
<legend>Result</legend>
<div id='af_div'></div>
</fieldset>
</section>
</body>
</html>

```

9.4 keykeydown(fn) | keypress(fn) | keyup(fn)

- event.keyCode 값에 의해 이벤트가 발생한 키 코드값을 가져 올 수 있음.
- event.type 에 의해 키보드의 상수값을 얻을 수 있다.

key event

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>key event</title>
<style>
#key_text { width:200px; height:20px; font-size:20px; }
.key_lbl { width:100px; text-align:right; display:inline-block; }
</style>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(function(){
    $('#key_text').keydown(function(ev){
        var str= 'key code=' + ev.keyCode + ', key event=' + ev.type;
        $('#af_div').text(str);
    })
})
</script>
</head>
<body>
<h3>keypress | keyup | keydown </h3>
<label class='key_lbl'>Input</label><input type='text' id='key_text' /><br/>

```

```
<fieldset>
<legend>Result</legend>
<div id='af_div'></div>
</fieldset>

</body>
</html>
```

9.5 trigger(fn)

이벤트를 자동으로 발생 시킨다.

- 절차
 - 이벤트를 처리할 함수를 지정
 - 이벤트 추가
 - trigger() 수행

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>trigger</title>
<style>
#key_text { width:200px;height:20px;font-size:20px;}
.key_lbl{width:100px;text-align:right;display:inline-block;}
</style>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
function chk(ev){
    $('#af_div').text("이벤트 발생 !!!");
}

$(document).ready(function(){
    $('#trigger_btn').click(chk);
    $('#trigger_btn').trigger('click');
});
</script>
</head>
<body>
<h3>trigger </h3>
<button type='button' id='trigger_btn'>누르지 않아도 발생</button>

<fieldset>
<legend>Result</legend>
<div id='af_div'></div>
</fieldset>

</body>
</html>
```

9.6 toggle(even, odd)

- even : toggle 이 짝수번째 발생했을 경우
- odd : toggle 이 홀수번째 발생했을 경우
- 1.9.X 버전에서는 even, odd 기능이 사라짐.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>toggle</title>
<style>
#key_text { width:200px;height:20px;font-size:20px;}
.key_lbl{width:100px;text-align:right;display:inline-block;}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
function evenFunc(){
    $('#af_div').text("짝수 번째 이벤트 발생 !!!");
}
function oddFunc(){
    $('#af_div').text("홀수 번째 이벤트 발생 !!!");
}

$(document).ready(function(){
    $('#toggle_btn').toggle(evenFunc, oddFunc);
})
</script>
</head>
<body>
<h3>toggle </h3>
<button type='button' id='toggle_btn'>Toggle Button</button>

<fieldset>
<legend>Result</legend>
<div id='af_div'></div>
</fieldset>

</body>
</html>
```

9.7 focus() | focus(fn)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>toggle</title>
<style>
```

9 Event > 9.7 focus() | focus(fn)

```
#key_text { width:200px;height:20px;font-size:20px;}
.key_lbl{width:100px;text-align:right;display:inline-block;}
</style>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
function focusFunc(){
    $('#focus_message').text("아이디는 영숫자만 입력하세요.");
}

$(document).ready(function(){
    $('#focus_text').focus(focusFunc);
})
</script>
</head>
<body>
<h3>focus</h3>
<label>아이디 </label><input type='text' id='focus_text'>
<output id='focus_message' />
</body>
</html>
```

9.8 bind(event, data, fn) | bind(event, fn)

이벤트들을 map 형태로 묶어서 지정할 수 있다.

키를 누를때나 마우스를 클릭할 때 이벤트 발생

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>bind</title>
<style>
#key_text { width:200px;height:20px;font-size:20px;}
.key_lbl{width:100px;text-align:right;display:inline-block;}
</style>
<script src='../Lib/jquery-1.8.3.js'></script>
<script>
function bindFunc(ev){
    if(ev.type == 'keydown')
        $('#bind_message').text('press');
    else
        $('#bind_message').text("click");
}

$(document).ready(function(){
    $('#bind_text').bind('keydown click', bindFunc);
})
</script>
</head>
<body>
<h3>bind </h3>
<label>아이디 </label><input type='text' id='bind_text'>
<output id='bind_message' />
</body>
</html>
```

9 Event > 9.8 bind(event, data, fn) | bind(event, fn)

9.9 one(type, data, fn)

지정된 이벤트를 한번만 실행한다.

10 effect

- Animation Speed
 - slow : 600/1000 초
 - fast : 200/1000 초
 - default : 400/1000 초
 - 속도는 n/1000 초 단위로 설정할 수 있음.(효과가 시작되고 끝날때 까지 걸리는 시간을 의미)
- callback : 효과가 끝나면 호출되는 함수를 의미

10.1 종류

- show() | show(speed [, callback])
- hide() | hide(speed [, callback])
- toggle() | toggle(speed [, callback])
- slideDown([speed] [, callback]) | slideUp([speed] [, callback])
- slideToggle([speed] [,callback])
- fadeIn([speed] [,callback]) | fadeOut([speed] [,callback])
- fadeTo([speed] , opacity [,callback])
- animate(params [,speed] [, easing] [, callback])

10.2 현재 상태값 체크

- 요소명.is(":visible") : 현재 상태가 보여져 있는 상태
- 요소명.is(":hidden") : 현재 상태가 감추어져 있는 상태

10.3 show

- 요소를 보이게 함.
- show()
- show(speed [, callback])

```
<!DOCTYPE HTML >
<html>
<head>
<title> show </title>
<meta charset='utf-8'>
<style>
#show_div{ border:1px solid #aaa;width:200px;
            text-overflow:clip;}
.show_btn{ width:100px;height:30px}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(function(){
    $('#show_show').click(function(){
        $('#show_div').show();
    })
    $('#show_show2').click(function(){
```

10 effect > 10.3 show

```
    $('#show_div').show('slow');
  })
  $('#show_show3').click(function(){
    $('#show_div').show(2000);
  })

  $('#show_hide').click(function(){
    $('#show_div').hide();
  })
  $('#show_hide2').click(function(){
    $('#show_div').hide('slow');
  })
  $('#show_hide3').click(function(){
    $('#show_div').hide(2000);
  })
})
</script>
</head>

<body>
  <button id='show_show' class='show_btn'>Show</button>
  <button id='show_show2' class='show_btn'>Slow Show</button>
  <button id='show_show3' class='show_btn'>speed Show</button><br/>
  <button id='show_hide' class='show_btn'>Hide</button>
  <button id='show_hide2' class='show_btn'>Slow Hide</button>
  <button id='show_hide3' class='show_btn'>speed Hide</button>
  <div id='show_div'>
    <label>speed option :</label>
    <ul>
      <li>slow
      <li>fast
      <li>nomal
      <li>n/1000
    </ul>
  </div>
</body>
</html>
```

10.4 hide

- 요소를 숨김
- hide()
- hide(speed [, callback])

10.5 toggle

- show() 와 hide()의 기능을 반복함.
- toggle()
- toggle(speed [, callback])

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>effect toggle</title>
```


10 effect > 10.5 toggle

```
<style>
.ef_box{ border:5px solid #aaa;
         white-space:nowrap;
         padding:10px;width:350px;
}
</style>

<script src='../lib/jquery-1.8.3.js'></script>

<script>
$(document).ready(function(){
    $box = $('.ef_box');

    $('#toggle').click(
        function(){ $box.toggle(); }
    );

    $('#toggle_show').click(
        function(){ $box.toggle('slow'); }
    );

    $('#toggle_speed').click(
        function(){ $box.toggle(2000); }
    );
}
)
</script>
</head>
<body>
<header>
    <h1>Toggle : nomal | show | n/1000<hr/></h1>
</header>
<button type='button' id='toggle'>Toggle</button>
<button type='button' id='toggle_show'>Toggle slow</button>
<button type='button' id='toggle_speed'>Toggle Speed</button>
<p/>
<div class='ef_box'>
    Toggle 버튼을 클릭하면 사라졌다 나타났다 ^^<br/>
    Toggle 버튼을 클릭하면 사라졌다 나타났다 ^^<br/>
    Toggle 버튼을 클릭하면 사라졌다 나타났다 ^^<br/>
    Toggle 버튼을 클릭하면 사라졌다 나타났다 ^^<br/>
</div>
</body>
</html>
```

10.6 slideDown | slideUp | slideToggle

- slideDown([speed] [, callback])
- slideUp([speed] [, callback])
- slideToggle([speed] [,callback])
- slideDown : 숨겨진 속성이여야 효과를 볼 수 있다. 숨겨진 요소들을 위에서 부터 아래로 차차 보여짐.
- slideUp : 이미 보여진 요소를 아래에서 부터 위로 차차 숨겨짐.

```
<!DOCTYPE html>
<html>
```

10 effect > 10.6 slideDown | slideUp | slideToggle

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>slideDown | slideUp | slideToggle</title>
<style>
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready( function(){
    $('#sl_slideup').click(function(){
        $('#sl_div').slideUp(2000, function(){});
    });

    $('#sl_slidedown').click(function(){
        $('#sl_div').hide();
        $('#sl_div').slideDown(2000,
            function(){
                //TODO
            });
    });

    $('#sl_slidetoggle').click(function(){
        $('#sl_div').slideToggle(2000, function(){});
    });

});
</script>
</head>
<body>
<button id='sl_slideup'>Sliding UP</button>
<button id='sl_slidedown'>Sliding DOWN</button>
<button id='sl_slidetoggle'>Sliding Toggle</button>

<div id='sl_div'>
    <img src='a.jpg'>
</div>

</body>
</html>
```

10.7 fadeIn([speed] [.callback]) | fadeOut([speed] [.callback])

지정된 요소를 서서히 보이게 하거나 감춘다.

```
<!DOCTYPE HTML >
<html>
<head>
<title> fadeIn | fadeOut </title>
<meta charset='utf-8'>
<style>
#show_div{ border:1px solid #800000;
    width:200px; height:200px;
    background-color:#c0c0c0;
    }
.show_btn{ width:100px;height:30px}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
```

10 effect > 10.7 fadeIn([speed] [.callback]) | fadeOut([speed] [.callback])

```
<script>
$(document).ready(function(){
  $('#fade_in').click(function(){
    $('#show_div').fadeIn(2000);
  })
  $('#fade_out').click(function(){
    $('#show_div').fadeOut(2000);
  })
})
</script>
</head>

<body>
<button id='fade_in' class='fade_in_btn'>fade in</button>
<button id='fade_out' class='fade_out_btn'>fade out</button>
<div id='show_div'></div>
</body>
</html>
```

10.8 fadeTo([speed] , opacity [.callback])

opacity : 투명도 (0~1) 1:불투명, 0:투명
요소의 투명도를 지정된 opacity 까지 서서 진행시킨다.

지정된 opacity 까지 fade 를 진행

```
<!DOCTYPE HTML >
<html>
<head>
<title> fadeTo </title>
<meta charset='utf-8'>
<style>
#show_div{ border:1px solid #800000;
width:200px; height:200px;
background-color:#c0c0c0;
}
.show_btn{ width:100px;height:30px}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(function(){
  $('#fade_in').click(function(){
    $('#show_div').fadeTo(2000,0.5);
  })
  $('#fade_out').click(function(){
    $('#show_div').fadeTo(2000,0);
  })
})
</script>
</head>

<body>
<button id='fade_in' class='fade_in_btn'>fadeTo 0.5 </button>
<button id='fade_out' class='fade_out_btn'>fadeTo 0.0</button>
<div id='show_div'></div>
</body>
</html>
```

10 effect > 10.8 fadeTo([speed] , opacity [callback])

10.9 animate(params [, speed] [, easing] [, callback])

params : 효과를 표현할 스타일 셋. map 형태로 지정할 수 있음.

easing : 플러그인을 지정하여 외부 효과를 지정할 때.

```
<!DOCTYPE HTML >
<html>
<head>
<title> animate </title>
<meta charset='utf-8'>
<style>
#show_div{ border:1px solid #800000;
width:200px; height:200px;
background-color:#c0c0c0;
}
.show_btn{ width:100px;height:30px}
</style>
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(function(){
  $('#ani_btn').click(function(){
    $('#show_div').animate({width:'toggle', opacity:'toggle'}, 2000);
  })
})
</script>
</head>

<body>
<button id='ani_btn' >animation</button>
<div id='show_div'></div>
</body>
</html>
```

11 Interactions

11.1 Tab Widget

```
<!doctype html>

<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Tabs</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css"
/>
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>

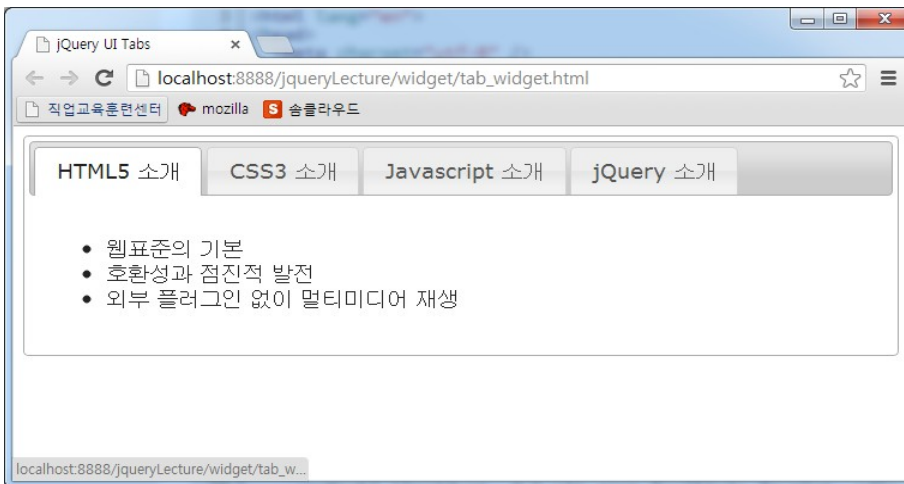
  <script>
$(function() {
  $( "#tabs" ).tabs();
});
</script>
</head>
<body>

<div id="tabs">
  <ul>
    <li><a href="#tabs-1">HTML5 소개</a></li>
    <li><a href="#tabs-2">CSS3 소개</a></li>
    <li><a href="#tabs-3">Javascript 소개</a></li>
    <li><a href="#tabs-4">jQuery 소개</a></li>

  </ul>
  <div id="tabs-1">
    <ul>
      <li>웹표준의 기본</li>
      <li>호환성과 점진적 발전</li>
      <li>외부 플러그인 없이 멀티미디어 재생</li>
    </ul>
  </div>
  <div id="tabs-2">
    html 이 할 수 없는 다양한 ui 표현.
  </div>
  <div id="tabs-3">
    <ul>
      <li>가장 저 평가된 언어중 하나.</li>
      <li>웹의 영역뿐만 아니라 애플리케이션에서도 활용할 수 있도록 개발진행</li>
      <li>서버 클라이언트 기능을 모두 스크트만으로 처리 될 수 있도록 개발진행</li>
    </ul>
  </div>
  <div id="tabs-4">
    현재 나... ^^
  </div>
</div>
```

11 Interactions > 11.1 Tab Widget

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



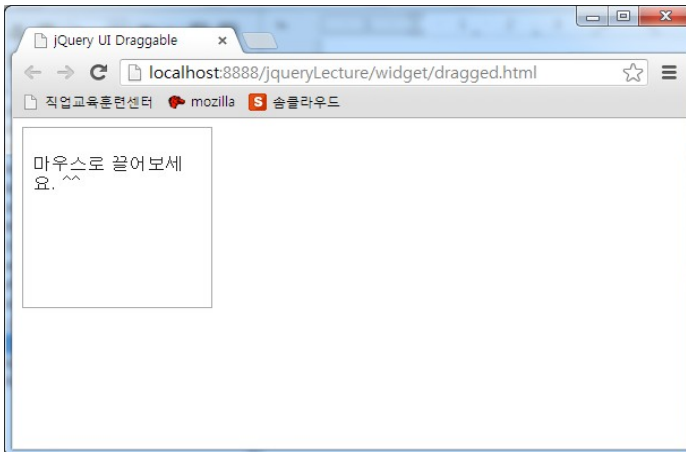
11.2 Dragged

```
<!doctype html>

<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Draggable</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <style>
    #draggable { width: 150px; height: 150px; padding: 0.5em; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable();
    });
  </script>
</head>
<body>

<div id="draggable" class="ui-widget-content">
  <p>마우스로 끌어보세요. ^^</p>
</div>

</body>
</html>
```



11.3 Droppable

```
<!doctype html>

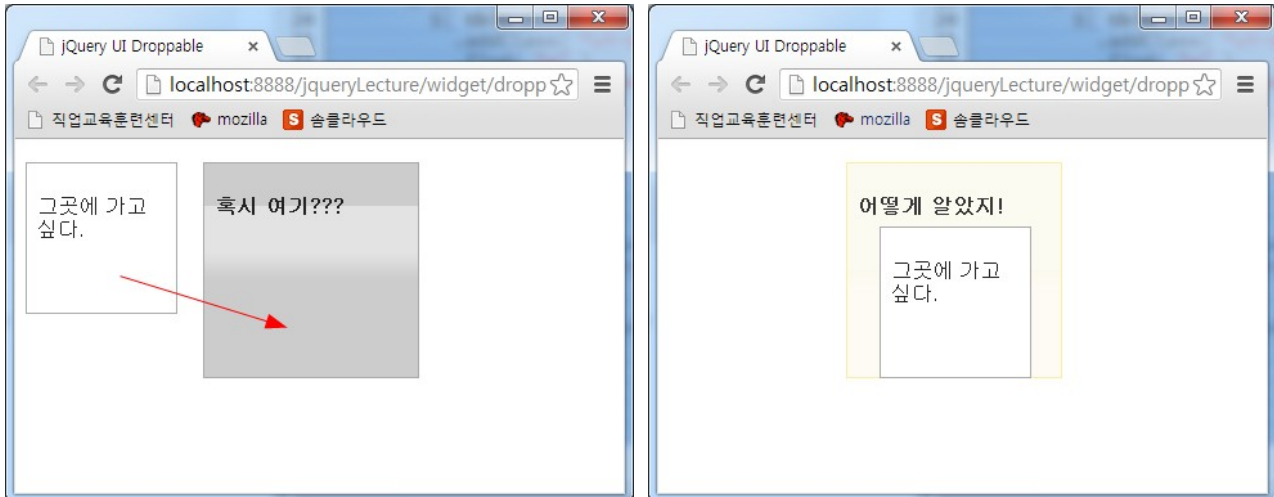
<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Droppable</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <style>
    #draggable { width: 100px; height: 100px; padding: 0.5em; float: left; margin: 10px 10px 10px 0; }
    #droppable { width: 150px; height: 150px; padding: 0.5em; float: left; margin: 10px; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable();
      $( "#droppable" ).droppable({
        drop: function( event, ui ) {
          $( this )
            .addClass( "ui-state-highlight" )
            .find( "p" )
              .html( "어떻게 알았지!" );
        }
      });
    });
  </script>
</head>
<body>

<div id="draggable" class="ui-widget-content">
  <p>그곳에 가고 싶다.</p>
</div>

<div id="droppable" class="ui-widget-header">
```

11 Interactions > 11.3 Droppable

```
<p>혹시 여기??</p>
</div>
</body>
</html>
```

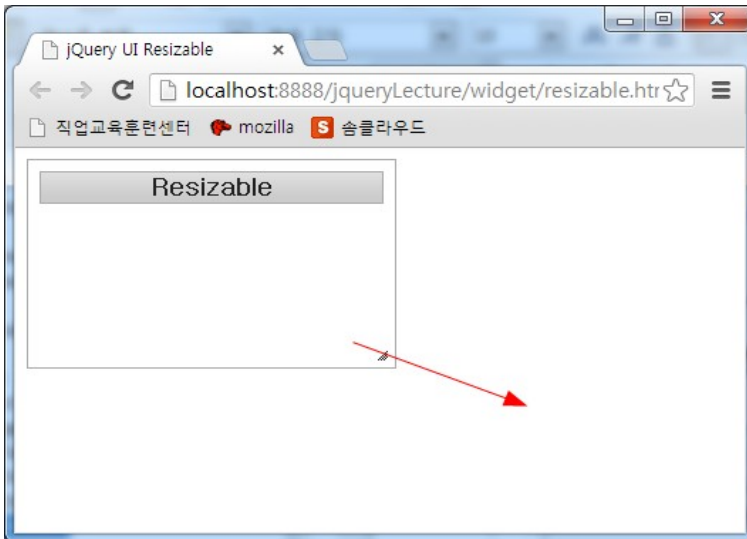


11.4 Resizable

```
<!doctype html>

<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Resizable</title>
  <link rel="stylesheet"
    href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <style>
    #resizable { width: 150px; height: 150px; padding: 0.5em; }
    #resizable h3 { text-align: center; margin: 0; }
  </style>
  <script>
    $(function() {
      $( "#resizable" ).resizable();
    });
  </script>
</head>
<body>

<div id="resizable" class="ui-widget-content">
  <h3 class="ui-widget-header">Resizable</h3>
</div>
</body>
</html>
```

11.5 Selectable

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Selectable</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />

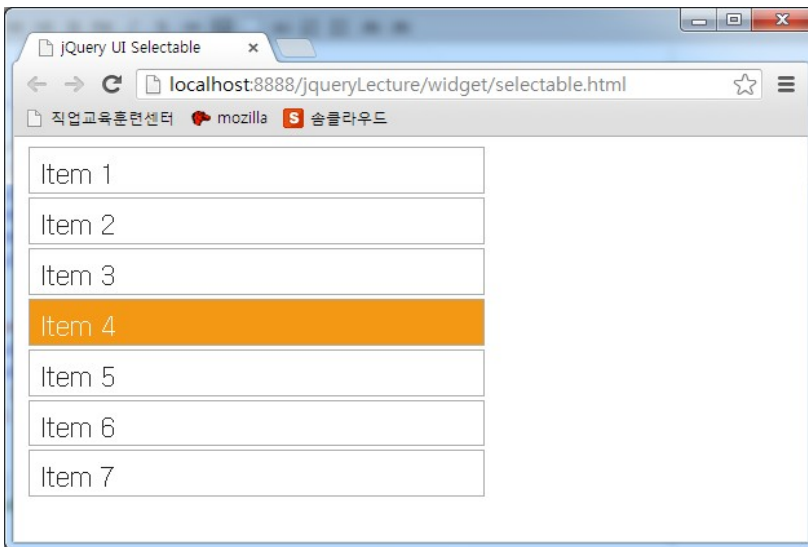
  <style>
    #feedback { font-size: 1.4em; }
    #selectable .ui-selecting { background: #FECA40; }
    #selectable .ui-selected { background: #F39814; color: white; }
    #selectable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
    #selectable li { margin: 3px; padding: 0.4em; font-size: 1.4em; height: 18px; }
  </style>
  <script>
    $(function() {
      $( "#selectable" ).selectable();
    });
  </script>
</head>
<body>

<ol id="selectable">
  <li class="ui-widget-content">Item 1</li>
  <li class="ui-widget-content">Item 2</li>
  <li class="ui-widget-content">Item 3</li>
  <li class="ui-widget-content">Item 4</li>
  <li class="ui-widget-content">Item 5</li>
  <li class="ui-widget-content">Item 6</li>
  <li class="ui-widget-content">Item 7</li>
</ol>
```

```

</body>
</html>

```



11.6 Sortable

```

<!doctype html>

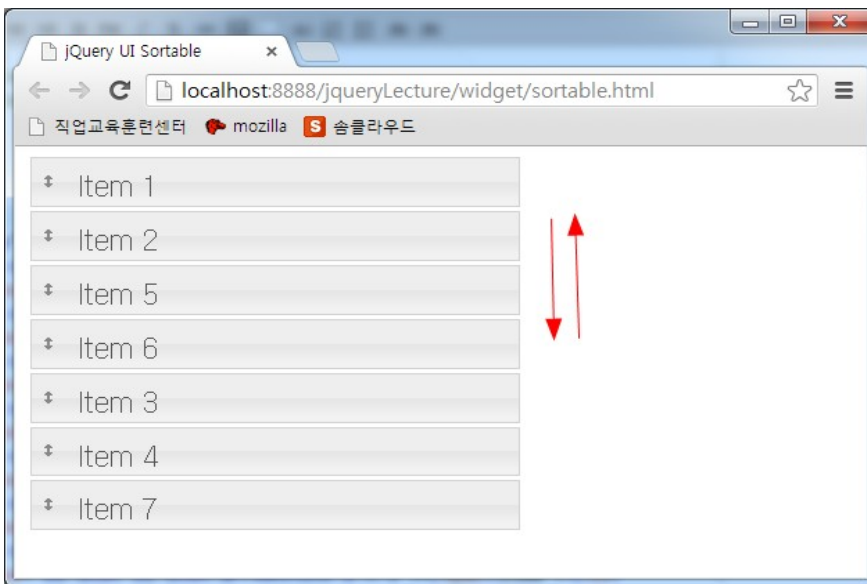
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Sortable</title>
  <link rel="stylesheet"
    href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <style>
    #sortable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
    #sortable li { margin: 0 3px 3px 3px; padding: 0.4em; padding-left: 1.5em;
      font-size: 1.4em; height: 18px; }
    #sortable li span { position: absolute; margin-left: -1.3em; }
  </style>
  <script>
    $(function() {
      $( "#sortable" ).sortable();
      $( "#sortable" ).disableSelection();
    });
  </script>
</head>
<body>

<ul id="sortable">
  <li class="ui-state-default">
    <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 1</li>
  <li class="ui-state-default">
    <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 2</li>

```

11 Interactions > 11.6 Sortable

```
<li class="ui-state-default">
  <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 3</li>
<li class="ui-state-default">
  <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 4</li>
<li class="ui-state-default">
  <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 5</li>
<li class="ui-state-default">
  <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 6</li>
<li class="ui-state-default">
  <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 7</li>
</ul>
</body>
</html>
```



12 Widgets

12.1 Accordion

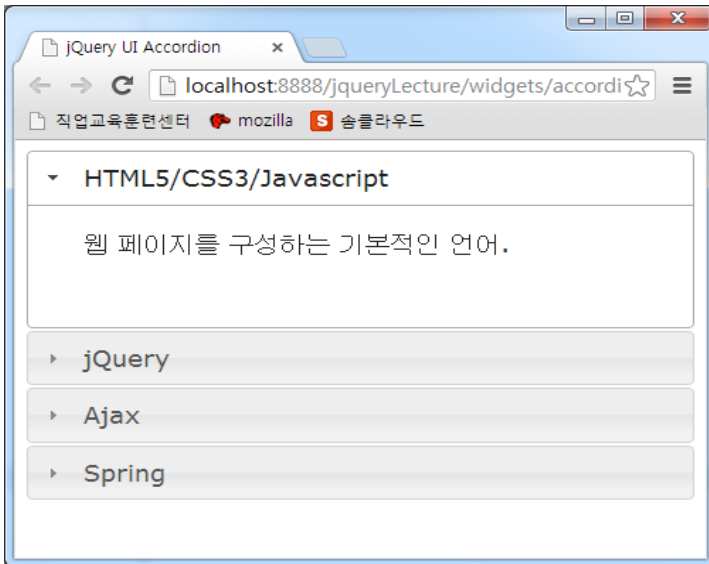
```
<!doctype html>

<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Accordion</title>
  <link rel="stylesheet"
    href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <script>
    $(function() {
      $( "#accordion" ).accordion();
    });
  </script>
</head>
<body>

<div id="accordion">
  <h3>HTML5/CSS3/Javascript</h3>
  <div>
    웹 페이지를 구성하는 기본적인 언어.

  </div>
  <h3>jQuery</h3>
  <div>
    <p>
      UI 의 크로스 브라우징의 가장 좋은 대안.
    </p>
  </div>
  <h3>Ajax</h3>
  <div>
    자바스크립트를 사용하여 동적인 페이지 삽입할수 있는 프레임워크.
  </div>
  <h3>Spring</h3>
  <div>
    유연성을 극대화 시킨 정부 표준 프레임워크중 하나.
  </div>
</div>
</body>
</html>
```

12 Widgets > 12.1 Accordion



12.2 Autocomplate

```
<!doctype html>

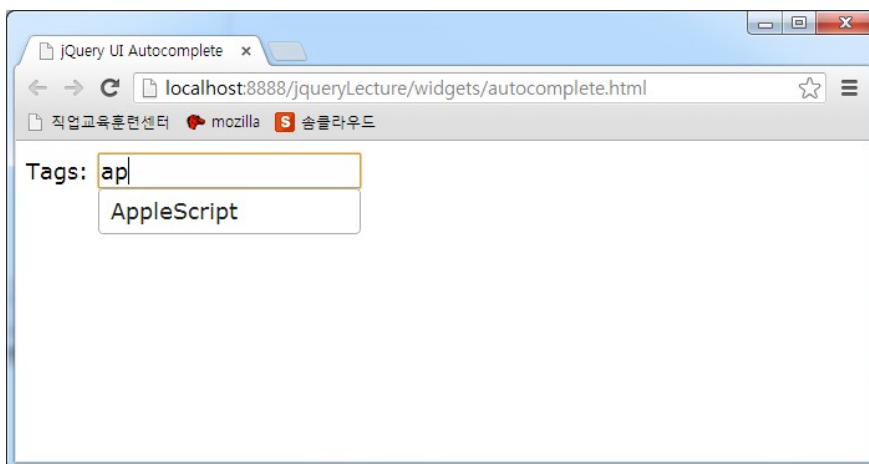
<html lang="ko">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Autocomplete</title>
  <link rel="stylesheet"
    href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <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>

```



12.3 Menu

```

<!doctype html>

<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>jQuery UI Menu</title>
  <link rel="stylesheet"
    href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css" />
  <script>
    $(function() {
      $( "#menu" ).menu();
    });
  </script>
</html>

```

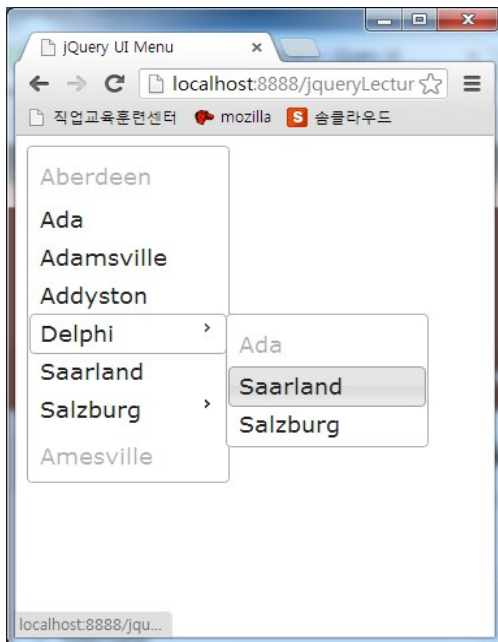
```

.ui-menu { width: 150px; }
</style>
</head>
<body>

<ul id="menu">
  <li class="ui-state-disabled">
    <a href="#">Aberdeen</a></li>
    <li><a href="#">Ada</a></li>
    <li><a href="#">Adamsville</a></li>
    <li><a href="#">Addyston</a></li>
  <li>
    <a href="#">Delphi</a>
    <ul>
      <li class="ui-state-disabled"><a href="#">Ada</a></li>
      <li><a href="#">Saarland</a></li>
      <li><a href="#">Salzburg</a></li>
    </ul>
  </li>
  <li><a href="#">Saarland</a></li>
  <li>
    <a href="#">Salzburg</a>
    <ul>
      <li>
        <a href="#">Delphi</a>
        <ul>
          <li><a href="#">Ada</a></li>
          <li><a href="#">Saarland</a></li>
          <li><a href="#">Salzburg</a></li>
        </ul>
      </li>
      <li>
        <a href="#">Delphi</a>
        <ul>
          <li><a href="#">Ada</a></li>
          <li><a href="#">Saarland</a></li>
          <li><a href="#">Salzburg</a></li>
        </ul>
      </li>
      <li><a href="#">Perch</a></li>
    </ul>
  </li>
  <li class="ui-state-disabled"><a href="#">Amesville</a></li>
</ul>
</body>
</html>

```

12 Widgets > 12.3 Menu



13 Ajax

13.1 메서드 종류

- Global Event Handler
 - ajaxStart() | ajaxSend() | ajaxSuccess()
 - ajaxComplete() | ajaxError() | ajaxStop()
- 데이터 전송 방법
 - \$.getScript() | \$.getJSON()
 - \$.get() | \$.post()
- load()
- serialize()
- evalScript()
- Trigger inside
 - \$.getIfModified()
 - \$.ajaxTimeout()
 - \$.ajaxSetup()
 - \$.ajax()

13.2 ajax process flow

- 요청 성공시 callback 순서
 - start -> send -> success -> complete -> stop
- 요청 실패시 callback 순서
 - start -> send -> error -> complete -> stop

13.3 \$.ajax()

- url : 요청할 url(필수 옵션)
- async(옵션) : 비동기적으로 호출할지에 대한 불리언.기본값 true(비동기방식)
- beforeSend(옵션) : 요청하기 전 실행할 콜백 함수
- cache(옵션) : 응답을 브라우저 캐시에 캐싱할지를 설정(기본값 :true)
- complete(옵션) : 요청이 끝난 후 실행할 콜백 함수
- contentType(옵션) : 요청 할때 포함할 MIME 콘텐츠 타입(기본값 : application/x-www-form-urlencoded)
- context(옵션) : 콜백함수 내부에서 this 키워드로사용할 객체. 보통은 DOM 엘리먼트를 사용함. (기본값은 window)
- data(옵션) : 서버에 보낼 문자열 값이나 자바스크립트 데이터 객체
- dataFilter(옵션) : Ajax 요청 성공시 콜백 함수를 실행하기 전에 서버에서 내려온 응답 값에 대해 먼저 처리할 함수
- type : 'POST' | 'GET'
- dataType : 서버 측에서 응답 받을 데이터의 형식을 문자열로 지정
 - 'html' | 'xml' | 'text' | 'json' | 'script' | 'jsonp'
- error(옵션) : 요청 실패 시 실행할 콜백 함수
- global(옵션) : 메서드가 실행될 때 불리언 값에 따라 전역 Ajax 함수의 실행 여부를 설정(기본

13 Ajax > 13.3 \$.ajax()

- 값:true)
- ifModified(옵션) : 서버에서 Ajax 요청을 보내고 받은 응답이 이전 요청의 응답 결과와 같은지 확인해야 할지를 나타내는 값(기본값 :false)
- jsonp(옵션) : JSONP를 이용해 Ajax 요청을 하고 나서 서버가 응답할 때 실행할 콜백함수를 지정.(기본값은 callback)
- username(옵션) : HTTP 인증을 위한 사용자 암호.
- password(옵션) : HTTP 인증 암호.
- processData(옵션) : 객체 형식 데이터를 쿼리 문자열 방식으로 변환할지 결정.(기본값:true)
- scriptCharset(옵션) : dataType 매개 변수의 값이 script 옵션이나 jsonp 형식일 때 사용할 데이터 코드셋.
- success(옵션) : 요청이 성공했을 때 실행할 콜백함수.
- timeout(옵션) : 요청 대기 시간. 단위는 밀리초
- type(옵션) : 요청에 사용할 GET 이나 POST 같은 HTTP 메서드 타입(기본값 : GET)
- xhr(옵션) : XMLHttpRequest 객체 생성시 사용되는 콜백함수. 기본값은 브라우저 사양에 따라 다르게 처리됨.
- callback
 - success:
 - error:
 - complete:

기본구조예

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script src='../lib/jquery-1.8.3.js'></script>
<script>
$(document).ready(function(){
    $.ajax({
        type:'POST',
        url : 'test.jsp',
        dataType:'html',
        success: function(html, status){
            $('#result').append(html);
        },
        error:function(xhr, status, error){
            alert(status);
        }
    });
})
</script>
<title>simple ajax code</title>
</head>
<body>
<h3>Simple Ajax Code<hr/></h3>
<div id='result'></div>
</body>
</html>
```

13.4 load(url [, params] [,callback])

- 서버로 부터 파일을 읽어 DOM에 적용되었을 때
- \$.get()과 다른점은 읽어들인 문서중 일부분만을 처리하고 나머지는 제거할 수 있다.
- 매개변수가 사용되면 POST 유형으로 전달됨.

[예] 읽어들인 모든 문서 처리 예

jquery
<pre>\$a = \$('#ajax_result'); \$a.load('test.jsp');</pre>
html
<pre><div id='ajax_result'></div></pre>
test.jsp
<pre><%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%> <% request.setCharacterEncoding("utf-8"); %> <!DOCTYPE html> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> </head> <body> <% request.setCharacterEncoding("utf-8"); %> 하이 방가... jobtc 입니다.
 load()에 의해 로딩된 자료입니다. <hr/> </body> </html></pre>



[예] 읽어들인 문서 내용중 일부분만을 처리

jQuery
<pre><script> \$(document).ready(function(){ \$a = \$('#ajax_result'); \$a.load('test.jsp #part');// test.jsp 문서를 읽어들이 id='part'인 부분만 처리 });</pre>

13 Ajax > 13.4 load(url [, params] [,callback])

```
})  
</script>  
</head>  
<body>  
<h3>Ajax load() Test</h3>  
<div id='ajax_result'></div>  
</body>
```

test.jsp 문서 내용

```
<body>  
<% request.setCharacterEncoding("utf-8"); %>  
하이 방가... jobtc 입니다.<br/>  
load()에 의해 로딩된 자료입니다.  
<hr/>  
<div id='part' class='part'>부분적으로 로딩될 부분 ^^</div>  
</body>
```



[예] 매개변수 전달예(매개변수는 POST 타입으로 전달됨)

jQuery

```
<script>  
$(document).ready(function(){  
    $a = $('#ajax_result');  
    $a.load('test2.jsp', 'message=박원기&pwd=1234');  
})  
</script>  
</head>  
<body>  
<h3>Ajax load() Test</h3>  
<div id='ajax_result'></div>  
</body>
```

test2.jsp 문서 내용

```
<body>  
<% request.setCharacterEncoding("utf-8");  
    String message = "";  
    String pwd="";  
    message = request.getParameter("message");  
    if(message == null) message = "메시지 없음";  
    pwd = request.getParameter("pwd");  
    if(pwd == null) pwd="none";  
%>  
하이 방가... jobtc 입니다.<br/>  
load()에 의해 로딩된 자료입니다.
```

13 Ajax > 13.4 load(url [, params] [,callback])

```
<hr/>
<li>Message : <%=message %>
<li>Pwd : <%=pwd %>
</body>
```



13.5 ajaxStart() | ajaxStop()

- ajaxStart(callback) : ajax 가 시작되었을 때
- ajaxStop(callback) : ajax 가 종료되었을 때

ajax

```
var str;
$('#result').ajaxStart(function(){
    str = '1.ajax start<br/>';
    $('#result2').append(str);
});
$('#result').ajaxStop(function(){
    str = '3.ajax stop<br/>';
    $('#result2').append(str);
});

$('#js_btn').click(function(){
    $('#result').load('test.jsp');
    str = "2.load<br/>";
    $('#result2').append(str);
});
```

html

```
<div id='result'></div>
<div id='result2'></div>
<button id='js_btn'>Javascript Test</button>
```

13.6 \$.get() | \$.post()

- HTTP 의 GET 메서드를 사용하여 서버에서 데이터를 불러온다.
- \$.get(url [, data] [, success] [,dataType])
- \$.post(url [, data] [, success] [,dataType])

get 기본예

13 Ajax > 13.6 \$.get() | \$.post()

```
<script>
$(document).ready(function(){
    $.get('test.jsp', function(data){
        $('#result').html(data);
    })
})
</script>

</head>
<body>
<div id='result'></div>
</body>
```

[예] 매개변수 사용예

jQuery

```
<script>
$(document).ready(function(){
    $.get('test2.jsp', 'message=jobtc&pwd=1234', function(data){
        $('#result').html(data);
    })
})
</script>

</head>
<body>
<div id='result'></div>
</body>
```

test2.jsp

```
<body>
<% request.setCharacterEncoding("utf-8");
String message = "";
String pwd="";
message = request.getParameter("message");
if(message == null) message = "메시지 없음";
pwd = request.getParameter("pwd");
if(pwd == null) pwd="none";
%>
하이 방가... jobtc 입니다.<br/>
load()에 의해 로딩된 자료입니다.
<hr/>
<li>Message : <%=message %>
<li>Pwd : <%=pwd %>
</body>
```

[예] post 메서드를 사용한 예(test2.jsp 파일은 위와 동일)

jQuery

```
<script>
$(document).ready(function(){
    $.post('test2.jsp', 'message=jobtc&pwd=1234', function(data){
        $('#result').html(data);
    })
})
```

13 Ajax > 13.6 \$.get() | \$.post()

```
})  
</script>  
  
</head>  
<body>  
<div id='result'></div>
```

13.7 \$.getScript()

외부에 정의된 자바스크립트 파일을 읽어 들인다.

외부에 정의된 자바스크립트 함수 a()를 읽어 들여 실행

jquery

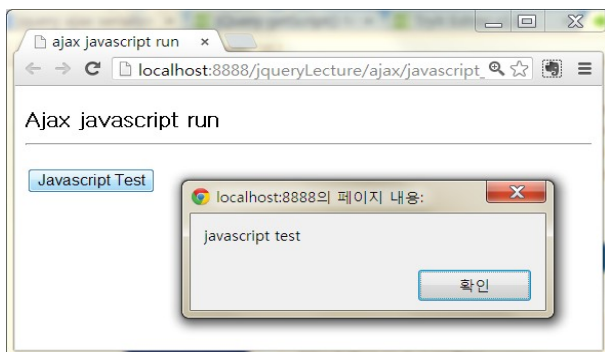
```
$.getScript('test.js');  
$('#result').text('버튼을 클릭하면 외부의 자바스크립트가 실행됨');  
$('#js_btn').click(function(){  
    a();  
});
```

html

```
<div id='result'></div>  
<button id='js_btn'>Javascript Test</button>
```

test.js

```
function a(){  
    alert('javascript test');  
}
```



13.8 \$.ajaxSetup()

- Ajax 요청에 대한 기본값을 정의 한다.
- 향후 \$.ajaxSetup()이 호출될 때까지 모든 호출에서 기본값으로 사용된다.

기본값 설정 예

```
$.ajaxSetup({  
    url : 'jobtc.jsp'  
});
```

13 Ajax > 13.8 \$.ajaxSetup()

14 parsing

jQuery 가 제공하는 파싱은 아래와 같다.

- `parseJSON()`
- `parseXML()`
- `parseHTML()`

jQuery 에서 데이터를 parsing 하는 방법은 크게 두가지 방법으로 사용될 수 있다.

문서 유형	dataType 미지정	dataType 지정
html	<code>\$.parseHTML(data)</code>	dataType : "html" 지정후 <code>\$.parseHTML(data)</code> 는 사용하지 않음
json	<code>\$.parseJSON(data)</code>	dataType : "json" 지정후 <code>\$.parseJSON(data)</code> 는 사용하지 않음
xml	<code>\$.parseXML(data)</code>	dataType: "xml" 지정후 <code>\$.parseXML(data)</code> 는 사용하지 않음.

즉, dataType 속성을 사용하게 되면 별도로 parsing 해 주지 않아야 된다.

14.1 parseJSON

기존에 사용하던 `JSON.parse()`가 있는데 이는 각 브라우저에 종속적이기 때문에 보안에 좋지만 환경에 독립적이지 못하기 때문에 범용성 측면에서는 그리 좋은 방법이 아닐 수 있다.

json.js

```
[
  { "mid": "a001", "irum": "hong1" },
  { "mid": "a002", "irum": "hong2" },
  { "mid": "a003", "irum": "hong3" },
  { "mid": "a004", "irum": "hong4" },
  { "mid": "a005", "irum": "hong5" }
]
```

parseJSON.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script src = '../Lib/jquery-3.2.1.min.js'></script>
<script>
```

```

$(document).ready(function(){
    $('#input').click(function(){
        $.ajax({
            type:"GET",
            url:"json2.js",
            dataType : "json",
            success:function(json){
                //var list = $.parseJSON(json);
                var listLen = list.length;
                var contentStr = "<ul>";
                for(var i=0; i<listLen; i++){
                    contentStr += "<li>" + list[i].mid+ "[ " + list[i].irum +
                ]</li>";
                }
                $("#before").append(json);
                $("#after").append(contentStr);
            }
        });
    });
});
</script>
</head>
<body>
<input type='button' id='input' value='parse Test'>
<h3>before</h3>
<div id='before'></div>

<h3>after</h3>
<div id='after'></div>

</body>
</html>

```

- dataType:'json'을 사용하던지, 데이터를 읽고나서 \$.parseJSON()을 사용한다.

14.2 jsp 에서 읽어 오기

jsp 에서 JSON 타입의 데이터를 생성할 경우 jsp 의 각종 지시문과 스트림트영역 및 JSON 영역을 제외하고 일반적인 html 태그는 모두 삭제되어야 한다.

MemberDao 와 MemberVo 객체는 사전에 존재한다고 가정한다.

json.jsp [json 타입의 데이터 생성 부분]

1. <%@page import="member.MemberVo"%>
2. <%@page import="member.MemberDao"%>
3. <%@ page language="java" contentType="text/html; charset=UTF-8"
4. pageEncoding="UTF-8"%>
5. <%
6. String mid = "";
7. if(request.getParameter("mid") != null){
8. mid = request.getParameter("mid");

```

9. }
10. MemberDao dao = new MemberDao();
11. MemberVo vo = dao.selectOne(mid);
12.
13. %>
14. {
15.   "mid" : "<%=vo.getMid() %>" ,
16.   "irum" : "<%=vo.getIrum() %>",
17.   "phone": "<%=vo.getPhone() %>"
18. }

```

생성된 json 타입의 데이터를 읽는 방법은 크게 두 가지로 읽을 수 있다.

parseJSON.jsp

```

1. <%@ page language="java" contentType="text/html; charset=UTF-8"
2.   pageEncoding="UTF-8"%>
3. <!DOCTYPE html>
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7. <title>parseJSON</title>
8. <script src = 'http://code.jquery.com/jquery-3.2.1.min.js' ></script>
9. <script>
10. $(document).ready(function(){
11.   $('#parseJSON #btnFind').click(function(){
12.     $param = $('#parseJSON #findStr').val();
13.
14.     $.ajax({
15.       type      : 'get',
16.       url       : './jqueryInAjax/json.jsp?mid=' + $param,
17.       success:function(obj){
18.
19.         $data = $.parseJSON(obj);
20.
21.         $rs = $('#parseJSON #result');
22.         $rs.empty();
23.         $rs.append('<li>Mid : ' + $data.mid);
24.         $rs.append('<li>Irum : ' + $data.irum);
25.         $rs.append('<li>Phone : ' + $data.phone);
26.
27.         /* 두번째 방법
28.         type      : 'get',
29.         dataType : 'json',
30.         url       : 'json.jsp?mid=' + $param,
31.         success:function(obj){
32.
33.         $rs = $('#parseJSON #result');
34.         $rs.empty();

```

```

35.     $rs.append('<li>Mid : ' +obj.mid);
36.     $rs.append('<li>Irum : ' + obj.irum);
37.     $rs.append('<li>Phone : ' + obj.phone);
38.     */
39.
40.
41. },
42. error : function(xhr, status, error){
43.     alert(error);
44. }
45.
46. });
47. });
48. })
49.
50.
51. </script>
52. </head>
53. <body>
54. <div id='parseJSON'>
55.   <h3>parseJSON</h3>
56.   <input type='text' id='findStr' />
57.   <input type='button' id='btnFind' value='검색'>
58.   <hr/>
59.   <div id='before'></div>
60.   <div id='result'></div>
61.
62. </div>
63. </body>
64. </html>

```

14.3 parseXML

xmlData.xml

```

<?xml version='1.0' encoding='utf-8'?>
<members>
  <member>
    <name>홍길동</name>
    <addr>서울</addr>
    <phone>010-1111-1111</phone>
  </member>
  <member>
    <name>이길동</name>
    <addr>부산</addr>
    <phone>010-1111-2222</phone>

```

```

</member>
<member>
  <name>성춘향</name>
  <addr>전라도</addr>
  <phone>010-1111-3333</phone>
</member>
<member>
  <name>임걱정</name>
  <addr>제주</addr>
  <phone>010-1111-4444</phone>
</member>
<member>
  <name>이몽룡</name>
  <addr>대전</addr>
  <phone>010-1111-5555</phone>
</member>
</members>

```

parseXML.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script src = '../Lib/jquery-3.2.1.min.js'></script>

<script>
$(document).ready(function(){
    $('#input').click(function(){
        $.ajax({
            type: "GET"
            ,dataType: "xml"
            ,url: "xmlData.xml"
            ,success: function(xml){
                var xmlData = $(xml).find("member");
                var listLength = xmlData.length;
                if (listLength){
                    var contentStr = "";
                    $(xmlData).each(function(){
                        contentStr += $(this).find("name").text() + " / "
                        + $(this).find("addr").text() + "("
                        + $(this).find("phone").text() + "</br>";
                    });
                    $("#after").append(contentStr);
                }
            }
        });
    });
});
</script>
</head>

```

```

<body>
<input type='button' id='input' value='parse Test'>
<h3>after</h3>
<div id='after'></div>
</body>
</html>

```

14.4 jsp 에서 XML 읽어오기

jsp 에서 XML 문서를 만들려면 <?xml ... ?> 태그와 jsp 지시문의 contentType='text/xml'이 필요하다.

xml.jsp

```

1. <?xml version='1.0' encoding='utf-8' ?>
2. <%@page import="member.MemberVo"%>
3. <%@page import="member.MemberDao"%>
4. <%@ page language="java" contentType="text/xml; charset=UTF-8"
5.     pageEncoding="UTF-8"%>
6. <%
7. String mid = "a666";
8. if(request.getParameter("mid") != null){
9.     mid = request.getParameter("mid");
10. }
11. MemberDao dao = new MemberDao();
12. MemberVo vo = dao.selectOne(mid);
13.
14. %>
15. <members>
16. <member>
17.     <mid><%=vo.getMid() %></mid>
18.     <irum><%=vo.getIrum() %></irum>
19.     <phone><%=vo.getPhone() %></phone>
20. </member>
21. </members>

```

xml 타입의 데이터를 파싱하는 방법은 크게 두가지로 사용될 수 있다.

parseXML.jsp

```

1. <%@ page language="java" contentType="text/html; charset=UTF-8"
2.     pageEncoding="UTF-8"%>
3. <!DOCTYPE html>
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7. <title>parseXML</title>
8. <script src = 'http://code.jquery.com/jquery-3.2.1.min.js' ></script>

```

```

9. <script>
10. $(document).ready(function(){
11. $('#parseXML #btn').click(function(){
12.     $rs = $("#result");
13.     $findStr = ($("#parseXML #findStr").val());
14.
15.     // 첫번째 방법
16.     $.ajax({
17.         type      : 'get',
18.         url       : './jqueryInAjax/xml.jsp?mid='+$findStr,
19.         success   : function(obj){
20.             $obj = $.parseXML(obj);
21.             $data = $(obj).find("member");
22.             $data.each(function(){
23.                 $str = '<ul>';
24.                 $str += '<li>mid : ' + $(this).find('mid').text() + '<\li>';
25.                 $str += '<li>irum : ' + $(this).find('irum').text() + '<\li>';
26.                 $str += '<li>phone : ' + $(this).find('phone').text() + '<\li>';
27.                 $str += '<\ul>';
28.             });
29.
30.             $rs.html($str);
31.         }
32.
33.         /* 두번째 방법
34.         type      : 'get',
35.         url       : './jqueryInAjax/xml.jsp?mid='+$findStr,
36.         dataType: 'xml',
37.         success   : function(obj){
38.             $data = $(obj).find("member");
39.             $data.each(function(){
40.                 $str = '<ul>';
41.                 $str += '<li>mid : ' + $(this).find('mid').text() + '<\li>';
42.                 $str += '<li>irum : ' + $(this).find('irum').text() + '<\li>';
43.                 $str += '<li>phone : ' + $(this).find('phone').text() + '<\li>';
44.                 $str += '<\ul>';
45.             });
46.
47.             $rs.html($str);
48.         }
49.         */
50.     });
51. });
52. })
53. </script>
54. </head>
55. <body>
56. <div id='parseXML'>
57. <h3>parseXML</h3>
58. <input type='text' id='findStr' />

```

```
59. <input type='button' value='parseXML' id='btn'/><hr/>
60. <div id='result'></div>
61. </div>
62. </body>
63. </html>
```

14.5 parseHTML

htmlData.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<font color='#0000ff' size='7'>Hi HTML parser...</font>

<div id='part'>이부분은 HTML의 부분 영역입니다....</div>
</body>
</html>
```

parseHTML.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script src = '../Lib/jquery-3.2.1.min.js'></script>
<script>
$(document).ready(function(){
    $('#input').click(function(){
        $.ajax({
            type:"GET",
            url:"htmlData.html",
            success:function(html){
                var list = $.parseHTML(html);
                $("#before").text(html);
                $("#after").html(list);
            }
        });
        //일 부부만 처리
        $('#part').load('htmlData.html #part');
    });
});
</script>
```


14 parsing > 14.5 parseHTML

```
</head>
<body>
<input type='button' id='input' value='parse Test'>
<h3>before</h3>
<div id='before'></div>

<h3>after</h3>
<div id='after'></div>

<h3>part</h3>
<div id='part'></div>

</body>
</html>
```

15 응용예

15.1 소스 코드 읽기

개발된 소스를 실행화면과 함께 보여주기 위한 방법이다.

먼저 간단하게 html 문서를 작성한다. 코드를 보여줄 버튼과 감출버튼. 그리고, 읽은 코드가 표시될 위치이다.

8 행의 jQuery 라이브러리는 독자의 경로를 선택하던, CDN 를 사용한 URL 을 사용해도 무방하다.

15.1.1 HTML

```

1. <%@ page language="java" contentType="text/html; charset=UTF-8"
2.     pageEncoding="UTF-8"%>
3. <!DOCTYPE html>
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7. <title>Insert title here</title>
8. <script src='/Lecture_jquery/lib/jquery-1.8.3.js'></script>
9. <style>
10.
11. </style>
12. <script>
13.
14. </script>
15. </head>
16. <body>
17. <h3>소스보기</h3>
18. <input type='button' value='code view' id='code_view' />
19. <input type='button' value='code hide' id='code_hide' />
20. <div id='view'></div>
21. <div id='code'></div>
22.
23. </body>
24. </html>

```

15.1.2 CSS

필요한 내용만 간단하게 스타일을 지정했다.

```

<style>
#code {
    display: none;
}
#view{

```

```

    font-size: 9pt;
}
#code ol{
    tab-size: 2;
    font-size: 9pt;
    font-family: monospace;
}
</style>

```

15.1.3 jQuery

```

1. <script>
2. $(document).ready(function(){
3.     var code = $('#code');
4.     var page = "test.jsp";
5.     $('#code_view').click(function(){
6.         code.show(); // 소스코드 위치 보여주기
7.         code.empty(); //기존 내용 모두 삭제
8.
9.         $.get(page, function(data) {
10.             $('#view').html(data); //실행결과 보기
11.             var str = "<h3>Source Code</h3><hr/><ol>";
12.
13.             var lines = data.split('\n');
14.             lines.splice(0,1); //첫번째 배열값 삭제
15.
16.             $.each(lines, function(i, elt) {
17.                 //문자열 파싱(html tag -> text type)
18.                 elt = elt.replace("<", "&lt;");
19.                 elt = elt.replace("</", "&lt;/");
20.                 elt = elt.replace(">", "&gt;");
21.                 elt = elt.replace(" ", "&nbsp;");
22.                 elt = elt.replace("\t", "&emsp;");
23.                 str += "<li>" + elt;
24.             });
25.             code.append(str + '</ol>');
26.         })
27.     });
28.
29.     $('#code_hide').click(function() {
30.         code.empty();
31.         code.hide();
32.     })
33.
34. });
35. </script>

```

15.1.4 테스트 파일

어떤 형태의 텍스트 파일이든 상관없다.

test.jsp

```

1. <%@ page language="java" contentType="text/html; charset=UTF-8"
2.     pageEncoding="UTF-8"%>
3. <!DOCTYPE html>
4. <html>
5. <head>
6. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7. <title>Insert title here</title>
8. </head>
9. <body>
10. <h3>page test.</h3>
11. <ol>
12. <li>Java</li>
13. <li>JavaScript</li>
14. <li>CSS</li>
15. <li>HTML5</li>
16. <li>jQuery</li>
17. <li>Ajax</li>
18. </ol>
19. </body>
20. </html>

```

15.1.5 실행결과

[code view] 버튼을 클릭한 후 화면이다. [code hide] 버튼이 클릭되면 읽혀진 내용이 모두 사라지고 해당 영역도 감추어 진다.

소스보기

code view
code hide

page test.

- Java
- JavaScript
- CSS
- HTML5
- jQuery
- Ajax

Source Code

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5. <title>Insert title here</title>
6. </head>
7. <body>
8. <h3>page test.</h3>
9. <ol>
10. <li>Java</li>
11. <li>JavaScript</li>
12. <li>CSS</li>
13. <li>HTML5</li>
14. <li>jQuery</li>
15. <li>Ajax</li>
16. </ol>
17. </body>
18. </html>

```

