



jQuery 메소드

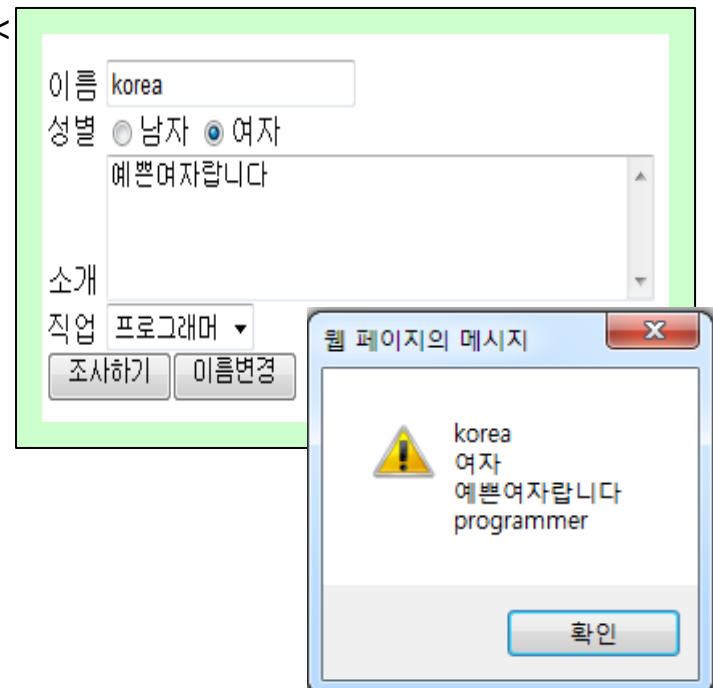
조작관련(Manipulation) 메소드들

◆ 내용 확인 및 변경 메소드 들

메소드	설 명
html()	일치된 요소의 html 내용을 가져온다 . innerHTML 기능과 동일하다 .일치된 요소가 여러 개 라면 그 중 첫 번째 요소의 html 내용만 가져온다
html(code)	일치된 요소의 본문을 html 내용으로 변경한다 일치된 요소가 여러 개 라면 모든 요소에 적용된다
text()	일치된 모드 요소를 내용을 가져온다 내용 중에 Html 코드가 있다면 html 코드는 제외한다
text(str)	일치된 모든 요소의 내용을 str 로 변경한다
val()	해당 입력 요소의 value 속성값을 가져온다
val(data)	해당 입력 요소의 value 속성값을 data 로 변경한다

예제(val)

```
<body> <form>
  <label for="user">이름 </label>
  <input type="text" id="user" value="korea"> <br>
  <label >성별 </label>
  <input type="radio" id="gend" name="gend" value="남자" checked >남자
  <input type="radio" id="gend" name="gend" value="여자" >여자 <br>
  <label for="intro">소개 </label>
  <textarea id="intro" cols="40" rows="4"> </textarea>
  <label for="work">직업 </label>
  <select id="work">
    <option value="programmer">프로그래머 </option>
    <option value="progammer">프로게이머 </option>
    <option value="whitehand">백수 </option>
  </select> <br>
  <button id="btnview">조사하기 </button>
  <button id="btnchange">이름변경 </button>
</form> </body>
```



조작관련(Manipulation) 메소드들

◆ 선택 요소내부에 추가하는 메소드 들

메소드	설 명
append(content)	일치된 요소 내부의 마지막 위치에 content를 추가한다
appendTo(selector)	선택된 요소를 selector에 일치된 모든 요소들의 내부 마지막 위치에 추가한다. 만일, 일치된 요소가 본문에 존재하면 그 요소를 제거한 후 복사한다. (즉, 이동)
prepend(content)	append(content)와 동일, 다만, 내부의 처음위치에 추가한다.
prependTo(selector)	appendTo(selector)와 동일, 다만, 내부의 처음위치에 추가한다.

조작관련(Manipulation) 메소드들

◆ 선택 요소 외부 에 추가하는 메소드 들

메소드	설 명
<code>after(content)</code>	일치된 요소 뒤에 content를 삽입한다. 요소 내부가 아닌 외부에 삽입된다.
<code>insertAfter(selector)</code>	선택된 요소를 selector에 의해 일치된 모든 요소들 뒤쪽에 삽입한다. 요소 내부가 아닌 외부에 삽입된다.
<code>before(content)</code>	일치된 요소 앞에 content를 삽입한다. 요소 내부가 아닌 외부에 삽입된다.
<code>insertBefore(selector)</code>	<code>insertAfter(selector)</code> 와 유사하나, 요소 앞쪽에 삽입한 다. 요소 내부가 아닌 외부에 삽입된다.

예제

```
<script>
  $(function(){
    //p를 클릭하면 그림을 추가
    $('p').click(function() {    });
  });
</script>
<body>
  <p title="Tulips">튤립</p>
  <p title="Chrysanthemum">국화</p>
  <p title="Koala">코알라</p>
  <p title="Hydrangeas">수국    </p>
</body>
```



조작관련 (삭제 복사) 메소드

◆ 삭제 메소드

메소드	설 명
<code>empty()</code>	선택된 모든요소의 자식요소를 지운다
<code>remove</code> <code>remove(selector)</code>	선택된 모든 요소와 그의 자식요소를 지운다 <code>\$('p').remove('.test')</code> <p></p> <p class='test'></p>

◆ 복사 메소드

메소드	설 명
<code>clone()</code>	일치된 요소를 복사하고 선택한다
<code>clone(true)</code>	일치된 요소의 이벤트 처리기를 포함하여 복사하고 선택한다

필터링 메소드

메소드	설 명
<code>eq(index)</code>	선택요소 중 <code>index</code> 번째의 요소
<code>first()</code>	선택요소 중 가장 첫 번째 요소
<code>last()</code>	선택요소 중 가장 마지막 번째 요소
<code>even()</code>	선택요소 중 짝수번째
<code>odd()</code>	선택요소 중 홀수번째
<code>filter(selector)</code>	선택요소 중 <code>Selector</code> 와 일치하는 요소를 필터링한다 <code>filter(':gt(1)')</code> <code>filter(':even')</code> <code>filter(':first')</code>
<code>is(selectorElement)</code>	선택요소가 <code>selectorElement</code> 와 일치하는지 판단 <code>true/ false</code> 를 리턴

찾기탐색 메소드

메소드	설 명
add()	선택확장
next() nextAll() nextUntil(selector)	선택요소의 다음 형제요소들
prev() preveAll() prevUntil(selector)	선택요소의 이전 형제요소들
parent() pratents() parents(selector) parentsUntil(selector)	선택요소의 부모요소
siblings() siblings(selector)	선택요소의 앞뒤 모든 요소들
find()	선택요소의 후손요소들

찾기메소드(parent-is)

- ◆ div로 감싸는 p태그와 그렇지 않은 p태그를 이용하여 문자열을 작성한다
- ◆ p태그를 클릭하면 부모가 div인지 판단하여 맞다면 p태그를 복사하여
- ◆ 글자색 빨강색으로 설정하여 부모에 붙여넣는다

무궁화 꽃이 피었습니다

은주는 예쁘다~

사뿐이 즈려뺏고 가시옵소서

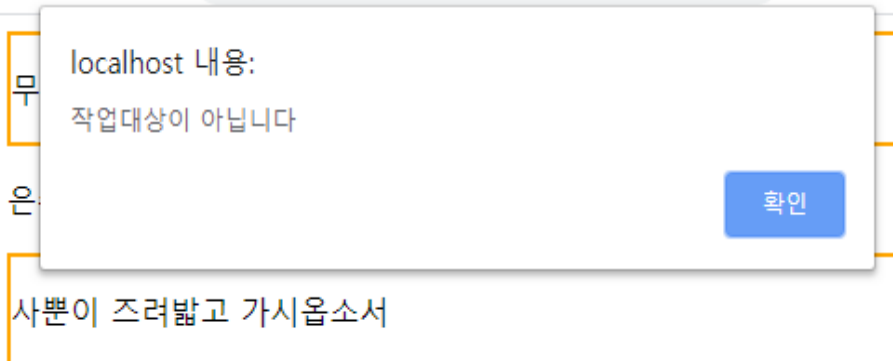
무궁화 꽃이 피었습니다

무궁화 꽃이 피었습니다

은주는 예쁘다~

사뿐이 즈려뺏고 가시옵소서

사뿐이 즈려뺏고 가시옵소서



스타일시트(css)관련 메소드

메소드	설 명
<code>css(name)</code>	매치되는 첫번째 요소의 스타일의 속성을 반환한다 예) <code>\$(this).css('color')</code>
<code>css(name, value)</code>	매치되는 모든 요소들의 단일 스타일 속성을 설정한다 예) <code>\$(this).css('color' , 'red')</code>
<code>css(propertis)</code>	매치되는 모든 요소들의 스타일 속성에 키:값 을 설정한다 예) <code>\$(this.).css({'color':'blue', 'font-size':'20px' })</code>

스타일시트 (css)관련 메소드

메소드	설 명
<code>addClass(class)</code>	매치된 요소들의 집합에 지정된 css 클래스를 추가한다
<code>hasClass(class)</code>	지정된 클래스가 매치된 요소들의 집합 중 최소 한군데 이상 적용 되어 있으면 true 를 리턴 한다
<code>removeClass(class)</code>	매치된 요소들의 집합에서 지정된 css 클래스를 삭제한다
<code>toggleClass(class)</code>	매치된 요소들에 지정된 클래스가 적용되지 않았다면 적용하고 , 이미 적용되어 있다면 제거한다

속성(attribute) 관련 메소드

메소드	설 명
<code>attr(name)</code>	매치된 첫번째 요소의 name 에 지정된 속성의 값을 가져온다. 지정된 속성 명이 존재하지 않는다면 undefined 가 반환된다
<code>attr(properties)</code>	매치되는 모든 요소들의 속성을 키:값 의 형태로 지정한다
<code>attr(key, value)</code>	매치되는 모든 요소들의 속성을 단일 값으로 지정한다
<code>attr(key, fn)</code>	매치되는 모든 요소들의 단일속성에 대한 fn 에서 수행된 값을 지정한다
<code>removeAttr(name)</code>	매치된요소의 속성을 제거한다

속성 메소드

- ◆ attr()메소드를 이용하여 이미지의 이름으로 title속성을 설정 한다

이미지의 src속성값 가져오기

src속성값에서 이름부분 추출하기

추출된 이름으로 title속성 부여하기



눌러보세요

prop() -속성상태 설정 및 상태 얻기

◆ element가 가지는 실제적인 상태(활성화, 체크 선택여부)등을 제어
(예: checked, selected, disabled, readonly, multiple)

◆ 가져올때 true 또는 false 를 리턴하고

◆ 설정시 true 또는 false 로 설정한다

◆ 상태 설정(set)

`$(selector).prop("속성명", 값);`

==> 값이 true이면 해당 속성을 설정하는 것이고
값이 false이면 해당 속성을 해제하는 것이다.

◆ 상태 얻기(get)

`$(selector).prop("속성명");`

==> 해당 속성이 설정되어 있으면 true,
설정되어 있지 않으면 false를 반환한다.

prop() 속성상태 설정 및 상태 얻기

<body>

<form>

체크박스(라디오버튼) : <input type="checkbox" id="checkTest" checked>

리스트박스(select객체) :

<select id="selTest">

<option value="1">하나</option>

<option value="2">둘</option>

<option value="3" selected >셋</option>

<option value="4">넷</option>

</select>

text객체 (readonly) : <input type="text" value="가나다" id="txtTest">

button객체(disabled) :

<input type="button" value="실행" id="runBtn" >

</form>

</body>

prop() 속성상태 설정 및 상태 얻기

- ◆ 실행버튼 클릭할때마다 각 상태값을 선택과 해제(true/false)로 번갈아가면서 변경

체크박스(라디오버튼) : ☒

리스트박스(select객체) :

text객체 (readonly) :

button객체(disabled) :

전체선택의 상태 값으로 1~5 의 상태값을 설정

체크박스(전체선택/해제) :
☒ 전체선택 ☒ 1 ☒ 2 ☒ 3 ☒ 4 ☒ 5

이벤트관련 메소드

메소드	설 명
bind(type, data, fn, map)	매치된 요소에 이벤트처리를 바인딩한다 type : 이벤트종류, data: fn의 파라미터값, Version3.0 deprecated ,Use the on() method map: {event:function, event:function, ...}
unbind(type, fn)	매치된 요소에 이벤트 처리기를 제거한다 Version3.0 deprecated Use the off() method instead
one(type, data, fn)	bind()와 같지만 이벤트를 한번만 실행하고 자동으로 이벤트를 제거한다
----map 표현---- \$("p").on({ mouseover: function(){ \$(this).css("background-color", "lightgray"); }, mouseout: function(){ \$(this).css("background-color", "lightblue"); }, click: function() { \$(this).css("background-color", "yellow"); } });	

이벤트관련 메소드

메소드	설 명
<code>delegate(selector, type, data, fn)</code>	매치된 요소에 이벤트처리를 바인딩한다 동적으로 작성된 새로운 요소에서도 이벤트실행 type : 이벤트종류, data:fn 의 파라미터값, Version3.0 deprecated Use the <code>on()</code> method instead
<code>undegate(selector, type, data, fn)</code>	매치된 요소에 이벤트 처리기를 제거한다 Version3.0 deprecated Use the <code>off()</code> method instead
<code>trigger(type)</code>	매치된 요소에 대하여 이벤트 타입에 해당하는 이벤트 처리기를 모두 실행한다 type 에 사용자 정의 이벤트 가 올 수 있다
<pre>\$("#button").click(function(){ \$(this).background("blue") \$("#p").trigger("click"); }); \$("#p").on("click", function(){ \$(this).text("hello~~~"); });</pre>	

이벤트관련 메소드

메소드	설 명
on(type, data, fn, map)	매치된 요소에 이벤트처리기를 바인딩한다 bind방식
on(type, selector, data, fn, map)	delegate 방식 동적으로 작성된 새로운 요소 map: {event:function, event:function, ...}
off(event, selector)	매치된 요소에 이벤트처리기를 제거한다

----map 표현----

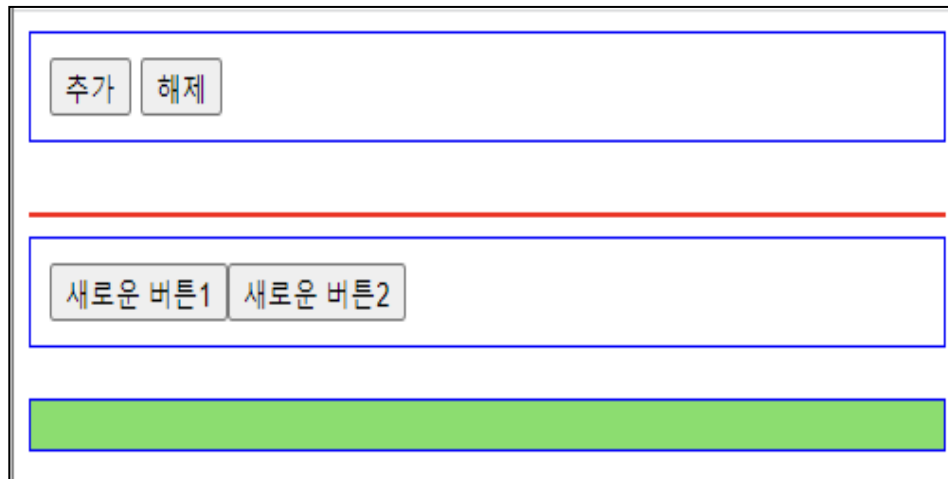
```
$("#p").on({  
  mouseover: function(){ $(this).css("background-color", "lightgray"); },  
  mouseout: function(){ $(this).css("background-color", "lightblue"); },  
  click: function() { $(this).css("background-color", "yellow"); }  
});
```

delegate

- ◆ bind()메소드는 선택자로 검색한 모든 엘리먼트에 핸들러를 등록한다
- ◆ \$('h1')의 검색 결과가 몇개이든가에 상관없이 문서내의 모든 <h1>엘리먼트에 똑같은 핸들러가 등록 된다
- ◆ 그러나 새로 추가되는 엘리먼트에는 핸들러가 지정되지 않아 이벤트가 발생해도 핸들러가 호출되지 않는다 .
- ◆ 미래에 추가될 엘리먼트에 대해서도 핸들러를 미리 등록하는 메서드가 필요 - \$(document).delegate(선택자, 이벤트명, 핸들러);
- ◆ 특정 엘리먼트에 대해 핸들러를 지금 당장 등록하는 것이 아니라
- ◆ 문서가 변경될때마다 핸들러가 등록되어야 하므로 document수준에서 메소드를 호출한다 (document 는 이벤트 대상객체의 부모가 될 수 있다)
- ◆ 이처럼 실행중에 자동으로 등록되는 살아있는 핸들러를 라이브 핸들러 라고 한다 .
- ◆ version 3.0에서 모든 이벤트 등록 메소드를 통합하는 메소드가 새로 도입 : on(이벤트명, [선택자], 핸들러) off(이벤트명, [선택자], 핸들러)
- ◆ 선택자 가 있으면 delegate와 같고 없으면 bind와 같다

delegate undelegate

- ◆ 추가버튼 클릭시 새로운 버튼을 생성하고 아래요소의 배경색을 랜덤으로 변경
- ◆ 해제버튼 실행
- ◆ bind기능과 delegate기능으로 비교 실행



◆ 추가 버튼 클릭할때마다

이름	<input type="text"/>
파일	<input type="button" value="파일 선택"/> 선택된 파일 없음
	<input type="button" value="추가"/>

◆ 파일첨부와 삭제버튼을 생성한다

◆ 삭제버튼 클릭시 해당라인의 파일첨부를 삭제

이름	<input type="text"/>
파일	<input type="button" value="파일 선택"/> 선택된 파일 없음
	<input type="button" value="추가"/>
파일 :	<input type="button" value="파일 선택"/> 선택된 파일 없음
	<input type="button" value="삭제"/>
파일 :	<input type="button" value="파일 선택"/> 선택된 파일 없음
	<input type="button" value="삭제"/>
파일 :	<input type="button" value="파일 선택"/> 선택된 파일 없음
	<input type="button" value="삭제"/>