
jQuery
write less, do more.

Manipulation

jQuery - 문서 조작

jQuery_framework
자바스크립트 라이브러리

jQuery(HTML) 팩토리 함수의 재발견!

jQuery() 팩토리 함수는 문서의 요소를 선택하는 용도 이외에도
요소를 만들어 내는 능력이 있습니다.

```
jQuery('<p>새롭게 생성될 내용</p>')
jQuery('<div />')
```

$(HTML) 팩토리 함수

```
$('<div> aside content </div>').addClass('posAbs top_bottom0');
```

$(HTML) 팩토리 함수

```
$('<input type="button" value="toggle" id="toggleButton">').insertAfter('#disclaimer');
$('#toggleButton').click(function() { $('#disclaimer').toggle(); });
```

1.4

$(HTML, { attr }) 팩토리 함수

```
$('<input/>', {
    type: 'button',
    value: 'toggle',
    id: 'toggle_button',
    css: {
        background: '#c00',
        color: '#eee'
    },
    click: function() {
        $(this).css('color', '#212121')
    }
}).appendTo('body');
```

Changing

jQuery - 요소 변경하기

.text([text]) 함수

text() 함수는 해당 요소의 텍스트노드를 가져오거나, 새로운 텍스트노드로 설정하는 제이쿼리 객체의 메소드로 html <Tag>를 포함하지 않습니다.

```
jQuery().text('text');
```

'jQuery() 래퍼객체의 텍스트노드 내용을 text로 변경하라.'

.text() 함수

```
$('#content>p:first').text();
```

.text(text) 함수

```
$('span:first').text('text');
```

1.4

.html([HTML | function(index, oldHTML)]) 함수

html() 함수는 ()안의 요소의 내용을 가져오거나, 새로운 내용으로 설정하는
제이쿼리 객체의 메소드로 html <Tag>를 포함하는 것이 text() 와의 차이점입니다.

```
jQuery().html('<element> text </element>');
```

'jQuery() 제이쿼리 래퍼객체의 HTML 내용을 <element> text </element>로 변경하라.'

.html() 함수

```
$('#content>p:first').html();
```

.html('text') 함수

```
$('#content>p:first').html('text');
```

.html('<element> text </element>') 함수

```
$('#header p').html('<element> text </element>');
```

1.4

.html(function(index, oldHTML)) 함수

```
$('li').filter(function(i) {
    return (i%3) == 0;
}).html(
    function(i, html) {
        return i + '번째 문장 내용 변경이 ' + html + '에서 시작되었습니다.';
    }).css('color', 'red');
```

.attr(attribute, [value, { map }, function(index, attr)]) 함수

attr() 함수는 요소의 속성 값을 가져오거나, 설정하는 제이쿼리 객체의 메소드입니다.

```
jQuery().attr('속성 이름');
```

'jQuery() 제이쿼리 래퍼객체의 속성 값을 읽어와라.'

```
jQuery().attr('속성 이름', '속성 값');
```

'jQuery() 제이쿼리 래퍼객체의 속성 값을 설정와라.'

.attr(attribute) 함수

```
$('<ul id="abbr_titles"/>').insertAfter('#abbr');
$('abbr', '#abbr').each(function(i) {
    var $abbr_titles = $(this).attr('title');
    $('<li/>').text($abbr_titles).appendTo('#abbr_titles');
});
```

.attr(attribute,value) 함수

```
$('<ul id="abbr_titles"/>').insertAfter('#abbr');
$('abbr', '#abbr').each(function(i) {
    $(this).attr( 'title', 'title_change' + i );
    var $abbr_titles = $(this).attr('title');
    $('<li/>').text($abbr_titles).appendTo('#abbr_titles');
});
```

.attr({attribute1 : value1 , attribute2: value2, ...}) 함수

```
$(‘<img />’).attr({
      id: ‘jungle_image’,
      class: ‘ad’,
      alt: ‘아카데미정글 광고 이미지’
});
```

.attr(attribute, function(index, attr)) 함수

```
$('<img/>').attr('alt', 'art_image').insertBefore('#image_area');
$("img[alt $= "image"]").attr('src', function() {
    return '../img/' + $(this).attr('alt') + '.jpg';
});
```

.removeAttr([selector]) 함수

removeAttr() 함수는 요소의 속성을 제거하는 메소드입니다.

1.6

.prop(property, [value, { map }, function(index, attr)]) 함수

prop() 함수는 요소의 속성 값을 가져오거나, 설정하는 제이쿼리 객체의 메소드입니다.

```
jQuery().prop('속성 이름');
```

'jQuery() 제이쿼리 래퍼객체의 속성 값을 읽어와라.'

```
jQuery().prop('속성 이름', '속성 값');
```

'jQuery() 제이쿼리 래퍼객체의 속성 값을 설정와라.'

For example, consider a DOM element defined by the HTML markup `<input type="checkbox" checked="checked" />`, and assume it is in a JavaScript variable named `elem`:

`elem.checked`	`true` (Boolean)
`$(elem).prop("checked")`	`true` (Boolean)
`elem.getAttribute("checked")`	`"checked"` (String)
`$(elem).attr("checked")` *(1.6+)*	`"checked"` (String)
`$(elem).attr("checked")` *(pre-1.6)*	`true` (Boolean)

1.6

.removeProp([selector]) 함수

removeProp() 함수는 요소의 속성 값을 제거하는 메소드입니다.

.val(value, function(index, value)) 함수

var() 함수는 폼 요소의 값을 가져오거나, 설정하는 제이쿼리 객체의 메소드입니다.

```
jQuery().val('값');
```

'jQuery(폼요소) 제이쿼리 래퍼객체의 값을 설정하라.'

.val() 함수

```
var $hobby_selected = $('select.hobby option:selected').val();
$('p.result_paras').text($hobby_selected);

$('input:checkbox:checked').val();

$('input:radio').filter(':checked').val();
```

.val(value) 함수

```
$('input:button.choice').val(‘선택 버튼’);
$(':checkbox.favorite_sports').val([‘축구’, ‘농구’, ‘야구’]);
```

Insert! in

jQuery - 요소 내부 앞, 뒤에 삽입

.prependTo(target) 함수

prependTo() 함수는 '~의 내부 앞에 자식요소로 삽입하라'라는 의미로
()안에 선택되는 요소의 내부 앞에 . 앞의 요소가 자식요소로 위치하게 됩니다.

```
B.prependTo(A);
```

'B를 A의 안쪽 앞에 자식요소로 삽입하라.'

.prependTo() 함수

```
$('<strong>START!</strong>').prependTo('#disclaimer');
```

.appendTo(target) 함수

appendTo() 함수는 '~의 내부 뒤에 자식요소로 삽입하라'라는 의미로
()안에 선택되는 요소의 내부 뒤에 . 앞의 요소가 자식요소로 위치하게 됩니다.

B.appendTo(A);
'B를 A의 안쪽 뒤에 자식요소로 삽입하라.'

.appendTo() 함수

```
$('<strong>END!</strong>').appendTo('#disclaimer');
```

.prepend(target) 함수

prepend() 함수는 prependTo()와 유사하지만, 차이가 있습니다.
()안에 부모요소가 아닌 자식요소가 위치한다는 점이죠.

A.prepend(B);
'A의 안쪽 앞에 B를 자식요소로 삽입하라.'

.prepend() 함수

```
$('#disclaimer').prepend('<strong>START!</strong>');
```

.append(target) 함수

append() 함수는 append()와 유사하지만, 차이가 있습니다.
()안에 부모요소가 아닌 자식요소가 위치한다는 점이죠.

A.append(B);

'A의 안쪽 뒤에 B를 자식요소로 삽입하라.'

.append() 함수

```
$('#disclaimer').append('<strong>END!</strong>');
```

Insert! out

jQuery - 요소 주변 앞, 뒤에 삽입

1.4

.before(target | function(index)) 함수

before() 함수는 '~ 앞에 삽입하라'라는 의미로
요소의 앞에 ()안의 요소가 위치하게 됩니다.

A.before(B);
'A 앞에 B를 삽입하라.'

.before() 함수

```
$('#disclaimer').before('<input type="button" value="toggle" id="toggleButton">');
$('#toggleButton').click(function() {
    $('#disclaimer').toggle();
});

$('h1').before('<p> new text </p>').prev().css('color', 'tan');
```

.before() 함수

```
$('h2').before(function( i ) {
    if( i % 3 == 0 ) {
        return '<span> 3의 배수 : heading2 앞 </span>';
    }
});
```

1.4

.after(target | function(index)) 함수

after() 함수는 '~ 뒤에 삽입하라'라는 의미로
요소의 뒤에 ()안의 요소가 위치하게 됩니다.

A.after(B);
'A 뒤에 B를 삽입하라.'

.after() 함수

```
$('#disclaimer').after('<input type="button" value="toggle" id="toggleButton">');
$('#toggleButton').click(function() {
    $('#disclaimer').toggle();
});

$('h1').after('<p> new text </p>').next().css('color', 'tan');
```

.after() 함수

```
$('h2').after(function( i ) {
    if( i % 3 == 0 ) {
        return '<span> 3의 배수 : heading2 뒤 </span>';
    }
});
```

.insertBefore(target) 함수

insertBefore() 함수는 '~의 앞에 삽입하라'라는 의미로
()안에 선택되는 요소의 앞에 . 앞의 요소가 위치하게 됩니다.

B.insertBefore(A);
'B를 A의 앞에 삽입하라.'

.insertBefore() 함수

```
$('<input type="button" value="toggle" id="toggleButton">').insertBefore('#disclaimer');
$('#toggleButton').click(function() {
    $('#disclaimer').toggle();
});

$('<p> new text </p>').insertBefore('h1').prev().css('color', 'tan');
```

.insertAfter(target) 함수

insertAfter() 함수는 '~의 뒤에 삽입하라'라는 의미로
()안에 선택되는 요소의 뒤에 . 앞의 요소가 위치하게 됩니다.

B.insertAfter(A);
'B를 A의 뒤에 삽입하라.'

.insertAfter() 함수

```
$('<input type="button" value="toggle" id="toggleButton">').insertAfter('#disclaimer');
$('#toggleButton').click(function() {
    $('#disclaimer').toggle();
});

$('<p> new text </p>').insertAfter('h1').next().css('color', 'tan');
```

Insert! wrap

jQuery - 요소 둘러싸기

.wrap(wrappingElement | function(index)) 함수

1.4

wrap() 함수는 '~ 을 개별적으로 감싸라'라는 의미로
요소를 ()안의 요소를 개별적으로 감쌉니다.

A.wrap(B);
'A를 B로 감싸라.'

DOM 요소 API,
CSS 선택자 표현식,
jQuery 객체,
HTML 코드

1.4

.wrap(wrappingElement | function(index)) 함수

wrap() 함수는 '~ 을 개별적으로 감싸라'라는 의미로
요소를 ()안의 요소를 개별적으로 감쌉니다.

A.wrap(B);
'A를 B로 감싸라.'

.wrap() 함수

```
$('.photo').wrap('<div class=“photo_wrap” />');

$('.photo').wrap(‘div.double_div’);
```

.wrap() 함수

```
$('h2').wrap(function( i ) {
        return '<div title="' + $(this).text() + '"/>';
});
```

.unwrap() 함수

unwrap() 함수는 '~ 을 감싼 부모요소를 제거하라'라는 의미로
요소를 ()안의 요소의 부모가 제거됩니다.

A.unwrap();
'A를 감싼 부모 요소를 제거하라.'

.unwrap() 함수

```
$('button.next_unwrap').click(function() {
    $(this).next().unwrap();
});
```

.wrap(), .unwrap() 응용

```
$('.class').toggle(
    function() {
        $(this).wrap(‘<div id=”new” />’);
    }),
    function() {
        $(this).unwrap();
    }
});
```

.wrapAll(wrappingElement) 함수

wrapAll() 함수는 '~ 을 모두 감싸라'라는 의미로
요소를 ()안의 요소들을 감쌉니다.

A.wrapAll(B);
'A를 B로 감싸라.'

.wrapAll() 함수

```
var $newDiv = $('<div/>');
$newDiv.attr('id', 'new_div').prependTo('body');
$('p').wrapAll($newDiv);
```

1.4

.wrapInner(wrappingElement | function(index)) 함수

wrapInner() 함수는 '~ 으로 ()요소를 감싸라'라는 의미로
()안의 요소를 해당 요소가 감쌉니다.

A.wrapInner(B);
'A로 B를 감싸라.'

.wrapInner() 함수

```
$('.inner').wrap('<div class="inner_wrapInner" />');

$('h2').wrapInner(function() {
        return '<div title="' + $(this).text() + '"/>';
});
```

removing

jQuery - 요소 제거하기

.remove([selector]) 함수

remove() 함수는 DOM Tree에서 ()안 표현식과 일치하는
모든 요소(이벤트 포함)를 제거하는 메소드입니다.

```
jQuery().remove();
```

‘jQuery() 제이쿼리 래퍼객체를 DOM Tree에서 제거하라’

```
jQuery().remove(“:contains(text)’);
```

‘text를 포함한 jQuery() 제이쿼리 래퍼객체를 제거하라’

.remove() 함수

```
$(‘button’).click(function() {
    $('p').remove();
});

$(‘button’).click(function() {
    $('p').remove(':contains(“자바스크립트”)');
    $('abbr').remove('[title="alpha"]');
});
```

.empty() 함수

empty() 함수는 선택된 대상의 내용을 모두 비우는 메소드입니다.
remove() 함수와의 차이점은 대상을 제거하는 것이 아니라는 것입니다.

```
jQuery().empty();
```

'jQuery() 제이쿼리 래퍼객체의 모든 자식요소, 텍스트 노드를 비워라.'

.empty() 함수

```
$(‘button’).click(function() {
    $('p.cookbook').empty();
});
```

1.4

.detach([selector]) 함수

detach() 함수는 remove()와 비슷하지만, 이벤트는 유지하는 메소드입니다.

jQuery().detach();

‘jQuery() 제이쿼리 래퍼객체를 DOM 에서 제거는 하나, 가지고 있는 이벤트는 유지.’

.detach() 함수

```
$(‘p’).click(function() {
    console.log(‘클릭!’)
});

$(‘button.detach’).click(function() {
    var paras = $('p').detach();
});

$(‘button.attach’).click(function() {
    paras.prependTo(‘.attach’);
});
```

Copying

jQuery - 요소 복제하기

.clone() 함수

clone() 함수는 해당 요소를 복제하는 메소드입니다.

A.clone();

'A 요소를 복제하라.'

.clone() 함수

```
$('>p', '#content').filter(':first').clone().appendTo('#aside');
```

.clone(true) 함수

clone() 함수는 해당 요소뿐만 아니라 적용된 이벤트 핸들러까지 복제하는 메소드입니다.

A.clone(true);

'A 요소의 모든 것을 복제하라.'

.clone(true) 함수

```
$('h1').click(function() {
    $('p:last').clone(true).insertAfter(this);
});
```

jQuery_framework
자바스크립트 라이브러리

```
$('h1')
    .data('자바스크립트 라이브러리','제이쿼리')
    .click(function() {
        console.log('h1을 눌렀군요!');
    })
    .clone()
    .insertAfter('ul');
```

```
$('h1')
    .data('자바스크립트 라이브러리','제이쿼리')
    .click(function() {
        console.log('h1을 눌렀군요!');
    })
    .clone(true)
    .insertAfter('ul');
```

Replacing

jQuery - 요소 대체하기(제거하고~)

1.4

.replaceWith(new content | function(index)) 함수

replaceWith() 함수는 해당 요소를 제거하고, ()안의 내용으로 대체하는 메소드입니다.

```
A.replaceWith(B);
```

'A 요소를 제거하고, B로 대체하라.'

.replaceWith() 함수

```
$('<button/>').text('love').prependTo('body').click(function() {
    $(this).replaceWith('<strong>'+ $(this).text() +'</strong>');
})
```


```
for( var i = 0; i < 20; i++ ) {
    $('<button/>')
     .text('생성된 버튼' + ( i + 1 ))
     .prependTo('body')
     .after('<br />')
};
$('button').each(function( i ) {
    $(this).click(function(){
        $(this).replaceWith(function() {
            return ( i % 3 == 0 ) ?
                    '<em>' + $(this).text() + '</em>' : '<strong>' + $(this).text() + '</strong>'
        });
    });
});
```

.replaceAll(selector) 함수

replaceAll() 함수는 ()안의 요소를 제거하고, .앞의 요소로 대체하는 메소드입니다.

```
B.replaceAll(A);
```

'A 요소를 제거하고, B 요소로 대체하라.'

.replaceAll() 함수

```
$(‘button’).click(function () {
    $('<span/>').text('change p:first').replaceAll('p:first')
});
```