



Traversing

jQuery - 순회(탐색)

Filtering

jQuery - 필터링

1.4

.eq(index | -index) 필터 함수

eq(정수) 함수는 ()안에 전달된 (정수+1)의 순서와 일치하는 jQuery객체를 필터링하여 반환(return)합니다.



.eq(정수) 함수 : jQuery 객체를 돌려줌.

```
$('div').eq(2).addClass('floatLeft');
```

1.4

.not(selector | elements | function(index)) 필터 함수

not() 함수는 ()안의 전달된 값을 통해 일치하는 대상을 제외한
필터링된 jQuery객체를 반환(return)합니다.

.not() 함수 : jQuery 객체를 돌려줌.

```
$('#div').not('#header, #footer').addClass('floatLeft');  
$('#p').not(  
    function(index) {  
        return (++index % 3) == 0;  
    }  
).removeClass();
```


.filter(selector | function(index) | element | jQuery Object) 함수

filter() 함수는 ()안의 전달받은 값을 통해 필터링한 jQuery 객체를 반환(return)합니다.

.filter() 함수 : jQuery 객체를 돌려줌.

```
$('#p').filter(':contains("yamoo9")').addClass('ellipsis');  
  
$('#p').filter(  
    function() {  
        return $(this).css('display') === 'inline'  
    }  
).addClass('ellipsis');
```

.is(selector) 필터 함수

is() 함수는 ()안에 전달받은 값을 통해 필터링 한 후.. 참, 거짓을 반환(return)합니다.



.is() 함수 : 참 또는 거짓을 돌려줌.

```
if($('p').is(':has(span)')) {  
    $(this).addClass('ellipsis');  
};
```

.each(function(index, element)) 필터 함수

`each()` 함수는 대상들(요소노드세트) 내부 요소노드에 접근하여 개별적으로 다른 일을 제어할 수 있습니다. 마치 for문과 비슷하다고 이해하시면 좋습니다.

.each() 함수 : jQuery 객체를 돌려줌.

```
$('.li').each(  
    function(index) {  
        $(this).html(index + ':' + $(this).html());  
    }  
);
```

`.slice(start, [end])` 필터 함수

`slice()` 함수는 ()안의 필터를 통해 0 인덱스를 시작으로하는 start 숫자에 해당하는 요소에서 end-1 숫자에 해당하는 요소까지의 jQuery객체를 반환(return)합니다.



.slice() 함수 : jQuery 객체를 돌려줌.

```
$('#div').slice(0, 2).addClass('floatLeft');  
$('p').slice(3).addClass('overline');  
$(':header').slice(-1).addClass('last_heading');  
$('span').slice(3, -2).addClass('last_heading');
```


Finding

jQuery - 찾기

.find(selector) 함수

find() 함수는 ()안에 전달받은 값을 통해 대상의 하위 요소중 일치하는 요소를 찾아 jQuery 객체로 반환(return)합니다.



.find() 함수 : jQuery 객체를 돌려줌.

```
$('#p').find('em').addClass('emphasis');
```

.next([selector]) 함수

next() 함수는 ()안에 전달받은 값을 통해 일치하는 다음에 위치한 요소를 찾아 jQuery 객체로 반환(return)합니다.

.next() 함수 : jQuery 객체를 돌려줌.

```
$('#p:first').next().text('첫번째 p의 다음 요소입니다.');
```

```
$('#p:gt(3)').next('.selected').text('p의 4번째 이후 요소중 selected 클래스를 가진 요소입니다.');
```



.prev([selector]) 함수

prev() 함수는 ()안에 전달받은 값을 통해 일치하는 이전에 위치한 요소를 찾아 jQuery 객체로 반환(return)합니다.

.prev() 함수 : jQuery 객체를 돌려줌.

```
$('#p:nth-of-type(3)').prev().text('3번째 p의 이전 요소입니다.');
```

```
$('#p:eq(3)').prev('.selected').text('p 3번째 이전 요소가 selected 클래스를 가졌습니다.');
```

.nextAll([selector]) 함수

nextAll() 함수는 ()안에 전달받은 값을 통해 일치하는 이후의 모든 요소들을 찾아 jQuery 객체로 반환(return)합니다.

.nextAll() 함수 : jQuery 객체를 돌려줌.

```
$('#p:nth-of-type(3)').nextAll().text('3번째 p의 이후 요소들입니다.');
```

```
$('#p:first').nextAll('.selected').text();
```

.prevAll([selector]) 함수

prevAll() 함수는 ()안에 전달받은 값을 통해 일치하는 이전의 모든 요소들을 찾아 jQuery 객체로 반환(return)합니다.

.prevAll() 함수 : jQuery 객체를 돌려줌.

```
$('#p:nth-of-type(3)').prevAll().text('3번째 p의 이전 요소들입니다.');
```

```
$('#p:last').prevAll('.selected').html();
```

.nextUntil([selector]) 함수

nextUntil() 함수는 ()안에 전달받은 값을 통해 일치하는 요소 전까지의 이후에 위치한 요소들을 찾아 jQuery 객체로 반환(return)합니다.

.nextUntil() 함수 : jQuery 객체를 돌려줌.

`$('#h2').nextUntil();` // h2 요소 이후의 모든 요소들. nextAll()과 같음.
`$('#p').nextUntil('u');` // h2 요소 이후의 요소 중, u 요소 이전까지의 요소들.

.prevUntil([selector]) 함수

prevUntil() 함수는 ()안에 전달받은 값을 통해 일치하는 요소 전까지의 이전에 위치한 요소들을 찾아 jQuery 객체로 반환(return)합니다.

.prevUntil() 함수 : jQuery 객체를 돌려줌.

`$('#h2').prevUntil();` // h2 요소 이전의 모든 요소들. prevAll()과 같음.
`$('#p').prevUntil('ul');` // h2 요소 이전의 요소 중, ul 요소 이전까지의 요소들.

.siblings([selector]) 함수

siblings() 함수는 ()안에 전달받은 값을 통해 일치하는
형제요소들(앞, 뒤 모두)을 찾아 jQuery 객체로 반환(return)합니다.

.siblings() 함수 : jQuery 객체를 돌려줌.

```
$('#p.selected').siblings().text('p.selected 주변 요소입니다.');
```

```
$('#headers').siblings('.selected').text('#headers 주변 요소 중 selected 클래스 소유 요소');
```

.parent([selector]) 함수

parent() 함수는 ()안에 전달받은 값을 통해 일치하는 부모요소를 찾아
jQuery 객체로 반환(return)합니다.



.parent() 함수 : jQuery 객체를 돌려줌.

```
$('#span.title').parent().addClass('view');  
$('#p').parent('#header').addClass('view');
```

.children([selector]) 함수

children() 함수는 ()안에 전달받은 값을 통해 일치하는 자식요소들을 찾아 jQuery 객체로 반환(return) 합니다.

.children() 함수 : jQuery 객체를 돌려줌.

```
$('#td.selected').parent().children().addClass('view');  
$('#li:not(span)').parent().children('.selected').addClass('view');
```

.parents([selector]) 함수

parents() 함수는 ()안에 전달받은 값을 통해 일치하는 부모요소들을 찾아
jQuery 객체로 반환(return) 합니다.



.parents() 함수 : jQuery 객체를 돌려줌.

```
$('#span.title').parents().addClass('view');  
$('#p').parents('.selected').addClass('view');
```

.parentsUntil([selector]) 함수

parentsUntil() 함수는 ()안에 전달받은 값을 통해 일치하는 요소 전까지의 이후에 위치한 부모요소들을 찾아 jQuery 객체로 반환(return)합니다.

.parentsUntil() 함수 : jQuery 객체를 돌려줌.

```
$('#span.title').parentsUntil(); // 전달받는 값이 없으면 parents()와 같음.  
$('#p').parentsUntil('.selected');
```

.contents() 함수

content() 함수는 ()안에 전달받은 값을 통해 해당 요소의 모든 노드(Node)들을 jQuery 객체로 반환(return)합니다.

.contents() 함수 : jQuery 객체를 돌려줌.

```
$('#p').contents().filter(function() {  
    return this.nodeType == 3;  
});
```

.closest(selector, [context]) 함수

closest() 함수는 ()안에 전달된 값을 통해 일치하는 부모 요소 중,
가까운 요소를 jQuery 객체로 반환(return)합니다.

.closest() 함수

```
$(this).closest('li').toggleClass('highlight');
```

chaining

jQuery - 연결하기

1.4

.add(selector | elements | html | [selector, context]) 함수

add() 함수는 ()안에 전달받은 값을 통해 ()안의 요소를 현재 요소에 추가한
jQuery 객체를 반환(return)합니다.



.add() 함수 : jQuery 객체를 돌려줌.

```
$('#p').add('span.selected').addClass('emphasis');
```


.andSelf() 함수

andSelf() 함수는 현재의 jQuery 객체가 이전의 객체와 달라졌을 경우, 현재의 요소에 이전의 요소를 추가한 jQuery 객체를 반환(return)합니다.



.andSelf() 함수 : jQuery 객체를 돌려줌.

```
$('#div').find('p').andSelf().addClass('emphasis');
```

.end() 함수

end() 함수는 현재의 jQuery 객체가 이전의 객체와 달라졌을 경우, 현재의 요소가 아닌 이전의 요소로 jQuery 객체를 반환(return)합니다.

.end() 함수 : jQuery 객체를 돌려줌.

```
$('#div').find('p').end().addClass('emphasis');
```