

jQuery 객체탐색

filtering 메소드

- 선택자로 지정한 객체를 기준으로 객체 그룹에서 위치를 기준으로 객체 선택하는 메소드

메소드	내용
<code>\$('s').first()</code>	선택된 요소 중 제일 처음 있는 요소 리턴
<code>\$('s').last()</code>	선택된 요소 중 제일 마지막에 있는 요소 리턴
<code>\$('s').eq(숫자)</code>	인덱스번호와 일치하는 요소 리턴 (0부터시작)
<code>\$('s').filter(“검색식 선택자”)</code>	인자값과 일치하는 요소만 리턴
<code>\$('s').not()</code>	인자값과 일치하지 않는 요소만 리턴

Ancestors 메소드

- 선택된 요소의 상위 요소들을 선택할 수 있는 메소드

메소드	내용
<code>\$('s').parent()</code>	선택된 요소의 바로 위 상위요소만 리턴
<code>\$('s').parents([인자])</code>	선택된 요소의 모든 상위 요소 리턴, 매개변수가 있으면 매개변수와 일치하는 부모만 리턴
<code>\$('s').parentsUntil([인자])</code>	선택된 요소부터 인자요소까지 범위의 요소 리턴

Descendants 메소드

- 선택된 요소의 하위 요소들을 선택할 수 있는 메소드

메소드	내용
<code>\$('s').children([인자])</code>	선택된 요소의 모든 자손(다음레벨)객체를 리턴 선택된 요소의 인자와 일치하는 자손(다음레벨)객체 리턴
<code>\$('s').find(인자)</code>	선택된 요소의 인자와 일치하는 모든 후손(모든 레벨) 객체 리턴

Sideways 메소드

- 같은 레벨에 있는 요소(형제)를 선택할 수 있는 메소드

메소드	내용
<code>\$('s').siblings([인자])</code>	선택된 요소와 같은 레벨(형제)에 있는 요소리턴 선택된 요소의 같은 레벨(형제) 중 인자와 일치하는 요소 리턴
<code>\$('s').next()</code>	선택된 요소의 같은 레벨(형제)들 중 선택된 요소 다음의 한개 요소를 리턴
<code>\$('s').nextAll()</code>	선택된 요소의 같은 레벨(형제)들 중 선택된 요소 다음의 모든 요소를 리턴
<code>\$('s').nextUntil(인자)</code>	선택된 요소부터 같은 레벨(형제)들 중 인자까지 범위의 모든 요 소를 리턴

Sideways 메소드

메소드	내용
<code>\$('s').prev()</code>	선택된 요소의 같은 레벨(형제)들 중 선택된 요소 이전의 한개 요소를 리턴
<code>\$('s').prevAll()</code>	선택된 요소의 같은 레벨(형제)들 중 선택된 요소 이전의 모든 요소를 리턴
<code>\$('s').prevUntil(인자)</code>	선택된 요소의 같은 레벨(형제)들 중 선택된 요소 이전의 요소를 리턴

객체탐색 - 순회(탐색) 메소드(Traversing)

요소가 있는지 찾는 메소드

- 선택자로 지정된 범위에 특정한 요소가 존재하는지 찾을 때 사용하는 메소드

메소드	내용
<code>\$('s').is(인자)</code>	선택된 요소의 범위내에 인자와 동일한 요소가 있는지 찾아서 있으면 true, 없으면 false 리턴

jQuery 객체조작

each 메소드

- 배열을 관리하는 for in문과 비슷한 메소드로 객체나 배열의 요소를 검사하는 메소드
- index : 객체, 배열의 순번
- item : 값을 보관하는 변수(객체) ‘.’으로 접근

메소드	내용
<code>\$.each(배열이름, function(index, item) { })</code>	지정한 배열을 0번 부터 자동으로 불러와 순번을 index, 값을 item에 넣는 메소드
<code>\$('s').each(function(index,item){ })</code>	선택자로 선택한 요소를 index순번으로 item에 요소 값을 수정시 사용하는 메소드

each 메소드

```
$.each(배열(객체)이름, function(index, item){
```

배열(객체) 차례로 처리할 때 사용

```
}
```

```
);
```

```
$(‘선택자’).each(function(index, item){
```

선택된 요소들을 차례로 조작할때 사용

```
}
```

```
);
```

extend 메소드

- jQuery에서 객체의 내용을 추가하는 메소드

```
$.  
$.extend(객체이름, {  
    추가할 객체내용,  
    추가할 객체내용2,  
    .....  
}  
);
```

addClass()

➤ 선택자에 의해 선택된 요소에 클래스를 추가하는 메소드

1. `$(‘s’).addClass(“클래스이름”);`
//클래스를 여러 개 추가는 띄어쓰기로 구분
2. `$(‘s’).addClass(function(n){`
 인자는 선택된 요소들의 번호;
 return 값; // 리턴된 값을 클래스로 추가
});

removeClass()

➤ 선택자에 의해 선택된 요소에 클래스를 삭제하는 메소드

1. `$('s').removeClass("클래스이름");`
//클래스를 여러 개 추가는 띄어쓰기로 구분
2. `$('s').removeClass(function(n){`
 인자는 선택된 요소들의 번호;
 return 값; //리턴된 값과 일치하는 클래스삭제
});

attr()

➤ 선택자에 의해 선택된 요소에 속성을 속성값확인, 속성값 변경하는 메소드

1. `$('s').attr("속성명");`
//속성명의 속성값을 리턴
2. `$('s').attr("속성명", "속성값");`
//선택자의 속성과 속성값을 추가
3. `$('s').attr("속성명", function(index, value) { });`
//index : 선택된 요소 set의 index값, value : 현재 값
4. `$('s').attr({ 속성명: "속성값", 속성명: "속성값" });`
//다중으로 속성값 설정시 사용, 객체사용 가능

removeAttr()

➤ 선택자에 의해 선택된 요소에 속성을 제거하는 메소드

1. `$('s').removeAttr("속성명");`

//속성명의 속성을 제거, 여러개일 경우 띄어쓰기 사용

css()

- 선택자에 의해 선택된 요소의 속성값을 가져오거나 설정하는 메소드
 - ☞ 단축속성설정(예 : border, background)제대로 지원 않고 브라우저별 상이

1. `$('s').css("속성명");`
//속성명의 속성값을 리턴
2. `$('s').css("속성명", "속성값");`
//속성명의 속성값을 설정
3. `$('s').css("속성명", function(index, 현재값) { });`
//함수에서 리턴되는 값으로 속성을 설정
4. `$('s').css({오브젝트});`
//오브젝트에서 설정한 다중 속성을 설정

html()

- 선택된 요소의 content영역(innerHTML/텍스트노드)을 리턴 / 설정하는 메소드, html태그를 태그로 인식
- 리턴시 첫번째 요소 텍스트노드 리턴 / 전체 요소 리턴시 함수 사용

1. `$('s').html();`

//선택된 요소의 첫번째 텍스트 노드를 리턴

2. `$('s').html("text노드 내용");`

//선택된 요소의 text노드 내용을 대입

3. `$('s').html(function(index, 현재값) { });`

text()

- 선택된 요소의 content영역(innerHTML/텍스트노드)을 리턴 / 설정하는 메소드, html태그를 문자로 인식
- 리턴시 전체 요소 텍스트노드를 리턴

1. `$('s').text();`

//선택된 요소의 모든 텍스트 노드를 리턴

2. `$('s').text("text노드 내용");`

//선택된 요소의 text노드 내용을 대입

3. `$('s').text(function(index, 현재값) { });`

문서 객체 생성방법

구분	내용
html 태그	직접 html구문을 문자열로 작성 예) ' <code><p>직접 생성하는 방식</p></code> '
jQuery 방식	jQuery 객체 이용 구문작성 <code>\$(생성태그).css() html() text() attr()</code> 사용하여 생성
DOM 방식	<code>document.createElement('태그명').innerHTML('텍스트노드내용');</code> <code>document.createElement('태그명').setAttribute('속성명','속성값');</code>

문서 객체 추가/이동

➤ html내부에 생성한 요소를 추가하는 메소드

메소드	내용
<code>\$('s').append(생성구문)</code> <code>\$('s').append(function([n],[html]){ })</code>	생성된 요소를 선택자 뒤에 추가. 함수의 리턴된 값을 선택자 뒤에 추가. ☞ n : 선택된 요소set의 인덱스 번호 ☞ html : 현재 선택된 요소
<code>\$('s').after(생성구문)</code> <code>\$('s').after(function([n],[html]){ })</code>	
<code>\$('생성구문 태그명').appendTo('s')</code>	선택자에 생성된 요소나 태그를 뒤에 추가 ☞ 만일 추가되는 태그명이 문서에 있으면 위치를 변경시킴(이동)
<code>\$('생성구문 태그명').insertAfter('s')</code>	

문서 객체 추가/이동

메소드	내용
<code>\$('s').prepend(생성구문)</code> <code>\$('s').prepend(function([n],[html]){ })</code>	생성된 요소를 선택자 앞에 추가. 함수의 리턴된 값을 선택자 앞에 추가. ☞ n : 선택된 요소set의 인덱스 번호 ☞ html : 현재 선택된 요소
<code>\$('s').before(생성구문)</code> <code>\$('s').before(function([n],[html]){ })</code>	
<code>\$('생성구문 태그명').prependTo('s')</code>	선택자에 생성된 요소나 태그를 앞에 추가 ☞ 만일 추가되는 태그명이 문서에 있으면 위치를 변경시킴(이동)
<code>\$('생성구문 태그명').insertBefore('s')</code>	

clone 메소드

- html내부에 요소객체를 복사하는 메소드

```
$(‘태그명’).clone([true | false]).appendTo(“추가될위치”);
```

```
// 태그명의 태그를 복사하여 추가될 위치에 넣기
```

```
// 인자값 이벤트 핸들러까지 복사할지 여부까지
```

문서 객체 제거 메소드

➤ html내부에 요소객체를 제거하는 메소드

메소드	내용
<code>\$(<i>'s[s...]</i>').remove(<i>'s'</i>);</code>	문서 내에 선택자를 제거 후 삭제된 요소를 리턴 ☞ 인자값을 기준으로 삭제가능 ☞ 이벤트핸들러, 데이터까지 모두 삭제
<code>\$(<i>'s'</i>).detach();</code>	문서 내에 선택자를 제거 후 삭제된 요소를 리턴 ☞ 이벤트핸들러, 데이터는 삭제 안함 ☞ 값을 리턴받아 나중에 활용 가능
<code>\$(<i>'s'</i>).empty();</code>	문서 내에 선택자를 제거 후 삭제된 요소를 리턴 ☞ 선택자 자신과 속성은 삭제 안하고 내용만 삭제