

객체 조작

content 설정

▶ jQuery content

✓ html()

선택된 요소의 content영역(innerHTML/텍스트노드) 리턴하고

설정하는 메소드(html태그를 태그로 인식)

리턴 시 첫 번째 요소 텍스트 노드 리턴, 전체 요소 리턴 시 함수 사용

`$('요소 명').html();` * 선택된 요소의 첫 번째 텍스트 노드 리턴

`$('요소 명').html('텍스트 노드 내용');` * 선택된 요소의 text노드 내용 대입

`$('요소 명').html(function(index, 현재 값){});`

▶ jQuery content

✓ text()

선택된 요소의 content영역(innerHTML/텍스트노드) 리턴하고
설정하는 메소드(html태그를 태그로 인식 불가)
리턴 시 전체 요소 텍스트 노드 리턴

`$('요소 명').text();` * 선택된 요소의 모든 텍스트 노드 리턴

`$('요소 명').text('텍스트 노드 내용');` * 선택된 요소의 text노드 내용 대입

`$('요소 명').text(function(index, 현재 값){});`

클래스/속성 설정

▶ jQuery class

✓ addClass()

선택자에 의해 선택된 요소에 클래스를 추가하는 메소드

```
$('#요소 명').addClass('클래스 이름'); * 띄어쓰기로 클래스 여러 개 추가 가능  
$('#요소 명').addClass(function(n){  
    인자는 선택된 요소들의 번호;  
    return 값; * 리턴된 값을 클래스로 추가  
});
```

▶ jQuery class

✓ removeClass()

선택자에 의해 선택된 요소에 클래스를 삭제하는 메소드

```
$('#요소 명').removeClass('클래스 이름'); * 띄어쓰기로 클래스 여러 개 삭제
```

```
$('#요소 명').removeClass(function(n){
```

```
    인자는 선택된 요소들의 번호;
```

```
    return 값; * 리턴된 값과 일치하는 클래스 삭제
```

```
});
```

▶ jQuery class

✓ toggleClass()

선택된 요소에 class가 있으면 삭제하고 없으면 추가하는 메소드

```
$('#요소 명').toggleClass('클래스 이름'); * 띄어쓰기로 클래스 여러 개 가능  
$('#요소 명').toggleClass('클래스 명'[,function(index, curclass){}],  
true|false]);
```

* true는 추가만, false는 삭제만

▶ jQuery 속성

✓ attr()

선택자에 의해 선택된 요소에 속성 값 확인 및 변경하는 메소드

`$('요소 명').attr('속성 명');` * 속성 명의 속성 값 리턴

`$('요소 명').attr('속성 명', '속성 값');` * 선택자의 속성과 속성 값 추가

`$('요소 명').attr('속성 명', function(index, value){});`

* index : 선택된 요소의 index 값, value : 현재 값

`$('요소 명').attr({속성 명:'속성 값', 속성 명:'속성 값'});`

* 다중으로 속성 값 설정 시 사용, 객체 사용 가능

▶ jQuery 속성

✓ removeAttr()

선택자에 의해 선택된 요소에 속성을 제거하는 메소드

```
$('#요소 명').removeAttr('속성 명');
```

* 속성 명의 속성 제거, 여러 개일 경우 띄어쓰기 사용

▶ jQuery 속성

✓ css()

선택자에 의해 선택된 요소의 속성 값을 가져오거나 설정하는 메소드

* 단축속성설정(ex. border, background) 제대로 지원 않고 브라우저 별 상이

`$('.요소 명').css('속성 명');` * 속성 명의 속성 값 리턴

`$('.요소 명').css('속성 명', '속성 값');` * 속성 명의 속성 값 설정

`$('.요소 명').css('속성 명', function(index, 현재 값){});`

* 함수에서 리턴 되는 값으로 속성 설정

`$('.요소 명').css({오브젝트});` * 오브젝트에서 설정한 다중 속성 설정

객체 생성 및 제거

▶ jQuery 객체 생성

✓ 문서 객체 생성

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

▶ jQuery 객체 생성

✓ html 내부에 생성한 요소 추가

메소드	내용
<pre>\$(요소명).append(생성구문) \$(요소명).append(function([n],[html]){ })</pre>	<p>선택자 요소를 기준으로 매개변수에 작성된 생성된 요소 또는 함수의 리턴값을 추가</p> <p>* n : 선택된 요소set의 인덱스 번호 * html : 현재 선택된 요소</p>
<pre>\$(요소명).after(생성구문) \$(요소명).after(function([n],[html]){ })</pre>	
<pre>\$(요소명).prepend(생성구문) \$(요소명).prepend(function([n],[html]){ })</pre>	
<pre>\$(요소명).before(생성구문) \$(요소명).before(function([n],[html]){ })</pre>	
<pre>\$('생성구문 태그명').appendTo(요소명)</pre>	<p>생성된 요소를 매개변수로 지정한 선택자 요소에 추가</p> <p>* 만일 추가되는 태그명이 문서에 있으면 위치를 변경시킴(이동)</p>
<pre>\$('생성구문 태그명').insertAfter(요소명)</pre>	
<pre>\$('생성구문 태그명').prependTo(요소명)</pre>	
<pre>\$('생성구문 태그명').insertBefore(요소명)</pre>	

▶ jQuery 객체 복제

✓ clone()

html내부에 있는 요소를 복사하는 메소드

```
$('#태그 명').clone([true|false]).appendTo('추가될 위치');
```

* 태그 명의 태그를 복사하여 추가될 위치에 놓으며

인자 값은 이벤트 핸들러까지 복사할지 여부를 묻는 것

▶ jQuery 객체 제거

✓ html내부에 요소 객체 제거

메소드	내용
<code>\$('s[,s...]').remove(['s']);</code>	<p>문서 내에 선택자 제거 후 삭제된 요소 리턴</p> <ul style="list-style-type: none"> * 인자값을 기준으로 삭제 가능 * 이벤트핸들러, 데이터까지 모두 삭제
<code>\$('s').detach();</code>	<p>문서 내에 선택자 제거 후 삭제된 요소 리턴</p> <ul style="list-style-type: none"> * 이벤트핸들러, 데이터는 삭제 안 함 * 값을 리턴 받아 나중에 활용 가능
<code>\$('s').empty();</code>	<p>문서 내에 선택자 제거 후 삭제된 요소 리턴</p> <ul style="list-style-type: none"> * 선택자 자신과 속성은 삭제 안하고 내용만 삭제

기본 설정

▶ jQuery 배열 관리

✓ each()

- 객체나 배열을 관리하는 for in문과 비슷한 메소드로 객체나 배열의 요소를 검사
- index : 객체/배열의 순번
- item : 값을 보관하는 변수(객체), '.'으로 접근

메소드	내용
<code>\$.each(객체명, function(index, item){})</code>	지정한 객체를 0부터 자동으로 불러와 순번과 값을 각각 index와 item에 넣는 메소드
<code>\$('요소명').each(function(index, item){})</code>	선택자로 선택한 요소를 index순번으로 item에 요소 값을 수정 시 사용하는 메소드

▶ jQuery 배열 관리

```
$.each(배열(객체)이름, function(index, item){  
    배열(객체)을 차례로 처리할 때 사용  
});
```

```
$('선택자').each(function(index, item){  
    선택된 요소들을 차례로 조작할 때 사용  
});
```

▶ jQuery 객체 확장

✓ extend()

jQuery에서 객체의 내용을 추가하는 메소드

```
$.extend(객체 이름, {  
    추가할 객체 내용1,  
    추가할 객체 내용2,  
    .....  
});
```

▶ jQuery 충돌 방지

✓ noConflict()

jQuery변수를 \$ 외에 다른 것으로 지정하는 메소드로
다른 JavaScript플러그인을 사용 시 특수문자 \$변수를 중복되어 발생
하는 에러 방지를 위해 사용하기도 함

```
var 변수 명 = $.noConflict();
```

* 변수 명을 \$ 대신 사용