

15장. 문서 객체 조작

addClass() : 문서 객체에 클래스 속성 추가

`<h1> → <h1 class="item">`
↳ `$(‘h1’).addClass(‘item’);`

`<h1> → <h1 class="class0">`

`<h1> → <h1 class="class1">`

`<h1> → <h1 class="class2">`

①, 1, 2

↳ `$(‘h1’).addClass(function(index){
 return ‘class’+index;
});`

removeClass() : 문서 객체에 클래스 속성 제거

매개변수에 아무것도 입력하지 않으면
문서 객체의 모든 클래스를 제거한다.

`$(‘h1’).removeClass(‘select’);`

toggleClass() : 매개변수로 입력한 클래스 속성이
있으면 제거, 없으면 추가해줌.

attr() : 문서 객체의 속성과 관련된 모든 기능 수행
여러개 있으면 첫번째 객체로 가져온다.

↳ img 태그의 src 속성값?
\$(‘img’).attr(‘src’);

문서 객체의 속성 추가

1. (selector).attr(name, value);
 2. (selector).attr(name, function(index, attr){});
 3. (selector).attr(object);
- \$(‘img’).attr(‘width’, 200);
 - \$(‘img’).attr(‘width’, function(index){
return (index + 1) * 100;
});
 - \$(‘img’).attr({
width: 300,
height: function(index){
return (index + 1) * 100;
}
});

문서 객체의 속성 제거

removeAttr(name) : 문서 객체의 속성을 제거합니다.

```
<h1 data-index="0"> → <h1>  
$(h1).removeAttr('data-index')
```

css() : 문서 객체의 스타일 속성과 관련된 작업 수행

```
<style>  
    .first {color: red;}  
</style>
```

```
<h1 class="first">  
    ↳ 어떤 커스터마이징을  
        ↳ $(h1).css('color');
```

문서 객체의 스타일 추가

1. \$(selector).css(name, value);
2. \$(selector).css(name, function(index, style){});
3. \$(selector).css(object);

- \$(h1).css('color', 'red');
- var color = ['red', 'white', 'purple'];
 ↳ 1, 2
 \$(h1).css('color', function(index)){
 return color[index];
 };
- \$(h1).css({
 color: function(index){
 return color[index];
 },

```
        background-color: 'black'  
    };
```

문서 객체의 내부 검사 (innerHTML, textContent)

html(), text()

↳ 문서 객체 내부의 글자와 관련된 모든 기능 수행

<h1> Header - 1 </h1>

<h2> Header - 2 </h1>

\$('.h1').html() → HTML 태그인식, 첫번째 값만!

Header - 1

\$('.h1').text() → HTML 태그인식 X, 모든 값!

Header - 1 Header - 2

문서 객체의 내부 추가

1. \$(selector).html(value);

\$(selector).text(value);

2. \$(selector).html(function(index, html){});

\$(selector).text(function(index, html){});

<div> </div>

→ HTML 태그 인식

1. \$('#div').html('<h1>\$().html() Method </h1>');

↳ <div><h1> \$(().html() Method </h1> </div>
\$(div).text('<h1>\$(().html() Method </h1>)');
↳ <div><h1>> \$(().text() Method ...

2. \$('div').html(function(index){
 return '<h1>Header-' + index + '</h1>')

\$(div).html(function(index, html){
 return '*' + html + '*');
})

↳ 원래의 HTML
내용에 접근

문서 객체 제거

remove() : 문서 객체를 제거합니다.

empty() : 문서 객체 내부를 비웁니다.

\$('h1').first().remove(); → 첫번째 <h1> 제거.

\$(div).empty(); → body 태그의 모든 <h1> 제거.

문서 객체 생성. [텍스트 노드를 갖는 문서 객체
" " 갖지 않는 "

\$(()) : 문서 객체 생성

\$(<h1> </h1>).html('Hello').appendTo('body');

① 문서 객체 생성

② 내용을 추가

③ body 태그에 추가

텍스트 노드를 갖지 않는 문서객체

```
$( '<img>' ).attr( 'src', '~.jpg' ).appendTo( 'body' );  
$( '<img>' ).
```

src: '~.jpg',

width: 100

```
).appendTo( 'body' );
```

문서객체 삽입 1

\$(A).appendTo(B) : A를 B의 뒷부분에 추가

\$(A).prependTo(B) : " 앞부분 "

\$(A).insertAfter(B) : " 뒤 "

\$(A).insertBefore(B) : " 앞 "

문서객체 삽입 2

\$(A).append(B) : B를 A의 뒷부분에 추가

\$(A).prepend(B) : " 앞부분 "

\$(A).after(B) : " 뒤 "

\$(A).before(B) : " 앞 "

`$(selector).append(content, content ...)`

```
var h1 = '<h1>Header1 </h1>';  
var h2 = '<h2>Header2 </h2>';  
$('body').append(h1, h2, h1, h2);
```

`$(selector).append(function(index){});`

`<div> </div>
<div> </div>`

`var Content = [`

```
{ name: '신종희', age: '25' },  
{ name: '신종미', age: '27' }
```

`];`

`$('.div').append(function(index){`

`var item = Content[index];`

`var output = '';`

`output += '<h1>' + item.name + '</h1>';`

`output += '<h2>' + item.age + '</h2>';`

`return output;`

`});`

문서 객체 이동.

setInterval(function () {

 \$('img').first().appendTo('body');
 }, 2000);
 ↳ 첫번째 이미지가 body 태그에
 뒷부분으로 옮겨지면서
 이미지의 순서가 계속 변경된다.

문서 객체 복제)

\$('div').append(\$('h1').clone());

 ↳ <h1> ~ </h1> 복제하고
 <div> 티에 붙힌다