

# 객체조작





객체조작

# 객체조작

## each() 메소드

- 객체나 배열의 요소를 검사하는 메소드

```
$.each(배열이름,function(index,item){})
```

- 지정한 배열을 0번부터 불러와 순번을 index, 값을 item에 넣는 메소드
- item은 Element로 javascript객체이므로 jQuery메소드 사용 불가

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

- 선택자로 선택한 요소를 index 순번으로 item에 요소값 조작시 사용
- 선택된 요소를 차례로 조작할 때 사용



# 객체조작

## addClass() 메소드

- 선택자에 의해 선택된 요소에 클래스를 추가하는 메소드

```
$(선택자).addClass("클래스이름");
```

- 클래스를 여러 개 추가하는 경우 띄어쓰기로 구분
- Class를 통해 CSS디자인을 한 후 선택적을 적용하는 경우 많이 사용



# 객체조작

## removeClass() 메소드

- 선택자에 의해 선택된 요소에 클래스를 삭제하는 메소드

```
$(선택자).removeClass("클래스이름");
```

- 클래스를 여러 개 추가하는 경우 띄어쓰기로 구분
- Class를 통해 CSS디자인을 한 후 선택적을 삭제하는 경우 많이 사용



# 객체조작

## attr() 메소드

- 선택자에 의해 선택된 요소의 속성값을 확인 및 변경하는 메소드

```
$(선택자).attr("속성명");
```

- 선택자의 현재 속성 값을 확인

```
$(선택자).attr("속성명", "속성값");
```

- 선택자의 속성과 속성값을 추가

```
$(선택자).attr({"속성명1": "속성값1", "속성명2": "속성값2", ...});
```

- 여러개의 속성값 설정 시 객체 사용 가능



# 객체조작

## removeAttr() 메소드

- 선택자에 의해 선택된 요소의 속성을 제거하는 메소드

```
$(선택자).removeAttr("속성명");
```

- 속성명을 제거
- 제거할 속성이 여러 개인 경우 띄어쓰기 사용



# 객체조작

## css() 메소드

- 선택자에 의해 선택된 요소의 css속성 값을 가져오거나 설정하는 메소드  
→ 단축속성설정(border,background..)의 경우 제대로 지원하지 않고 브라우저별로 상이  
`$(선택자).css("속성명");`
- 선택자의 현재 속성 값을 확인

`$(선택자).css("속성명", "속성값");`

- 선택자의 속성과 속성값을 추가

`$(선택자).css({"속성명1": "속성값1", "속성명2": "속성값2", ...});`

- 여러개의 속성값 설정 시 객체 사용 가능



# 객체조작

## html() 메소드

- 선택된 요소의 **content**영역(innerHTML)을 리턴
- **content**영역에 태그가 있다면 태그까지 가져옴

```
$(선택자).html();
```



# 객체조작

## text() 메소드

- 선택된 요소의 content영역(innerHTML)을 리턴
- content영역에 태그가 있다면 태그는 가져오지 않음

```
$(선택자).text();
```



# 객체조작

## text() 메소드

- 선택된 요소의 content영역(innerHTML)을 리턴
- content영역에 태그가 있다면 태그는 가져오지 않음

```
$(선택자).text();
```

# 객체조작

## 문서객체 생성

구분	내용
HTML 태그 생성	직접 html 구문을 문자열로 작성 '<p>직접생성</p>'
속성 설정	<code>\$(생성태그).css()   html()   attr()</code> 등

# 객체조작

## 생성된 객체 추가/이동

구분	내용
<code>\$(생성객체).appendTo(선택자)</code>	생성객체를 선택자의 마지막 자식으로 추가
<code>\$(생성객체).prependTo(선택자)</code>	생성객체를 선택자의 첫번째 자식으로 추가
<code>\$(생성객체).insertAfter(선택자)</code>	생성객체를 선택자의 다음 형제로 추가
<code>\$(생성객체).insertBefore(선택자)</code>	생성객체를 선택자의 이전 형제로 추가

※ 만일 추가되는 객체가 문서에 있으면 위치를 변경

# 객체조작

## 생성된 객체 추가/이동

구분	내용
<code>\$(선택자).append(생성객체)</code>	생성객체를 선택자의 마지막 자식으로 추가
<code>\$(선택자).prepend(생성객체)</code>	생성객체를 선택자의 첫번째 자식으로 추가
<code>\$(선택자).after(생성객체)</code>	생성객체를 선택자의 다음 형제로 추가
<code>\$(선택자).before(생성객체)</code>	생성객체를 선택자의 이전 형제로 추가

※ 만일 추가되는 객체가 문서에 있으면 위치를 변경

# 객체조작

## clone()

- html 내부에 있는 요소를 복사하는 메소드

```
$(선택자).clone();
```

