

# 문서 객체 탐색

강사 : 강병준

# jQuery 필터 선택자

## ❖ 위치와 관련된 필터 선택자

- 선택자 사용해 문서 객체 선택

위치 지정

선택자를 사용해 위치 지정

---

선택자 형태	설명
--------	----

---

요소:odd	홀번째에 위치한 문서 객체를 선택
요소:even	짝수번째에 위치한 문서 객체를 선택
요소:first	첫번째에 위치한 문서 객체를 선택
요소:last	마지막에 위치한 문서 객체를 선택

---

# 조건에 만족하는 첫 번째와 마지막 엘리먼트에 스타일을 적용하기

:first, :last

```
$('#tr:first')  
$('#tr:last')
```

페이지에서 처음 혹은 마지막으로 일치하는 엘리먼트를 검색

- `html()`
  - 태그를 포함한 텍스트를 얻는다.

```
<td>one</td>
```

- `text()`
  - 태그를 제외한 텍스트만 얻는다.

```
one
```

```
<!DOCTYPE html><html><head>
<meta charset="UTF-8">
<title>조건에 만족하는 첫 번째와 마지막 엘리먼트에 스타일을 적용하기-
:first, :last</title>
<style>
    table { table-layout:auto; width: 75%;}
    td { color: blue; font-weight: bold; }
    .selectedstyle {color: red; font-size: 20pt; background-color: pink;
}
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $('tr:first').addClass('selectedstyle');
        $('tr:last').addClass('selectedstyle');
        console.log("조건에 만족하는 첫번째 엘리먼트 : "
            + $('tr:first').html());
        console.log("조건에 만족하는 마지막 엘리먼트 : "
            + $('tr:last').html());
    });
</script></head><body>
    <table>
        <tr><td>one</td></tr>          <tr><td>two</td></tr>
        <tr><td>three</td></tr>        <tr><td>four</td></tr>
        <tr><td>five</td></tr>
    </table>
</body></html>
```

# map 지정 방식

## ■ css(map)

- map 지정 방식은

- { '속성명' : '속성값' , ... } 형식으로 스타일을 지정한다

```
$( 'div span:first-child, div span:last-child' ).css(  
  {  
    'text-decoration' : 'underline',  
    'color' : 'red',  
    'font-size' : '20pt '  
  } );
```

## 메소드 체인 방식

```
$( 'div span:first-child, div span:last-child' ).css('font-size', '20pt')  
  .css('color', 'red')  
  .css('text-decoration', 'underline');
```

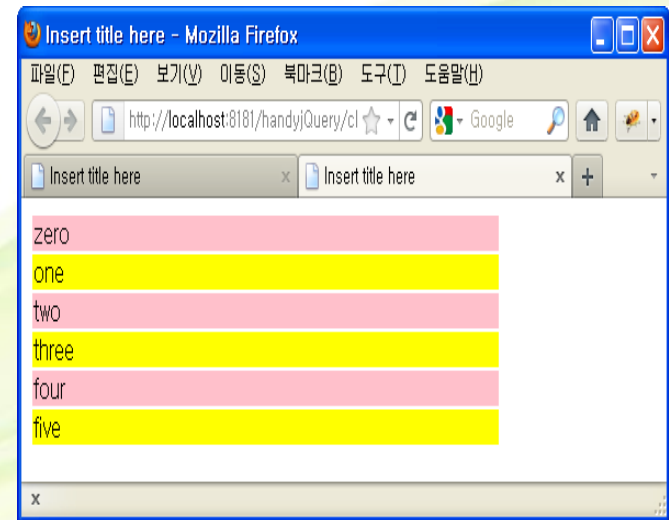
# 짝수와 홀수에 위치한 엘리먼트에 스타일을 적용하기

:odd, :even

```
$('#tr:even').css('backgroundColor', 'pink');  
$('#tr:odd').css('backgroundColor', 'yellow');
```

홀수나 짝수에 위치에 해당하는 엘리먼트를 선택  
인덱스가 0부터 시작, 첫 번째 엘리먼트가 짝수

- `css( ' 속성명 ' , ' 속성값 ' )`
  - 스타일을 적용한다.



```
<!DOCTYPE html>
<html><head>
<meta charset="UTF-8">
<title>조건에 만족하는 짝수와 홀수에 위치한 엘리먼트에 스타일을 적용하기-
:odd, :even</title>
<style>
    table {    table-layout:auto;    width: 75%;}
}
</style>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $('tr:even').css('backgroundColor', 'pink');
        $('tr:odd').css('backgroundColor', 'yellow');
    });
</script>
</head>
<body>
    <table>
        <tr><td>zero</td></tr>          <tr><td>one</td></tr>
        <tr><td>two</td></tr>          <tr><td>three</td></tr>
        <tr><td>four</td></tr>         <tr><td>five</td></tr>
    </table>
</body></html>
```

# jQuery 필터 선택자

필터 선택자를 사용한 테이블 줄무늬 부여 (1)

## ❖ body 태그 구성 : 줄무늬 모양의 표 만들기

locate.html

```
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    $(document).ready(function () {
      $('tr:odd').css('background', '#F9F9F9');
      $('tr:even').css('background', '#9F9F9F');
      $('tr:first').css('background', '#000000').css('color', '#FFFFFF');
    });
  </script></head><body>
  <table>
    <tr><th>이름</th><th>혈액형</th><th>지역</th></tr>
    <tr><td>강민수</td><td>AB형</td><td>서울특별시 송파구</td></tr>
    <tr><td>구지연</td><td>B형</td><td>미국 캘리포니아</td></tr>
    <tr><td>김미화</td><td>AB형</td><td>미국 메사추세츠</td></tr>
    <tr><td>김선화</td><td>O형</td><td>서울 강서구</td></tr>
    <tr><td>남기주</td><td>A형</td><td>서울 노랑진구</td></tr>
    <tr><td>윤하린</td><td>B형</td><td>서울 용산구</td></tr>
  </table>
</body>
```

이름	혈액형	지역
강민수	AB형	서울특별시 송파구
구지연	B형	미국 캘리포니아
김미화	AB형	미국 메사추세츠
김선화	O형	서울 강서구
남기주	A형	서울 노랑진구
윤하린	B형	서울 용산구



# jQuery 필터 선택자

## ❖ 함수 형태의 필터 선택자

jQuery가 제공하는 함수 필터 선택자

---

선택자 형태

설명

---

요소:contains(문자열) 특정 문자열을 포함하는 문서의 객체를 선택

요소:eq(n) n번째에 위치하는 문서 객체를 선택함

요소:gt(n) n번째 초과에 위치하는 문서 객체를 선택함

요소:has(h1) h1태그를 가지고 있는 문서 객체를 선택함

요소:lt(n) n번째 미만에 위치하는 문서 객체를 선택함

요소:not(선택자) 선택자와 일치하지 않는 문서 객체를 선택함

요소:nth-child(3n+1) 3n+1번째에 위치하는 문서 객체를 선택함

---

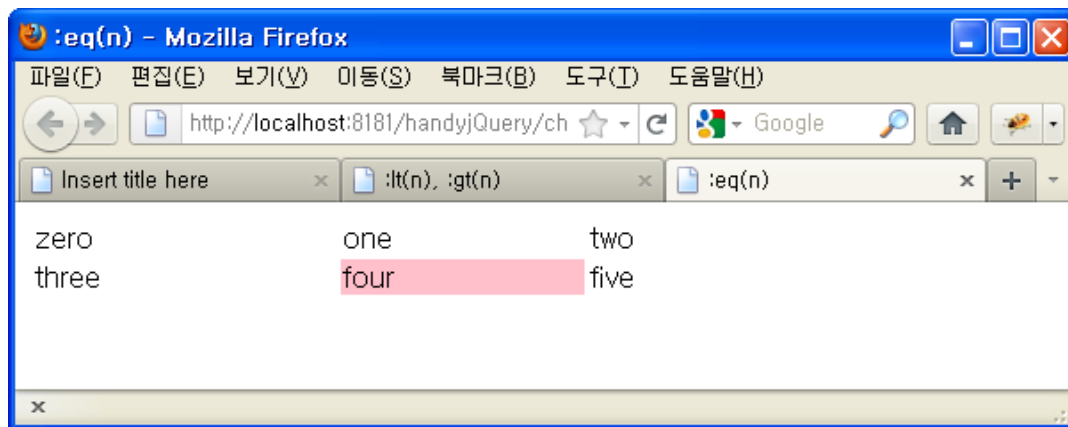
# n번째로 일치하는 엘리먼트에 스타일 적용하기

:eq

`$('td:eq(4)')`

n 번째 요소를 선택한다.

위치 기반 셀렉터에 해당하는 엘리먼트는 인덱스가 0부터 시작



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>n번째로 일치하는 엘리먼트에 스타일 적용하기-:eq(n)</title>
<style>
    table {    table-layout:auto;    width: 75%;}
}
</style>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $('td:eq(4)').css('background ', 'pink');
    });
</script>
</head>
<body>
    <table>
        <tr><td>zero</td><td>one</td><td>two</td></tr>
        <tr><td>three</td><td>four</td><td>five</td></tr>
    </table>
</body>
</html>
```

# n번째 엘리먼트 이후 혹은 이전에 위치한 엘리먼트에 스타일 해제하기

:lt(n), :gt(n)

```
$('#td:lt(2)').removeClass('textstyle');  
$('#td:gt(4)').removeClass('textstyle');
```

n 보다 작은, n 보다 큰 요소를 선택

- removeClass()
  - 인자로 지정된 스타일을 제거한다.



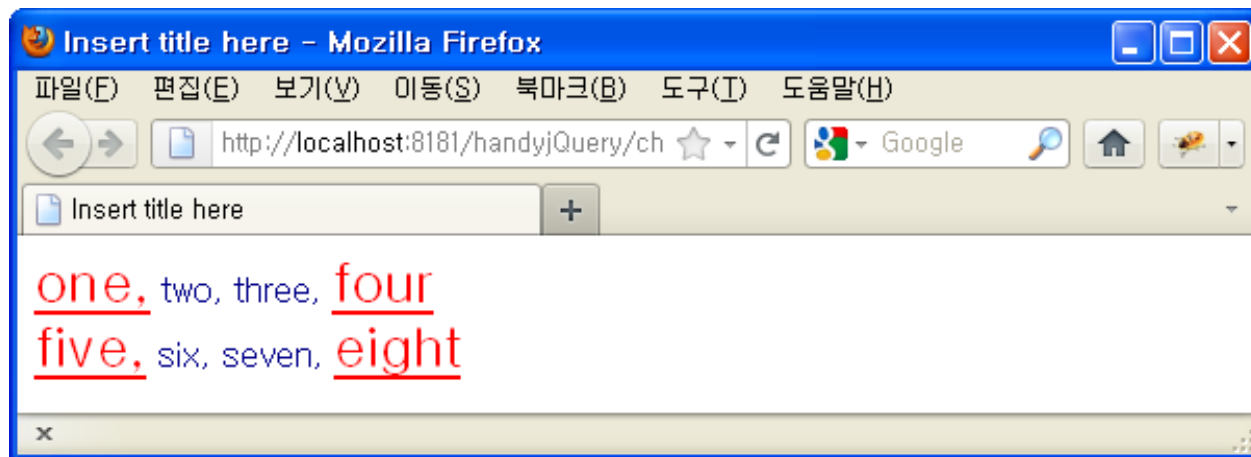
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> n번째 엘리먼트 이후 혹은 이전에 위치한 엘리먼트에 스타일 해제하기-
:gt(n),:lt(n)</title>
<style>
    table {    table-layout:auto;    width: 75%;}
    .textstyle { color: blue;    font-weight: bold; background-color: pink}
}
</style>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $('td').addClass('textstyle');
        $('td:lt(2)').removeClass('textstyle');
        $('td:gt(4)').removeClass('textstyle');
    });
</script>
</head>
<body>
    <table>
        <tr><td>zero</td><td>one</td><td>two</td></tr>
        <tr><td>three</td><td>four</td><td>five</td></tr>
    </table>
</body>
</html>
```

# 첫 번째와 마지막 자식 엘리먼트에 스타일 적용하기

:first-child, :last-child

```
$('#div span:first-child, div span:last-child')
```

부모 요소의 처음, 마지막 요소를 선택한다.



```
<!DOCTYPE html><html><head>
<meta charset="UTF-8">
<title>첫 번째와 마지막 자식 엘리먼트에 스타일 적용하기 :first-child, :last-
child</title>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $('div span:first-child, div span:last-child')
            .css('text-decoration', 'underline')
            .css('color', 'red')
            .css('font-size', '20pt');
    });
</script> </head> <body>
<div>
    <span>one,</span>
    <span>two,</span>
    <span>three,</span>
    <span>four</span>
</div>
<div>
    <span>five,</span>
    <span>six,</span>
    <span>seven,</span>
    <span>eight</span>
</div>
</body>
</html>
```

# jQuery 필터 선택자

그림 13-15 필터 선택자를 사용한 테이블 줄무늬 부여 (2)

이름	혈액형	지역
강민수	AB형	서울특별시 송파구
구지연	B형	미국 캘리포니아
김미화	AB형	미국 메사추세츠
김선화	O형	서울 강서구
남기주	A형	서울 노랑진구
윤하린	B형	서울 용산구

## ❖ 함수 형태의 필터 선택자

### ■ :nth-child() 필터 선택자

- 함수의 괄호 안에 식별자 n 사용해 문서 객체의 순서 나타냄
- 예제 코드 13-24의 첫 번째 nth-child() 필터 선택자
  - 3n+1번째에 위치하는 선택자 선택
    - » 1, 4, 7번째에 위치하는 문서 객체 선택

locate2.html

```
<script src="http://code.jquery.com/jquery-1.7.js"></script>
<script>
    $(document).ready(function () {
        $('tr:eq(0)').css('background', '#000000').css('color', 'White');
        $('td:nth-child(3n+1)').css('background', '#565656');
        $('td:nth-child(3n+2)').css('background', '#A9A9A9');
        $('td:nth-child(3n)').css('background', '#F9F9F9');
    });
</script>
```



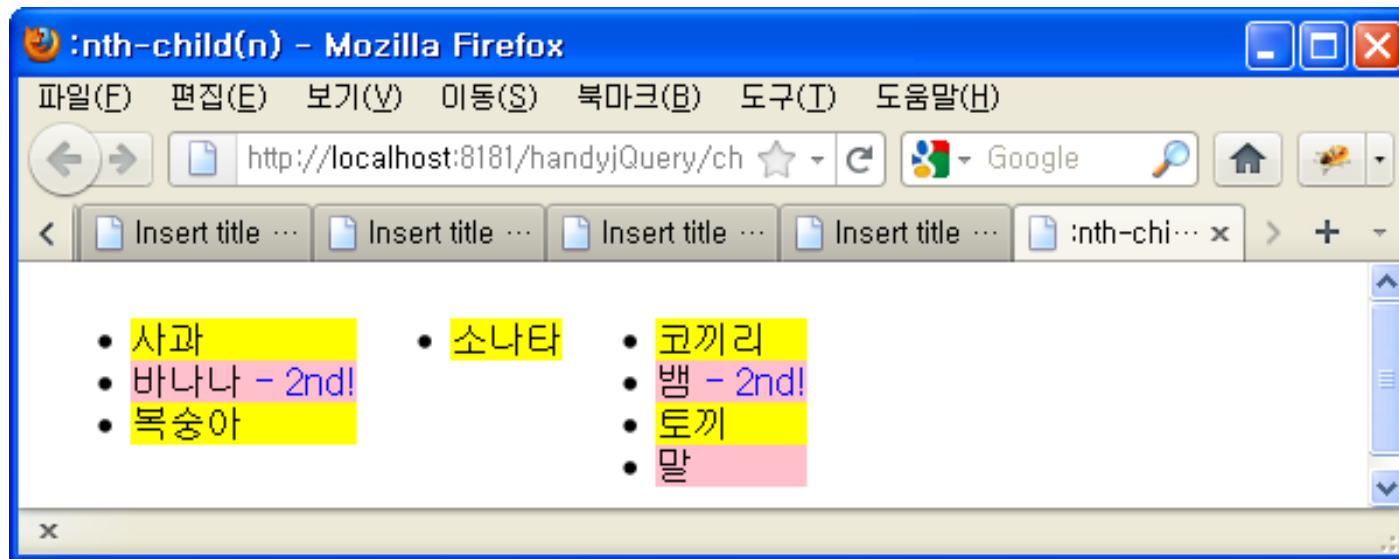
```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
<script type="text/javascript" src="http://code.jquery.com/jquery-1.7.js"></script>
<script type="text/javascript">
$(function(){
    $('tr:eq(0)').css('background', '#000000').css('color', 'White');
    $('td:nth-child(3n+1)').css('background', '#565656');
    $('td:nth-child(3n+2)').css('background', '#A9A9A9');
    $('td:nth-child(3n)').css('background', '#F9F9F9');
});
</script></head>
<body>
<table>
<tr><th>이름</th><th>혈액형</th><th>지역</th></tr>
<tr><td>강민수</td><td>AB형</td><td>서울특별시 송파구</td></tr>
<tr><td>구지연</td><td>B형</td><td>미국 캘리포니아</td></tr>
<tr><td>김미화</td><td>AB형</td><td>미국 메사추세츠</td></tr>
<tr><td>김선화</td><td>O형</td><td>서울 강서구</td></tr>
<tr><td>남기주</td><td>A형</td><td>서울 노량진구</td></tr>
<tr><td>윤하린</td><td>B형</td><td>서울 용산구</td></tr>
</table>
</body></html>

```

# n번째 혹은 짝수 또는 홀수 자식 엘리먼트 검색하기

:nth-child(n), :nth-child(even), :nth-child(odd)



# n번째 혹은 짝수 또는 홀수 자식 엘리먼트 검색하기

:nth-child(n), :nth-child(even), :nth-child(odd)

```
$( "ul li:nth-child(2)" ).append( "<span> - 2nd!</span>" );  
$( "ul li:nth-child(even)" ).css( "background", "pink" );  
$( "ul li:nth-child(odd)" ).css( "background", "yellow" );
```

인덱스가 1부터 시작한다.

:nth-child()는 부모 요소의 n 번째 자식 요소를 선택한다.

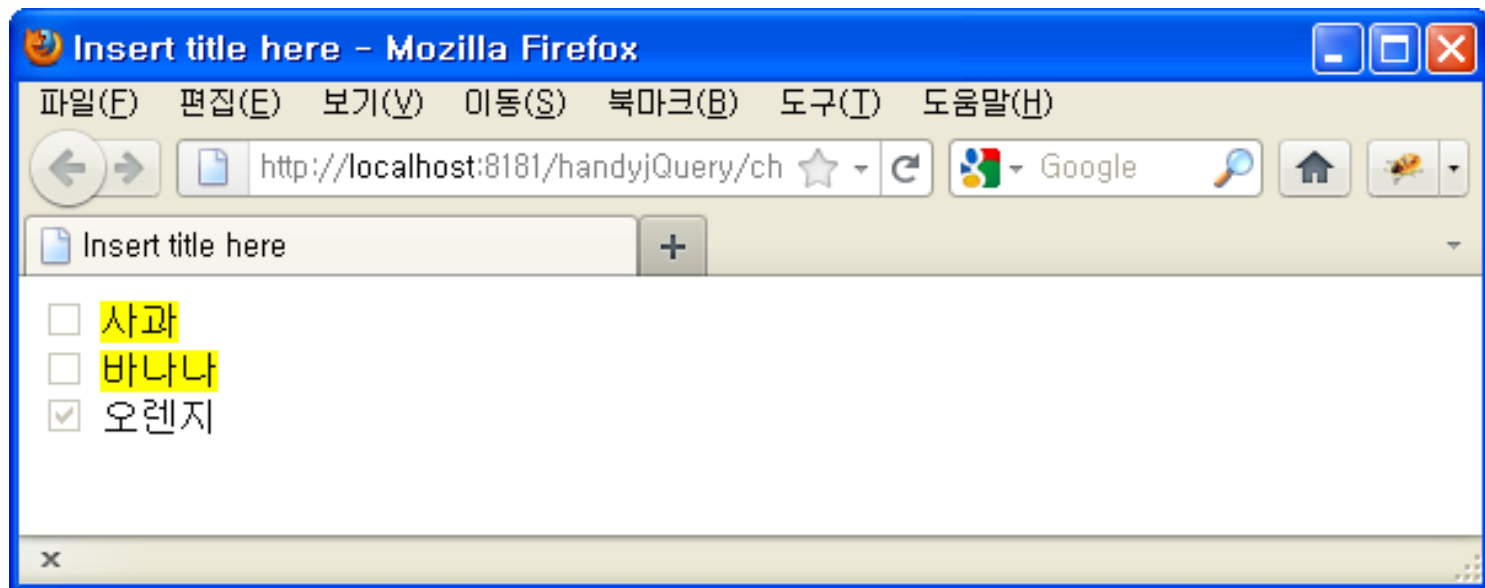
:nth-child(even)는 부모 요소의 **짝수** 번째 자식 요소를 선택한다.

:nth-child(odd)는 부모 요소의 **홀수** 번째 자식 요소를 선택한다.

```
<!DOCTYPE html><html><head>
<meta charset= "euc-kr">
<title>n번째 혹은 짝수 또는 홀수 자식 엘리먼트 검색하기-:nth-child(n), :nth-
child(even), :nth-child(odd)</title>
<style>
div { float: left; }
span { color: blue; }
</style>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function() {
    $("ul li:nth-child(2)").append("<span> - 2nd!</span>");
    $("ul li:nth-child(even)").css("backgroundColor", "pink");
    $("ul li:nth-child(odd)").css("backgroundColor", "yellow");
});
</script></head><body>
    <div>
        <ul>
            <li>사과</li>
            <li>바나나</li>
            <li>복숭아</li>
        </ul>
    </div>
    <div>
        <ul>
            <li>소나타</li>
        </ul>
    </div>
    <div>
        <ul>
            <li>코끼리</li>
            <li>뱀</li>
            <li>토끼</li>
            <li>말</li>
        </ul>
    </div>
</body></html>
```

# 조건에 반대되는 엘리먼트 검색하기

:not()



# 조건에 반대되는 엘리먼트 검색하기

:not()

```
$( 'input:not(:checked)' )
```

인자로 주어진 조건에 반대되는 엘리먼트를 추출한다.

:checked

jQuery 정의 셀렉터인 :checked는 일치하는 엘리먼트 중 선택 상태인 엘리먼트만 필터링한다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>조건에 반대되는 엘리먼트 검색하기-:not()</title>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(
        function() {
            $('input:not(:checked) + span').css('background-color', 'yellow');
        });
</script>
</head>
<body>
    <div>
        <input type="checkbox" name="fruit" /> <span>사과</span>
    </div>
    <div>
        <input type="checkbox" name="fruit" /> <span>바나나</span>
    </div>
    <div>
        <input type="checkbox" name="fruit" checked="checked" /> <span>오렌지</span>
    </div>
</body>
</html>
```

# 기본 필터링 메서드

## ❖ filter() 메서드 사용법 예제

- h3 태그에서 홀수 번째에 위치하는 문서 객체 선택 filter01
  - 홀수 번째에 위치하는 h3 태그의 스타일 속성 변경 (선택자)

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    $(document).ready(function () {
      $('h3:even').css({
        backgroundColor: 'Black',
        color: 'White'
      });
    });
  </script>
</head>
<body>
  <h3>Header-0</h3>    <h3>Header-1</h3>
  <h3>Header-2</h3>    <h3>Header-3</h3>
  <h3>Header-4</h3>
  <h3>Header-5</h3>
</body>
</html>
```



# 기본 필터링 메서드

## ❖ 기본 필터링 메서드

- jQuery의 선택자를 사용하면 원하는 문서 객체 대부분 선택 가능
- 기본 지원하지 않는 필터를 사용해 문서 객체를 선택해야 한다면?
  - 메서드 사용
  - 이 절에서는 `filter()` 메서드 사용법 연습

### jQuery 메서드

메서드 이름	설명
<code>filter()</code>	문서 객체를 필터링합니다.

`filter()` 메서드는 다음과 같은 두 가지 형태로 사용합니다.

```
1 $(selector).filter(selector);
```

```
2 $(selector).filter(function () { });
```

# 기본 필터링 메서드

## ❖ filter() 메서드 사용법 예제 filter02

- 1번 형태의 filter() 메서드 사용
  - filter() 메서드의 매개 변수에 선택자 입력
  - ‘:even’ 을 입력할 수도 있음

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="http://code.jquery.com/jquery-1.7.js"></script>
```

```
<script>
```

```
$(document).ready(function () {  
    $('h3').filter(':even').css({  
        backgroundColor: 'Black',  
        color: 'White'  
    });  
});
```

```
</script>
```

```
</head>
```

# 기본 필터링 메서드

## ❖ filter() 메서드 사용법 예제

### ■ 2번 형태의 filter() 메서드를 사용

- filter() 메서드의 매개 변수에 함수를 넣음
- 입력하는 함수는 매개 변수로 index를 가짐
- 함수에서 리턴하는 값에 따라 문서 객체 선택

filter() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

**Header-3**

**Header-4**

**Header-5**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="http://code.jquery.com/jquery-1.7.js"></script>
```

```
<script>
```

```
$(document).ready(function () {  
    $('h3').filter(function (index) {  
        return index % 3 == 0;  
    }).css({  
        backgroundColor: 'Black',  
        color: 'White'  
    });  
});
```

```
</script>
```

```
</head>
```

# 문서 객체 탐색 종료

## ❖ 예제 코드

- 한 줄로 서로 다른 문서 객체에 스타일 적용 가능 filter04

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').css('background', 'Orange').filter(':even').css('color',
'Red');
    });
  </script>
</head>
<body>
  <h1>Header-0</h1>
  <h1>Header-1</h1>
  <h1>Header-2</h1>
</body>
</html>
```

# 문서 객체 탐색 종료

- filter() 메서드 제거하려면 메서드 사용 filter05

```
<script src="http://code.jquery.com/jquery-1.7.js"></script>
<script>
    $(document).ready(function () {
        $('h1').css('background', 'Orange');
        $('h1:even').css('color', 'White');
        $('h1:odd').css('color', 'Red');
    });
</script>
```

end()메서드 : 문서객체 선택을 한 단계 뒤로 돌립니다 filter06

```
<script>
    $(document).ready(function () {
        $('h1').css('background', 'Orange').filter(':even').css('color',
'White').end().filter(':odd').css('color', 'Red');
    });
</script>
```

# 특정 위치의 문서 객체 선택

## ❖ 필터 선택자 이용

- 특정 위치에 존재하는 문서 객체 선택
- 필터 선택자는 자주 사용하므로 메서드로 제공

jQuery method

---

메서드	설명
-----	----

---

eq()	특정 위치에 존재하는 문서객체를 선택
first()	첫 번째에 위치하는 문서객체를 선택
last()	마지막에 위치하는 문서객체를 선택

---

```
<body>
  <div>
    <h1>Header-0</h1>
    <h1>Header-1</h1>
    <h1>Header-2</h1>
  </div>
</body>
```

# 특정 위치의 문서 객체 선택

## ❖ eq() 메서드

- 매개 변수에 숫자 입력 eq.html
  - 양수를 입력할 수도 있지만, 음수를 입력할 수도 있음
  - 음수를 입력하면 뒤쪽 기준으로 선택
  - 맨 뒤의 문서 객체가 -1번째

eq() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="http://code.jquery.com/jquery-1.7.js"></script>
```

```
<script>
```

```
$(document).ready(function () {
```

```
    $('h1').eq(0).css('background', 'Orange');
```

```
    $('h1').eq(-1).css('background', 'Red');
```

```
});
```

```
</script>
```

```
</head>
```

# 문서 객체 추가 선택

## ❖ add() 메서드

- jQuery는 문서 객체의 체이닝을 더 유연하게 도와줌
- 메서드 제공

add() 메서드 : 문서 객체를 추가로 선택합니다

```
<body>
  <h1>Header-0</h1>
  <h2>Header-1</h2>
  <h1>Header-2</h1>
  <h2>Header-3</h2>
  <h1>Header-4</h1>
</body>
</html>
```



# 문서 객체 추가 선택

add() 메서드의 사용

## ❖ 예제 코드 add.html

Header-0Header-1Header-2Header-3Header-4

- add() 메서드 활용 예제
- h1 태그의 background 스타일 속성에 Gray 입력
- h2 태그를 추가로 선택, float 스타일 속성에 left 입력

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
```

```
  <script>
```

```
    $(document).ready(function () {
```

```
      $('h1').css('background',
```

```
'Gray').add('h2').css('float', 'left');
```

```
    });
```

```
  </script>
```

```
</head>
```

# 문서 객체의 특징 판별

## ❖ 문서 객체가 특징을 가지고 있는지 판단

- is() 메서드는 매개 변수로 선택자 입력
- 선택한 객체가 선택자와 일치하는지 판별해 불리언 리턴

### jQuery 메서드

메서드 이름	설명
is()	문서 객체의 특징을 판별합니다.

```
<body>
  <h1 class="select">Header-0</h1>
  <h1>Header-1</h1>
  <h1 class="select">Header-2</h1>
</body>
</html>
```

# 문서 객체의 특징 판별

- h1 태그 중 select 클래스 속성이 있는 문서 객체에 스타일 적용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-
1.7.js"></script>
  <script>
    $(document).ready(function () {
      $('h1').each(function () {
        if ($(this).is('.select')) {
          $(this).css('background', 'Orange');
        }
      });
    });
  </script>
</head>
```

is() 메서드의 사용

**Header-0**

**Header-1**

**Header-2**

# 특정 태그 선택

## ❖ 문서 객체에서 특정 태그를 선택하는 방법

- XML 문서에서 데이터를 추출시 많이 사용하므로 예제 이해하도록
- 특정 태그를 선택할 때 메서드 사용

find() 메서드 : 특정태그를 선택합니다

```
<body>
```

```
</body>
```

```
</html>
```

# 특정 태그 선택

## ❖ XML 문서에서 데이터를 추출하는 예제

- find() 메서드 사용
- parseXML() 메서드를 사용해 문자열을 XML 문서 객체로 변경
- 인터넷 익스플로러에서는 문제가 발생
  - parseXML() 메서드 사용

# 특정 태그 선택

## ❖ 문자열에 XML 문서 입력

```
<script src="http://code.jquery.com/jquery-1.7.js"></script>
<script>
    // 변수를 선언합니다.
    var xml = '';
    xml += '<friends>';
    xml += '    <friend>';
    xml += '        <name>연하진</name>';
    xml += '    </friend>';
    xml += '    <friend>';
    xml += '        <name>윤명월</name>';
    xml += '        <language>Basic</language>';
    xml += '    </friend>';
    xml += '    <friend>';
    xml += '        <name>윤하린</name>';
    xml += '        <language>C#</language>';
    xml += '    </friend>';
    xml += '</friends>';

    $(document).ready(function () {
        // 변수를 선언합니다.
        var xmlDoc = $.parseXML(xml);
    });
</script>
```

# 특정 태그 선택

- `$()` 메서드의 매개 변수에 생성한 XML 문서 객체 입력
- `$()` 메서드에는 문서 객체를 곧바로 넣을 수 있음
- `find()` 메서드를 사용해 friend 태그 선택
  - friend 태그는 총 세 개이므로 `each()` 메서드 사용

```
$(document).ready(function () {  
    // 변수를 선언합니다.  
    var xmlDoc = $.parseXML(xml);  
    $(xmlDoc).find('friend').each(function (index) {  
  
        });  
    });  
</script>
```

# 특정 태그 선택

- . each() 메서드 안에서 find() 메서드를 한 번 더 사용해 데이터 추출
- . text() 메서드를 사용하면 태그 내부 글자를 가져올 수 있음
- . 각각의 데이터로 HTML 태그를 만든 후 innerHTML 속성에 추가

```
$(document).ready(function () {  
    // 변수를 선언합니다.  
    var xmlDoc = $.parseXML(xml);  
    $(xmlDoc).find('friend').each(function (index) {  
        // 변수를 선언합니다.  
        var output = '';  
        output += '<div>';  
        output += '    <h1>' + $(this).find('name').text() + '</h1>';  
        output += '    <p>' + $(this).find('language').text() + '</p>';  
        output += '</div> ';  
        // 출력합니다.  
        document.body.innerHTML += output;  
    });  
});  
</script>
```

```
<head>  
    <style>  
        div  
        {  
            margin:10px;  
            border:3px Solid Black;  
            border-radius:10px;  
            float:left;  
            width:120px; height:120px;  
            text-align:center;  
        }  
    </style>
```

XML 파싱을 사용한 문서 객체 생성

연하진

Ruby

윤명월

Basic

윤하린

C#