



Chapter 06-2

선택자와 필터

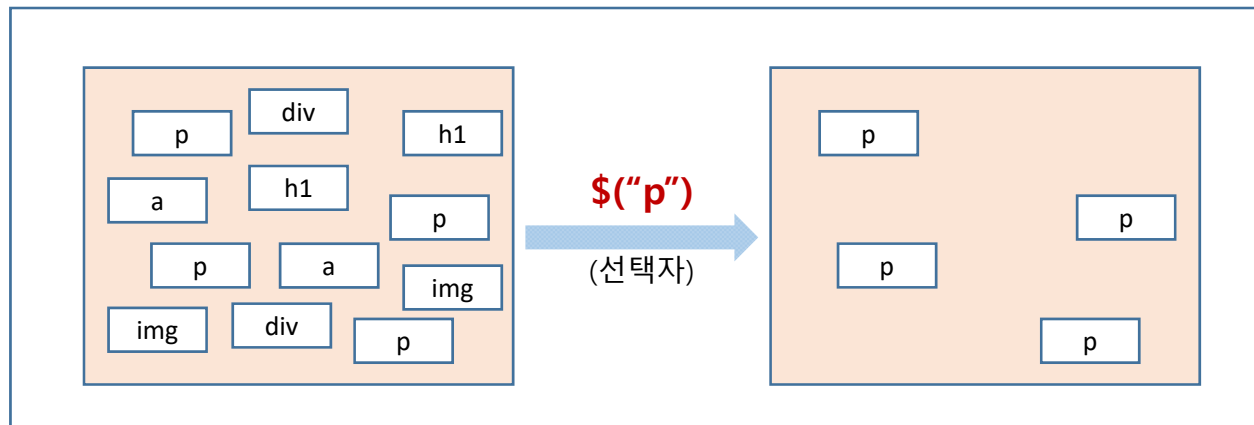


기본 선택자 (basic selectors)

선택자의 개요

■ 선택자 (selectors)

- 자바스크립트 라이브러리의 효과를 지정하기 위해 HTML의 DOM 요소를 선택하는 역할 수행
 - jQuery는 선택자를 사용해 HTML 문서 내에 있는 특정 요소를 선택
 - 기본적으로 CSS에서의 선택자와 같은 의미
- 선택자에 의해 선택된 요소는 jQuery 객체를 형성함
 - 선택된 요소(jQuery 객체)에는 jQuery가 제공하는 속성이나 메서드를 적용할 수 있음
 - 따라서 원하는 노드를 정확하게 찾는 것이 필수적임



선택자의 개요

■ 선택자의 형식

```
$( '선택자' )
```

• 선택자 사용의 예

```
$( 'h1' ).css( 'color', 'red' )
```

- **\$ ()**
 - jQuery 객체를 생성하는 함수로 jQuery() 함수로도 표현 가능
 - \$('h1').css('color', 'red') = jQuery('h1').css('color', 'red')
- **h1**
 - 선택자
 - 위의 예에서는 본문 내에 존재하는 모든 <h1> 태그를 의미함
- **css('color', 'red')**
 - 지정한 선택자에 효과를 적용하는 메서드 (글자 색을 red로 지정)

선택자의 개요

■ 기본 선택자의 종류

• 직접 선택자

- 요소를 직접 지정하여 선택하는 선택자
- 전체 선택자, 아이디 선택자, 클래스 선택자 등

• 인접 관계 선택자

- 특정 요소를 선택한 후, 그 요소를 기준으로 원하는 요소를 선택하는 선택자
- 자식 요소 선택자, 자손 선택자, 형제 선택자 등

직접 선택자

■ 전체 선택자

- 문서 내의 모든 문서 객체를 선택

`$('*')`

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("*").css("border","1px solid blue");
    });
  </script>
</head>
<body>
  <h1>제이쿼리</h1>
  <h2>선택자</h2>
  <h3>직접 선택자</h3>
</body>
</html>
```

6-selector-all.htm (p.172)

제이쿼리

선택자

직접 선택자

직접 선택자

■ 아이디 선택자

```
$("#아이디명")
```

- 특정한 id 속성 가진 문서 객체를 선택하는 선택자
 - 선택자로 id 이름을 지정할 경우 '#'기호를 아이디 이름 앞에 표현해야 함
 - HTML 문서 내에 지정된 아이디의 요소를 선택

■ 클래스 선택자

```
$(".클래스명")
```

- 특정한 class 속성 가진 문서 객체를 선택하는 선택자
 - 선택자로 클래스 이름을 지정할 경우 '.'기호를 클래스 이름 앞에 표현해야 함
 - HTML 문서 내에 지정된 클래스의 모든 요소를 선택

직접 선택자

■ 클래스 선택자

```
$(".클래스명")
```

- 특정한 class 속성 가진 문서 객체를 선택하는 선택자
 - 선택자로 클래스 이름을 지정할 경우 "."기호를 클래스 이름 앞에 표현해야 함
 - HTML 문서 내에 지정된 클래스의 모든 요소를 선택

■ 아이디 선택자

```
$("#아이디명")
```

- 특정한 id 속성 가진 문서 객체를 선택하는 선택자
 - 선택자로 id 이름을 지정할 경우 '#'기호를 아이디 이름 앞에 표현해야 함
 - HTML 문서 내에 지정된 아이디의 요소를 선택

직접 선택자

6-selector-id.htm (p. 173)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $("#tit").css("background-color", "#ff0").css("border", "2px solid #f00");
    });

  </script>
</head>
<body>
  <h1>제이쿼리</h1>
  <h2 id="tit">선택자</h2>
  <h3>직접 선택자</h3>
</body>
</html>
```

제이쿼리

선택자

직접 선택자

직접 선택자

6-selector-class.htm (P.174)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $(".tit").css("background-color", "#ff0").css("border", "2px dashed #f00");
    });

  </script>
</head>
<body>
  <h1>제이쿼리</h1>
  <h2 class="tit">선택자</h2>
  <h3>직접 선택자</h3>
</body>
</html>
```

제이쿼리

선택자

직접 선택자

직접 선택자

■ 요소 선택자

• 선택자로 HTML의 태그를 사용

- 현재 HTML문서 내에 존재하는 모든 요소 중 인자로 지정된 요소들을 모두 선택

`$('태그')`

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("h2").css("background-color","#0ff").css("border", "2px dashed #f00");
    });
  </script>
</head>
<body>
  <h1>제이쿼리</h1>
  <h2>선택자</h2>
  <h3>직접 선택자</h3>
</body>
</html>
```

6-selector-tag.htm (p. 175)

제이쿼리

선택자

직접 선택자

직접 선택자

■ 그룹 선택자

- 하나 이상의 태그를 동시에 선택하고자 할 경우 사용
 - 선택자들은 '콤마(,)로 구분함

`$("선택자1, 선택자2, ... , 선택자n")`

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("h1, #tit3").css("background-color", "#0ff").css("border", "2px dashed #f00");
    });
  </script>
</head>
<body>
  <h1>제이쿼리</h1>
  <h2>선택자</h2>
  <h3 id="tit3">직접 선택자</h3>
  <h3>인접 선택자</h3>
</body>
</html>
```

6-selector-group.htm (p. 176)

제이쿼리

선택자

직접 선택자

인접 선택자

직접 선택자

■ 종속 선택자

- 선택한 요소의 id 또는 class 값과 일치하는 요소를 선택

`$("선택자#아이디명")` 또는 `$("선택자.클래스명")`

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("h1.tit").css("background-color","#0ff")
      .css("border", "2px dashed #f00");
    });
  </script>
</head>
<body>
  <h1 class="tit">제이쿼리</h1>
  <h2>선택자</h2>
  <h3 class="tit">직접 선택자</h3>
</body>
</html>
```

6-selector-depend.htm (p. 177)

제이쿼리

선택자

직접 선택자

인접 관계 선택자

■ 자식 요소 선택자

- 부모 요소의 자식 요소 중 특정 자식 요소를 선택
 - 자식 요소의 하위 요소(후손 요소)는 선택하지 않음

```
$( '부모 요소 > 자식 요소' )
```

- [예] `$(div > p)`
 - `<div>` 태그의 자식 노드 중 `<p>` 태그를 선택 (자손은 선택하지 않음)

■ 후손(하위) 요소 선택자

- 특정 요소의 후손 요소를 선택 (자식을 포함해 모든 요소를 선택)

```
$( '부모 요소 자식 요소' )
```

- [예] `$(div p)`
 - `<div>` 태그 내에 있는 모든 `<p>` 태그를 모두 선택함

인접 관계 선택자

6-selector-child.htm (p. 181)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#wrap > h1").css("border","2px dashed #f00");
      $("#wrap > section").children( )
        .css({
          "background-color":"yellow",
          "border":"2px solid #f00"
        });
    });
  </script>
</head>
<body>
  <div id="wrap">
    <h1>인접 관계 선택자</h1>
    <p>내용1</p>
    <section>
      <h1>자식 요소 선택자</h1>
      <p>내용2</p>
    </section>
  </div>
</body>
</html>
```

인접 관계 선택자

내용1

자식 요소 선택자

내용2

인접 관계 선택자

6-selector-descendant.htm (p. 180)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#wrap h1")
      .css({
        "background-color":"yellow",
        "border":"2px dashed #f00"
      });
    });
  </script>
</head>
<body>
  <div id="wrap">
    <h1>인접 관계 선택자</h1>
    <p>내용1</p>
    <section>
      <h1>하위 요소 선택자</h1>
      <p>내용2</p>
    </section>
  </div>
</body>
</html>
```

인접 관계 선택자

내용1

하위 요소 선택자

내용2

인접 관계 선택자

■ 인접 형제 요소 선택자

- [기준 요소]를 기준으로 바로 우측에 인접한 하나의 [다음 요소]를 선택

`$('기준요소 + 다음요소')`

- [다음 요소]로 지정된 노드가 일치하지 않을 경우 선택되지 않음
- [예] `$('#first + li')`
 - id 속성이 first인 노드를 찾고 바로 다음에 나타나는 형제 노드를 선택
 - 바로 다음 노드가 가 아닐 경우 선택하지 않음

인접 관계 선택자

■ 형제 요소 선택자

- [기준 요소]를 기준으로 우측에 있는 모든 [다음 요소]를 선택

`$('기준요소 ~ 다음요소')`

- [다음요소]로 지정된 노드가 일치하지 않을 경우 선택되지 않음
- [예] `$('#first ~ li')`
 - id 속성이 first인 노드를 찾고 다음에 나타나는 모든 ``를 선택 (단, 부모가 같아야 함)
 - 다음 노드가 ``가 아닐 경우 선택하지 않음

인접 관계 선택자

6-selector-sib.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#one + li").css({backgroundColor:"yellow"});
      $("#one ~ li").css( {color:"red"} );
    });
  </script>
</head>
<body>
  <h2> Welcome mySite </h2>
  <ul class="list">
    <li id="one">One</li>
    <li id="two">Two</li>
    <ul>
      <li id="two-1">two-1</li>
      <li id="two-2">two-2</li>
      <li id="two-3">two-3</li>
    </ul>
    <li id="three">Three</li>
  </ul>
</body>
</html>
```

Welcome mySite

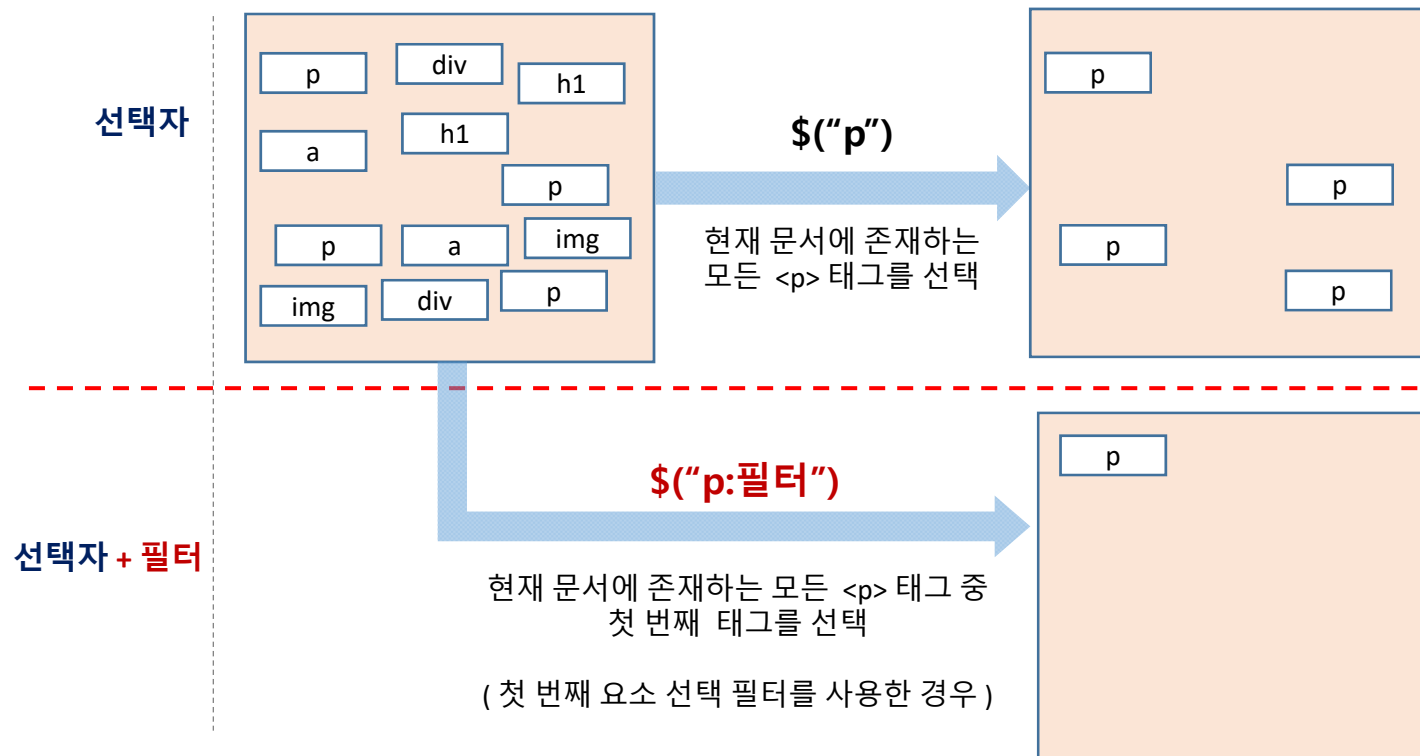
- One
- Two
 - two-1
 - two-2
 - two-3
- Three

필터(filter) 선택자

필터의 개요

■ 필터(filter)

- 선택자를 사용하는 것보다 정교하게 선택할 수 있도록 jQuery가 제공하는 선택자
 - 선택자와 함께 사용됨
 - 선택자를 사용해 1차 선택한 후, 선택된 내용을 대상으로 필터를 사용해 2차 선택 수행



필터의 개요

■ 필터의 형식과 종류

• 필터의 형식

\$("선택자:필터")

- 선택자에 의해 선택된 요소 중 필터에 해당하는 요소를 추가적으로 선택
- 선택자와 필터의 구분은 : (콜론)을 사용

• 필터의 종류

- 위치 관련 필터
- 자식 요소 관련 필터
- 내용 관련 필터

위치 관련 필터

■ 위치 관련 필터

- 요소의 위치를 참조해 요소를 선택하는 필터

구분	필터	설명
함수형	요소:eq(n)	위치 값이 n인 문서 객체를 선택
	요소:gt(n)	위치 값이 n보다 큰 문서 객체를 선택
	요소:lt(n)	위치 값이 n보다 작은 문서 객체를 선택
	요소:not(선택자)	인자인 선택자에 의해 선택된 문서 객체를 제외한 모든 문서 객체를 선택
필터형	요소:odd	위치 값이 홀수인 문서 객체를 선택
	요소:even	위치 값이 짝수인 문서 객체를 선택 (인덱스 값이 0인 경우 포함)
	요소:first	첫 번째 위치한 문서 객체를 선택
	요소:last	마지막에 위치한 문서 객체를 선택

위치 관련 필터

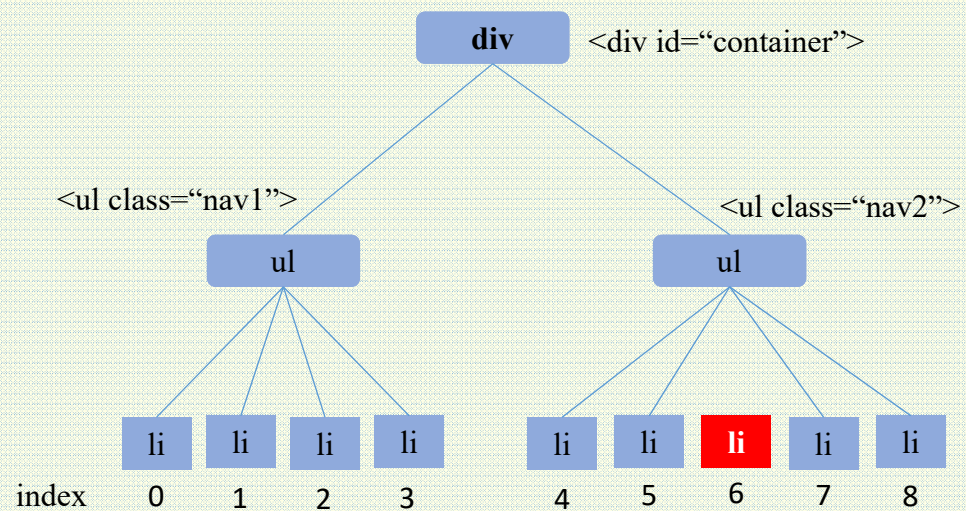
■ eq() 필터

`$('요소:eq(index)')`

- 선택된 요소들 중 index의 위치에 있는 요소를 선택
- index는 0부터 시작함

예 `$("div#container li:eq(6)")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

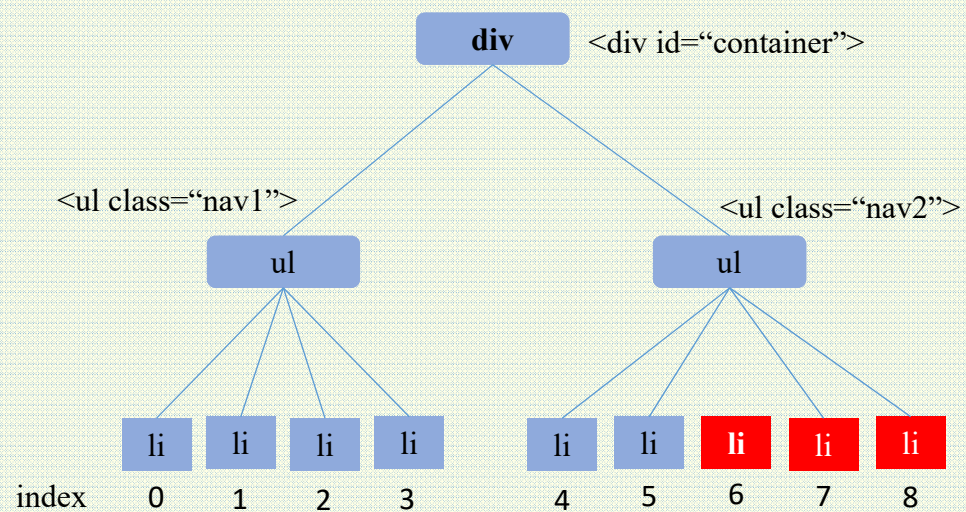
■ gt() 필터

`$('요소:gt(index)')`

- 선택된 요소들 중 인자로 지정된 index보다 큰 index를 가진 요소들을 모두 선택
- 자신은 포함되지 않음

예 `$("div#container li:gt(5)")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

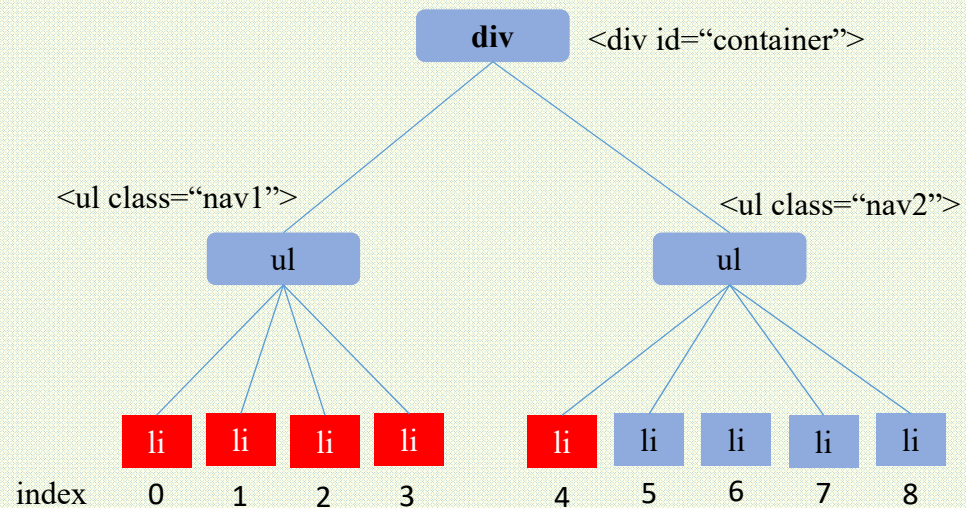
■ gt() 필터

`$('요소:lt(index)')`

- 선택된 요소들 중 인자로 지정된 index보다 작은 index를 가진 요소들을 모두 선택
- 자신은 포함되지 않음

예 `$("div#container li:lt(5)")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

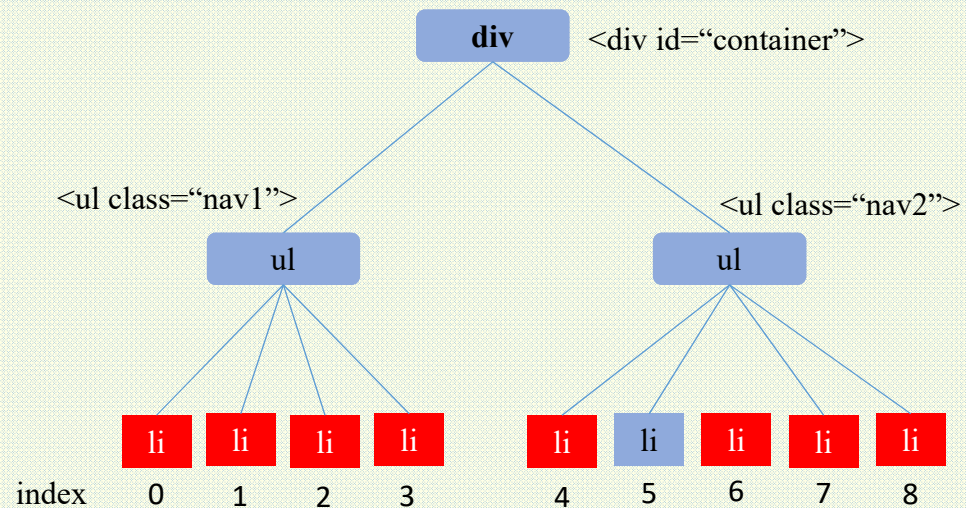
■ not() 필터

`$('요소:not(index)')`

- 선택된 요소들 중 인자로 지정된 index 가진 요소를 제외한 모든 모두 선택
- index는 필터를 사용해야 함

예 `$("div#container li:not(:eq(5))")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

6-filter-eq-gt-lt-not.htm (p.193)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#menu li").eq(2).css( {"background-color":"#ff0"} );
      $("#menu li:lt(2)").css( {"background-color":"#0ff"} );
      $("#menu li:gt(2)").css( {"background-color":"#f0f"} );
      $("#menu li:not(:eq(2))").css( {"color":"yellow"} );

    });
  </script>
</head>
<body>
  <h1>탐색 선택자</h1>
  <ul id="menu">
    <li>내용1</li>
    <li>내용2</li>
    <li>내용3</li>
    <li>내용4</li>
    <li>내용5</li>
  </ul>
</body>
</html>
```

탐색 선택자

- 내용1
- 내용2
- 내용3
- 내용4
- 내용5

위치 관련 필터

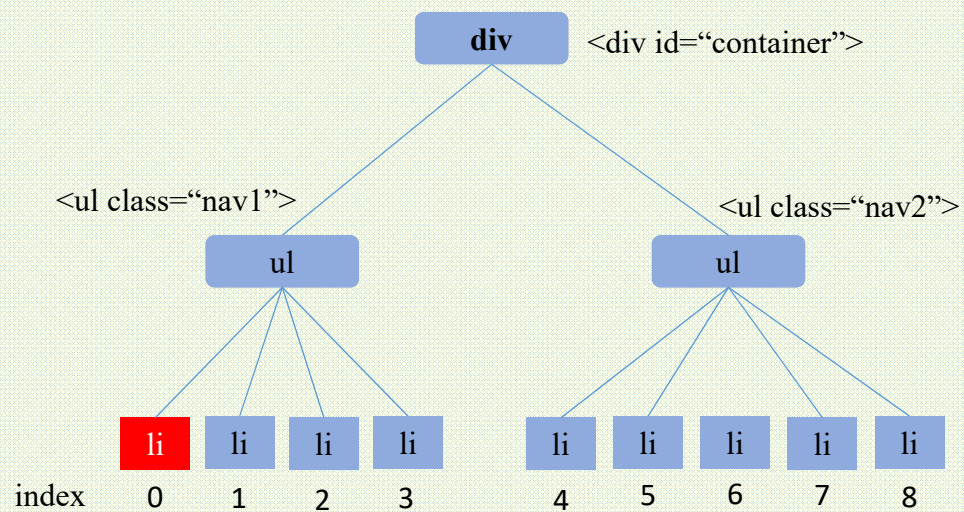
■ :first 필터

`$('요소:first')`

- 선택된 요소들 중 첫 번째 요소를 선택

예 `$("div#container li:first")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

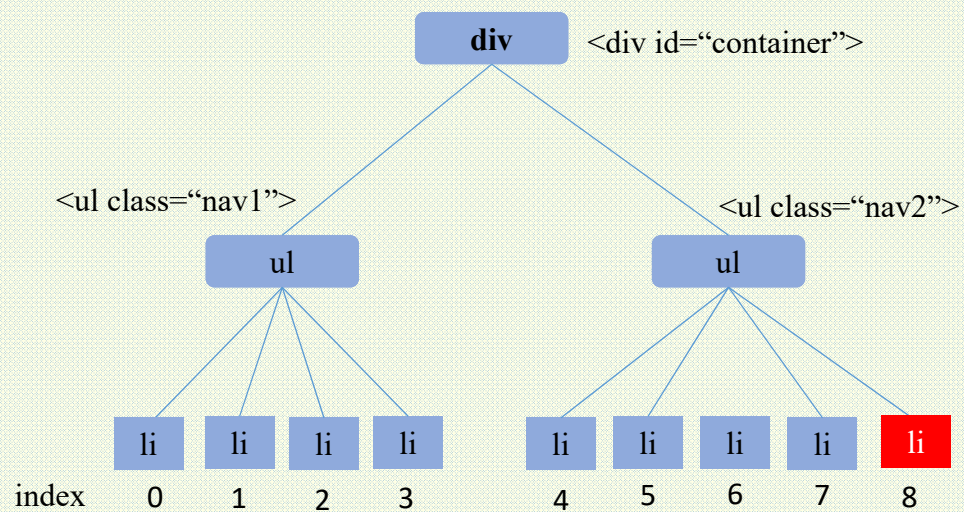
■ :last 필터

`$('요소:last')`

- 선택된 요소들 중 마지막 요소를 선택

예 `$("div#container li:last")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

6-filter-first-last.htm (p.191)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#menu li:first")
      .css({"background-color":"#ff0"});

      $("#menu li:last")
      .css({"background-color":"#0ff"});

    });
  </script>
</head>
<body>
  <h1>탐색 선택자</h1>
  <ul id="menu">
    <li>내용1</li>
    <li>내용2</li>
    <li>내용3</li>
    <li>내용4</li>
  </ul>
</body>
</html>
```

탐색 선택자

- 내용1
- 내용2
- 내용3
- 내용4

위치 관련 필터

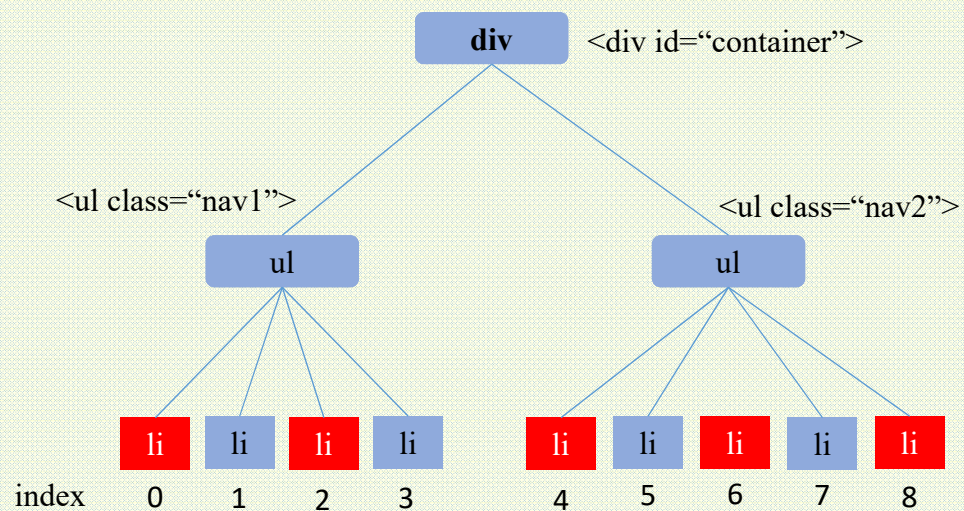
■ :even 필터

`$('요소:even')`

- 선택된 요소들 중 index가 짝수 번째 요소를 선택

예 `$("div#container li:even")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

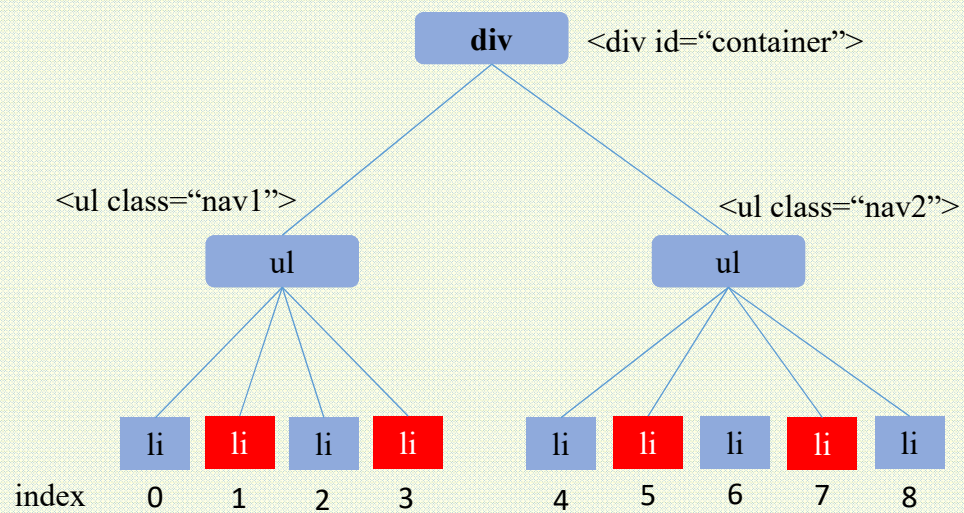
■ :odd 필터

`$('요소:odd')`

- 선택된 요소들 중 index가 홀수 번째 요소를 선택

예 `$("div#container li:odd")`

```
<div id="container">
  <ul class="nav1">
    <li> 0 </li>
    <li> 1 </li>
    <li> 2 </li>
    <li> 3 </li>
  </ul>
  <ul class="nav2">
    <li> A </li>
    <li> B </li>
    <li> C </li>
    <li> D </li>
    <li> E </li>
  </ul>
</div>
```



위치 관련 필터

6-filter-even-odd.htm (p.192)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){

      $("#menu li:even")
        .css({"background-color":"#ff0"});

      $("#menu li:odd")
        .css({"background-color":"#0ff"});

    });
  </script>
</head>
<body>
  <h1>탐색 선택자</h1>
  <ul id="menu">
    <li>내용1</li>
    <li>내용2</li>
    <li>내용3</li>
    <li>내용4</li>
  </ul>
</body>
</html>
```

탐색 선택자

- 내용1
- 내용2
- 내용3
- 내용4

자식 요소 관련 필터

■ 자식 요소 관련 필터

- 자식 노드와 자손 노드를 이용하는 선택자

필터	설명
: first-child : first-of-type	선택된 요소 중 첫 번째 자식 노드를 선택 - 부모 노드마다 각각 첫번째 요소가 선택됨
: last-child : last-of-type	선택된 요소 중 마지막 번째 자식 노드를 선택 - 부모 노드마다 각각 마지막 요소가 선택됨
: nth-child()	선택된 요소 중 nth-child()의 인자로 가지는 조건에 따라 자식 노드를 선택
: nth-last-child()	선택된 요소 중 마지막 위치에서 nth-child()의 인자로 가지는 조건에 따라 자식 노드를 선택
: only-child	선택된 요소 중 자식 노드가 하나뿐인 노드를 선택

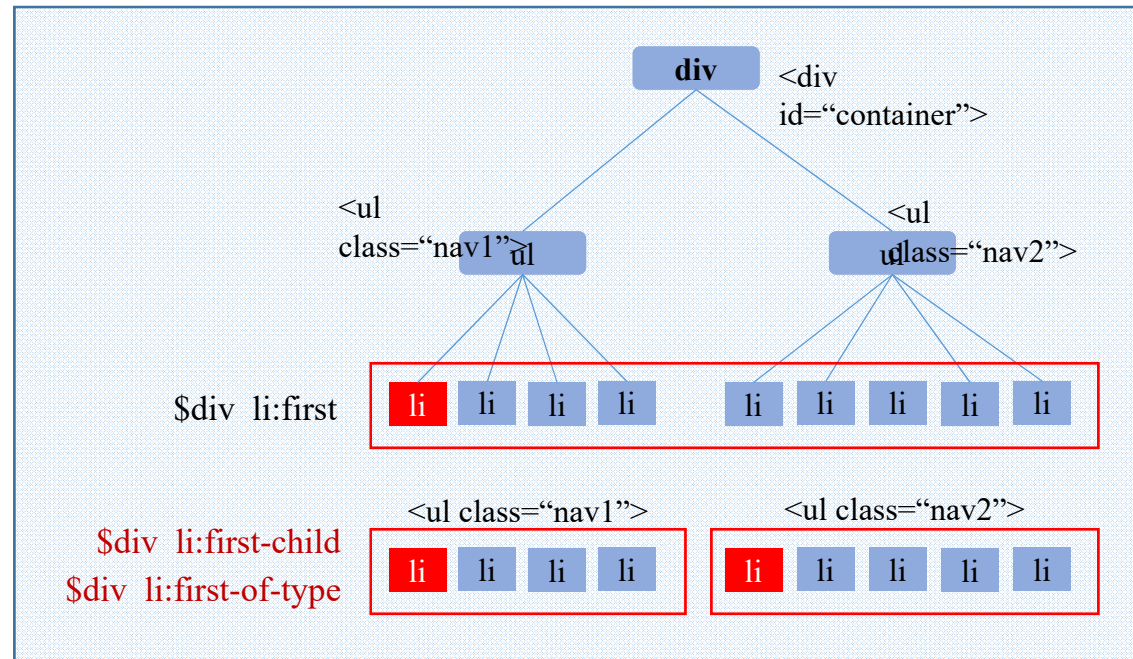
자식 요소 관련 필터

■ first-child 필터와 first-of-type 필터

`$('요소:first-child')`

`$('요소:first-of-type')`

- 선택자에 의해 선택된 요소 중 첫 번째 자식 요소를 선택
 - 부모 노드마다 각각 첫 번째 자식 요소가 선택됨



자식 요소 관련 필터

■ first-child 필터와 first-of-type 필터의 차이점

- first-child 필터

- 선택된 요소 중 필터로 지정한 요소가 반드시 첫 번째 요소여야 함
- 첫번째 요소가 아니면 효과가 적용되지 않음

- first-of-type 필터

- 선택된 요소 중 필터로 지정한 요소의 첫 번째 요소여야 함
- 반드시 첫 번째 요소가 아니어도 됨

자식 요소 관련 필터

6-filter-firstChild-firstofType.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("div p:first-of-type")
        .css({"background-color":"#0ff"});

      $("div p:first-child")
        .css({"color":"#f00"});
    });
  </script>
</head>
<body>
  <div>
    <div>text1-1</div>
    <p>text1-2</p>
    <p>text1-3</p>
  </div>
  <br>
  <div>
    <p>text2-1</p>
    <p>text2-2</p>
  </div>
</body>
</html>
```

text1-1

text1-2

text1-3

text2-1

text2-2

자식 요소 관련 필터

■ last-child 필터와 last-of-type 필터의 차이점

- last-child 필터
 - 선택된 요소 중 필터로 지정한 요소가 반드시 마지막 요소여야 함
 - 마지막 요소가 아니면 효과가 적용되지 않음
- first-of-type 필터
 - 선택된 요소 중 필터로 지정한 요소의 마지막 요소여야 함
 - 반드시 마지막 요소가 아니어도 됨
- 사용하는 방법은 first-child 필터와 first-of-type 필터와 같음

자식 요소 관련 필터

■ nth-child() 필터

```
$( '요소:nth-child( 인자 )' )
```

- 선택자에 의해 선택된 요소 중 인자로 지정된 규칙에 만족하는 요소를 선택
 - 인자로는 숫자, even, odd, 수식이 올 수 있음
- nth-child()에 사용되는 요소
 - nth-child(n)
 - n번째 있는 요소를 선택 (n은 index를 의미하는 것이 아님. 1부터 시작함)
 - nth-child(even)
 - 짝수 번째 있는 요소들을 선택
 - nth-child(odd)
 - 홀수 번째 있는 요소들을 선택
 - nth-child(수식)
 - nth-child(2n) : 짝수 번째 있는 요소들을 선택 (= :nth-child(even))
 - nth-child(2n+1) : 홀수 번째 있는 요소들을 선택 (= :nth-child(odd))
 - nth-child(3n+2) : 2, 5, 8, ... 번째 요소 선택

자식 요소 관련 필터

■ nth-last-child() 필터

```
$( '요소:nth-last-child( 인자 )' )
```

- 선택자에 의해 선택된 요소 중 마지막 요소부터 인자로 지정된 규칙에 만족하는 요소를 선택
- nth-last-child()에 사용되는 요소는 nth-child()에서와 동일

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $("#menu1 li:nth-child(1)")
        .css({"background-color":"#ff0"});

      $("#menu1 li:nth-child(2n)")
        .css({"border":"2px dashed #f00"});

      $("#menu2 li:nth-last-child(2)")
        .css({"background-color":"#0ff"});
    });

  </script>
</head>
```

자식 요소 관련 필터

```
<body>
  <h1>탐색 선택자</h1>
  <ul id="menu1">
    <li>내용1-1</li>
    <li>내용1-2</li>
    <li>내용1-3</li>
    <li>내용1-4</li>
  </ul>
  <ul id="menu2">
    <li>내용2-1</li>
    <li>내용2-2</li>
    <li>내용2-3</li>
  </ul>
</body>

</html>
```

탐색 선택자

- 내용1-1
- 내용1-2
- 내용1-3
- 내용1-4
- 내용2-1
- 내용2-2
- 내용2-3

자식 요소 관련 필터

■ only-child 필터

```
$('요소:only-child')
```

- 선택된 요소 중 자식 노드가 하나뿐인 요소를 선택

자식 요소 관련 필터

6-filter-only-child.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $("li:only-child")
        .css({"background-color":"#0ff"});
    });

  </script>
</head>

<body>
  <h1>탐색 선택자</h1>
  <ul id="menu1">
    <li>내용1-1</li>
    <li>내용1-2</li>
  </ul>
  <ul id="menu2">
    <li>내용2-1</li>
  </ul>
</body>
</html>
```

탐색 선택자

- 내용1-1
- 내용1-2
- 내용2-1

내용 관련 필터

■ 내용 관련 필터

- 문서에 존재한 요소들 중 특정 내용을 포함하고 있는 요소를 선택하는데 사용

내용 필터	설명
: contains()	선택된 요소 중 인자로 지정한 텍스트를 가지고 있는 요소를 모두 선택
: has()	선택된 요소 중 인자로 지정한 요소를 모두 선택
: empty	선택된 요소 중 내용이 존재하지 않은 빈 요소를 선택
: parent	선택된 요소 중 내용이 존재하는 요소를 선택 (empty 필터와 반대 개념)
: hidden	선택된 요소 중 보이지 않는 요소를 모두 선택
: visible	선택된 요소 중 보이는 요소를 선택

내용 관련 필터

■ contains() 필터와 has() 필터

- contains() 필터

```
$( '요소:contains( 텍스트 )' )
```

- 선택된 요소들 중 인자로 지정된 텍스트를 가지는 요소만을 선택

- has() 필터

```
$( '요소:has( 요소 )' )
```

- 선택된 요소들 중 인자로 지정된 요소(태그)를 가지는 요소만을 선택

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){

      $("#inner_1 p:contains(내용1)")
        .css({"background-color":"#ff0"});

      $("#inner_1 p:has(strong)")
        .css({"background-color":"#0ff"});

    });

  </script>
</head>
```


내용 관련 필터

```
<body>
```

```
<h3>콘텐츠 탐색 선택자</h3>
```

```
<section id="inner_1">
```

```
<h1>contains( ), has( )</h1>
```

```
<p><span>내용1</span></p>
```

```
<p><strong>내용2</strong></p>
```

```
<p><span>내용3</span></p>
```

```
</section>
```

```
</body>
```

```
</html>
```

콘텐츠 탐색 선택자

contains(), has()

내용1

내용2

내용3

내용 관련 필터

■ empty 필터와 parent 필터

• empty 필터

```
$('#요소:empty')
```

- 선택된 요소들 중 내용을 가지지 않는 빈 요소를 반환
- 시작 태그와 끝 태그 사이에 빈칸이 없어야 하며, 한 라인에 있어야 함
 - 공백 문자도 문자로 인식
 - 내용은 없지만 시작 태그와 끝 태그를 서로 다른 라인에 쓴 경우에도 비어 있지 않은 요소로 인식

• parent 필터

```
$('#요소:parent')
```

- 선택된 요소들 중 빈 요소가 아닌 요소들을 선택
- empty 필터와는 정반대의 의미를 가짐
 - 부모 요소와는 무관함

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){

      $("div#container li:parent")
      .css( {border : "1px solid red"} );

      $("div#container li:empty")
      .css( "background-color", "yellow" );

    });

  </script>
</head>
```

내용 관련 필터

```
<body>
  <div id="container">

    <div> fruit </div>

    <ul class="nav1">
      <li> apple </li>
      <li> banana </li>
      <li></li>
    </ul>

    <ul class="nav2">
      <li>
      </li>
      <li> tiger </li>
      <li> </li>
    </ul>

  </div>

</body>
</html>
```

fruit

- apple
- banana
-
-
- tiger
-

내용 관련 필터

■ hidden() 필터와 visible() 필터

• hidden() 필터

```
$('#요소:hidden')
```

- 선택된 요소 중 보이지 않는 요소를 모두 선택
- 요소가 보이지 않게 지정된 경우
 - CSS의 display 속성이 none으로 지정된 경우
 - <form>의 구성 요소 중 type이 hidden으로 지정된 경우
 - 요소의 width와 height가 각각 0으로 지정된 경우
 - 부모 요소가 보이지 않거나 숨겨져 있는 경우

• visible() 필터

```
$('#요소:visible')
```

- 선택된 요소 중 보이는 요소를 선택
 - hidden() 필터와 반대

내용 관련 필터

6-filter-hidden-visible.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $("p:visible").css("border","1px solid red");
      $("p:hidden").css("display","block");
    });

  </script>
</head>
<body>

  <h3>hidden, visible </h3>

  <p>내용1</p>
  <p style="display:none;">내용2</p>

</body>
</html>
```

hidden, visible

내용1

내용2

필터링(filtering) 메서드

필터링 메서드의 개요

■ filter() 메서드

```
$( '요소' ).filter( 인자 )
```

• 필터를 보완하기 위한 메서드

- 기본적으로 선택자와 필터를 사용하면 대부분의 요소를 선택할 수 있음
- jQuery는 요소를 선택하는 또다른 방법으로 filter()라는 메서드를 제공하고 있음
 - 기본적으로 제공되는 선택자와 필터로 탐색할 수 없는 경우 filter 메서드를 사용할 수 있음

• 필터 메서드를 사용하는 이유

- 메서드 체이닝(method chaining) 사용 가능
 - 메서드를 연결하여 동시에 적용하는 기법 사용 가능
- 선택자에 필터를 사용하는 것보다 filter 메서드를 적용하는 빠름
 - 선택자에서의 :first 필터와 탐색에서의 first() 메서드는 동일한 기능을 수행
 - :first보다 first()가 나중에 개발되어 속도도 빠른 편임

• 인자로 선택자, 필터, 콜백 함수를 사용할 수 있음

필터링 메서드의 개요

■ 인자로 선택자나 필터를 가지는 구조

```
$( "요소" ).filter( 선택자 | 필터 )
```

• 선택자를 가지는 경우

- 인자로 지정된 선택자에 의해 요소를 선택함
- [예] `$('li').filter('#st1').css('color', 'red')`
 - 요소들 중에서 st1이라는 아이디를 가지는 요소에 대해 글자색을 red로 지정

• 인자로 필터를 가지는 경우

- 인자로 지정된 필터에 해당하는 요소를 선택함
- [예] `$('li').filter(':even').css('color', 'red')`
 - 요소들 중에서 index가 짝수인 요소에 대해 글자색을 red로 지정

필터링 메서드의 개요

6-filtering-1.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').filter(':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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

필터링 메서드의 개요

■ 인자로 콜백함수를 가지는 구조

```
$("요소").filter( function() { ... } )
```

- 선택된 요소 중에서 인자로 가지는 콜백 함수를 만족하는 요소만을 선택

필터링 메서드의 개요

6-filtering-2.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').filter(function (index) {
        return index % 3 == 0;
      }).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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

필터링 메서드의 개요

■ 필터링 메서드

- filter() 메서드와 필터의 조합 일부를 메서드 형태로 표현한 것
- 예를 들어, 다음의 구문은 동일한 동작을 수행

- filter() 메서드와 .first 필터 사용

```
$("#p span").filter(":first").css("background-color","yellow");
```

- 필터링 메서드 사용

```
$("#p span").first().css("background-color","yellow");
```

- 필터링 메서드 = filter() 메서드 + 필터
 - 모든 필터들이 filter() 메서드와 결합되는 것은 아님
 - 자주 사용되는 필터와 filter() 메서드를 결합한 것임

필터링 메서드

■ 필터링 메서드의 종류

필터	설명
eq()	선택된 요소 중 인자로 가지는 index 번호에 해당하는 요소를 반환
first()	선택된 요소 중 첫 번째 요소를 반환
last()	선택된 요소 중 마지막 요소를 반환
has()	선택된 요소의 자식 요소 중 인자로 가지는 요소를 포함한 요소만을 반환
is()	선택된 요소들 중 인자로 가지는 선택자, 요소, jQuery 객체가 존재하면 true를 반환
not()	선택된 요소들 중 인자로 가지는 요소를 제외한 나머지 요소를 선택
slice()	선택된 요소들 중 인자로 가지는 조건에 만족하는 요소를 선택
add()	선택된 요소들에 추가로 요소를 선택
end()	객체의 선택을 한 단계 뒤로 돌림

필터링 메서드

■ first() 메서드와 last() 메서드

• first() 메서드

`$('요소').first()`

- 선택된 요소들 중 첫 번째의 요소를 선택

```
$( 'p' ).first().css( "background-color", "yellow" );  
= $( 'p' ).filter( ":first" ).css( "background-color", "yellow" );  
= $( 'p:first' ).css( "background-color", "yellow" );
```

• last() 메서드

`$('요소').last()`

- 선택된 요소들 중 첫 번째의 요소를 선택

```
$( 'p' ).last().css( "background-color", "yellow" );  
= $( 'p' ).filter( ":last" ).css( "background-color", "yellow" );  
= $( 'p:last' ).css( "background-color", "yellow" );
```

필터링 메서드

6-filtering-first-last.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').first().css("background","red");
      $('h3').last().css("background","green");
    });

  </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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

필터링 메서드

■ eq() 메서드

```
$('요소').eq( index )
```

- 선택된 요소들 중 인덱스 값이 index인 요소를 선택
 - index가 0인 경우
 - 선택된 요소 중 첫 번째 요소를 반환
 - index가 양수인 경우
 - 선택된 요소의 첫 번째 요소부터 n번째 요소를 반환 (첫 번째 요소의 n값은 0)
 - index가 음수인 경우
 - 선택된 요소의 마지막 요소부터 n번째 요소를 반환(마지막 요소의 n값은 1)

```
$('p').eq(2).css( "background-color", "yellow" );  
= $('p').filter(":eq(2)").css( "background-color", "yellow" );  
= $('p:eq(2)').css( "background-color", "yellow" );
```

필터링 메서드

6-filtering-eq.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').eq(1).css('background', 'orange');
      $('h3').eq(-2).css('background', 'red');
    });

  </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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

필터링 메서드

■ has() 메서드

```
$('요소').has( 요소 )
```

- 인자로 지정된 요소를 자식 요소로 가지고 있는 요소를 선택
 - 자식 요소를 선택하는 것이 아니라 자식 요소를 가지고 있는 해당 요소를 선택함에 유의
- [예] `$('li').has("ul").css('background-color', 'red');`
 - 선택된 요소들 중 요소를 자식 요소로 가지는 요소의 배경색을 red로 지정

```
$('li').has("ul").css( "background-color", "yellow" );  
= $('li').filter(":has('ul')").css( "background-color", "yellow" );  
= $('li:has('ul)').css( "background-color", "yellow" );
```

필터링 메서드

6-filtering-has.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('li').has('ul').css('backgroundColor', 'yellow');
    });

  </script>
</head>
<body>
  <ul>
    <li>충청북도
      <ul> <li>청주시</li>
        <li>충주시</li> </ul>
    </li>
    <li>충청남도
      <ol> <li>대전시</li>
        <li>천안시</li> </ol>
    </li>
  </ul>
</body>
</html>
```

- 충청북도
 - 청주시
 - 충주시
- 충청남도
 1. 대전시
 2. 천안시

필터링 메서드

■ not() 메서드와 is() 메서드

• not() 메서드

```
$( '요소' ).not( 요소 )
```

- 인자로 지정된 요소를 제외한 나머지 요소들을 선택

```
$( 'div' ).not( ".green, #blueone" ).css( "background-color", "yellow" );  
= $( 'div' ).filter( ':not( ".green, #blueone" )' ).css( "background-color", "yellow" );  
= $( 'div:not( ".green, #blueone" )' ).css( "background-color", "yellow" );
```

• is() 메서드

```
$( '요소' ).is( 요소 )
```

- 인자로 지정된 요소가 존재하는지 여부를 조사
 - 존재하는 경우 TRUE를 반환
 - 주로 이벤트와 함께 조건식으로 사용됨

필터링 메서드

6-filtering-not-is.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').not('.select').css('color','red');

      $('h3').each(function () {
        if ($(this).is('.select')) {
          $(this).css('background', 'orange');
        }
      });
    });

  </script>
</head>
<body>
  <h3 class="select">Header-0</h3>
  <h3>Header-1</h3>
  <h3 class="select">Header-2</h3>
  <h3>Header-3</h3>
</body>
</html>
```

Header-0

Header-1

Header-2

Header-3

필터링 메서드

■ slice() 메서드

```
$('요소').slice( start [, end] )
```

- index가 start인 요소부터 end인 요소를 선택
 - start
 - 추출을 위한 시작 인덱스 지정
 - 인덱스는 0부터 시작
 - 생략 불가
 - end
 - 추출을 위한 마지막 인덱스 지정
 - 인덱스는 1부터 시작
 - 생략 가능 (생략할 경우 start 부터 마지막 요소까지 모두 선택)

필터링 메서드

6-filtering-slice.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3')
        .slice(1,5)
        .css('backgroundColor','red')
        .slice(1,3)
        .css('backgroundColor','yellow');
    });

  </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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

필터링 메서드

■ add() 메서드

```
$('요소').add( 요소 )
```

- 선택된 범위를 확장하기 위한 메서드
 - 이미 선택되어진 요소에 인자로 지정된 요소를 추가로 선택
- 메서드 체이닝과 반대되는 개념
 - 메서드 체이닝은 범위를 좁혀가며 세부 선택을 수행
 - add() 메서드는 추가로 요소를 선택함으로써 기존에 선택 범위를 확장할 수 있음

필터링 메서드

6-filtering-add.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').slice(0,2)
        .add('h3:last')
        .css('backgroundColor','red');
    });

  </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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

필터링 메서드

■ end() 메서드

```
$('#요소'). end( 요소 )
```

- 요소가 선택되기 직전의 상태로 되돌림
 - 직전의 선택을 취소하는 것이 아님
- [예] 아래의 구문을 메서드 체이닝으로 구현할 경우

```
$('#h1').css('background', 'yellow');  
$('#h1:even').css('color', 'red');  
$('#h1:odd').css('color', 'blue');
```

- 잘못된 표현

```
$('#h1').css('background','yellow').filter(':even').css('color','red').filter(':odd').css('color','blue');
```

- 바른 표현

```
$('#h1').css('background','yellow').filter(':even').css('color','red').end(). filter(':odd').css('color','blue');
```

필터링 메서드

6-filtering-end.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">

    $(function(){
      $('h3').css('background', 'yellow')
      .filter(':even').css('color', 'blue')
      .end()
      .filter(':odd').css('color', 'red');
    });

  </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>
```

Header-0

Header-1

Header-2

Header-3

Header-4

Header-5

수고하셨습니다