

jQuery

jQuery기본 사용법, Selector

download

- <http://jquery.com/>
- download vs. link
 - compressed
 - jquery-버전번호.min.js
 - uncompressed
 - jquery-버전번호.js
- link (compressed or uncompressed)
 - <http://code.jquery.com/jquery-버전번호.js>
 - <http://code.jquery.com/jquery-latest.js>

download방식

- compressed

```
<script  
  src="jquery-버전번호.min.js">  
</script>
```

- uncompressed

```
<script  
  src="jquery-버전번호.js">  
</script>
```

link 방식

- 최신 라이브러리 이용

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

- compressed

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

- uncompressed

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

jQuery 기본 사용법

동작 실행

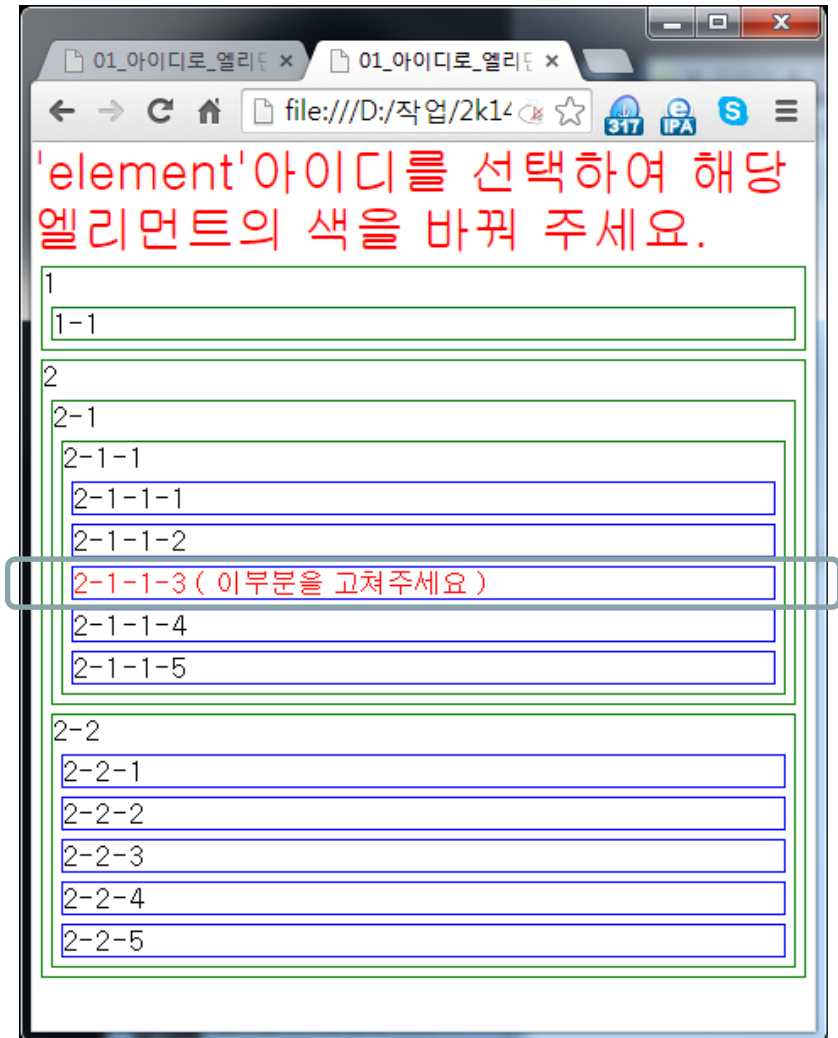
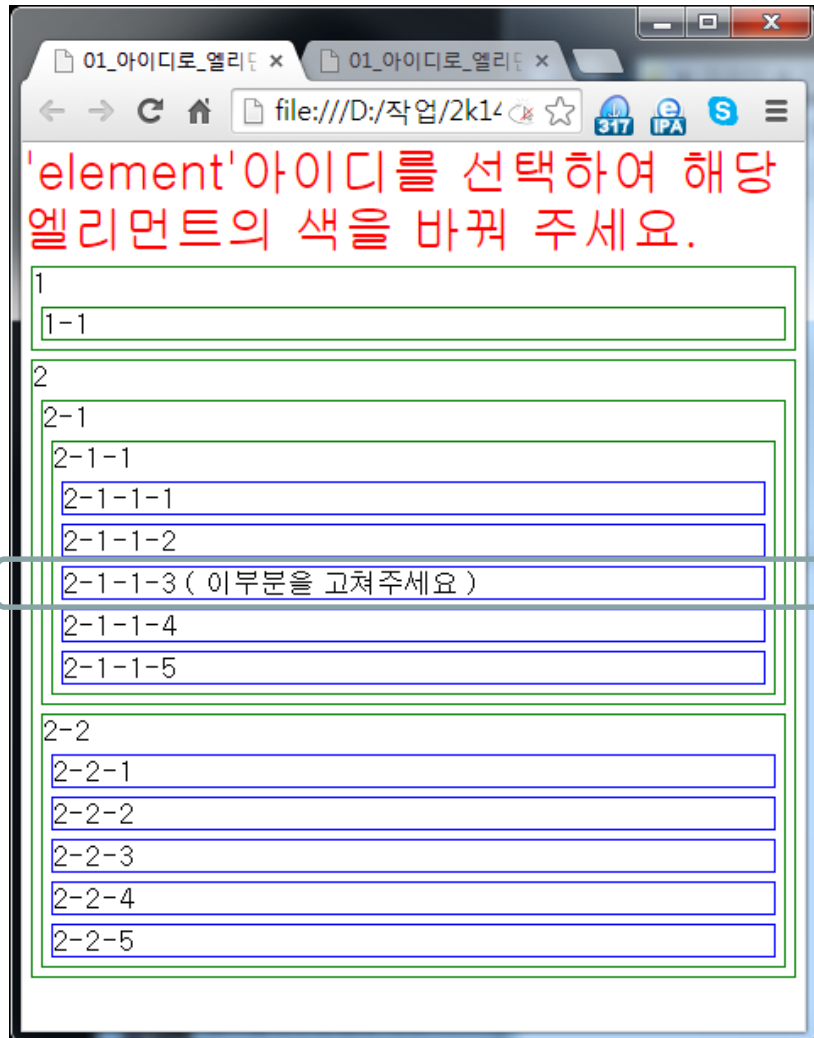
• `$ (' #menu ') . addClass (' hello ') ;`
객체 선택

- jQuery(selector) or \$(selector)
- Selector
 - 기본 CSS selector
 - 위치기반필터 selector
 - 상태필터 selector

기본 CSS selector

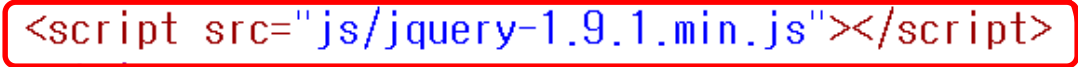
| 선택 예 | 설명 |
|---------|---|
| * | 모든 엘리먼트가 선택됩니다. |
| E | 태그명이 E인 엘리먼트가 선택됩니다. |
| E F | 태그명이 E인 엘리먼트의 자손 중 태그명이 F인 엘리먼트가 선택됩니다. |
| E>F | 태그명이 E인 엘리먼트의 자식 중 태그명이 F인 엘리먼트가 선택됩니다. |
| E+F | 태그명이 E인 엘리먼트의 바로 아래 태그명이 F인 형제 엘리먼트가 선택됩니다. |
| E~F | 태그명이 E인 엘리먼트의 형제들 중 태그명이 F인 엘리먼트가 선택됩니다. |
| E.C | 태그명이 E이고 Class C를 가진 엘리먼트가 선택됩니다. |
| E#I | 태그명이 E이고 id의 값이 I인 엘리먼트가 선택됩니다. |
| E[A] | 태그명이 E이고 어트리뷰트 A를 가진 엘리먼트가 선택됩니다. |
| E[A=V] | 태그명이 E이고 어트리뷰트 A의 값 V를 가진 엘리먼트가 선택됩니다. |
| E[A!=V] | 태그명이 E이고 어트리뷰트 A의 값이 V가 아닌 엘리먼트가 선택됩니다. |

기본 CSS selector: id



기본 CSS selector: id

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title>01_아이디로_엘리먼트찾기</title>
6   <script src="js/jquery-1.9.1.min.js"></script>
7   <style>
8     html * { margin:0; padding:0; }
9     body span { color:red; font-size: 30px;}
10    div { border:1px solid green; margin:5px; }
11    li { border:1px solid blue; list-style: none; margin:5px;}
12  </style>
13 </head>
14 <body>
15   <span>'element'아이디를 선택하여 해당 엘리먼트의 색을 바꿔 주세요.</span>
16   <div>
17     1
18     <div>
19       1-1
20     </div>
21   </div>
```



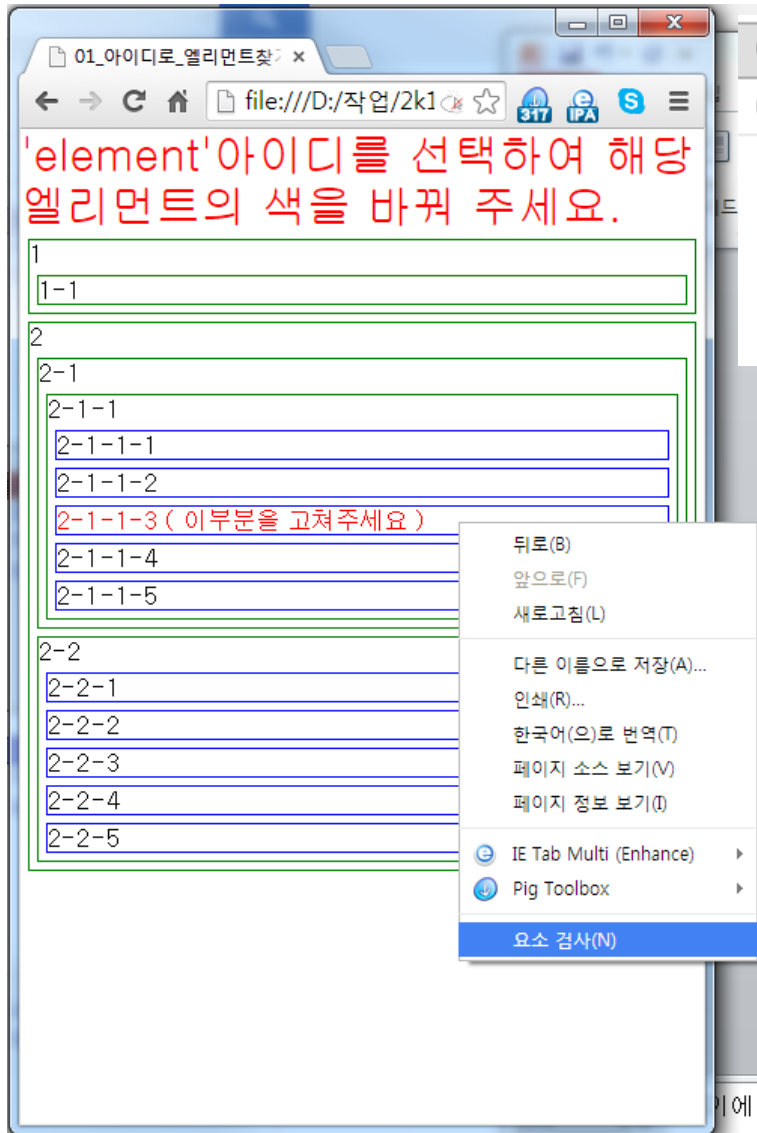
기본 CSS selector: id

```
21 </div>
22 <div>
23   2
24   <div>
25     2-1
26     <div>
27       2-1-1
28       <ul>
29         <li>2-1-1-1</li>
30         <li>2-1-1-2</li>
31         <li id="element">2-1-1-3 ( 이부분을 고쳐주세요 )</li>
32         <li>2-1-1-4</li>
33         <li>2-1-1-5</li>
34       </ul>
35     </div>
36   </div>
37   <div>
38     2-2
39     <ul>
40       <li>2-2-1</li>
41       <li>2-2-2</li>
42       <li>2-2-3</li>
43       <li>2-2-4</li>
44       <li>2-2-5</li>
45     </ul>
46   </div>
47 </div>
48 </body>
49 </html>
```

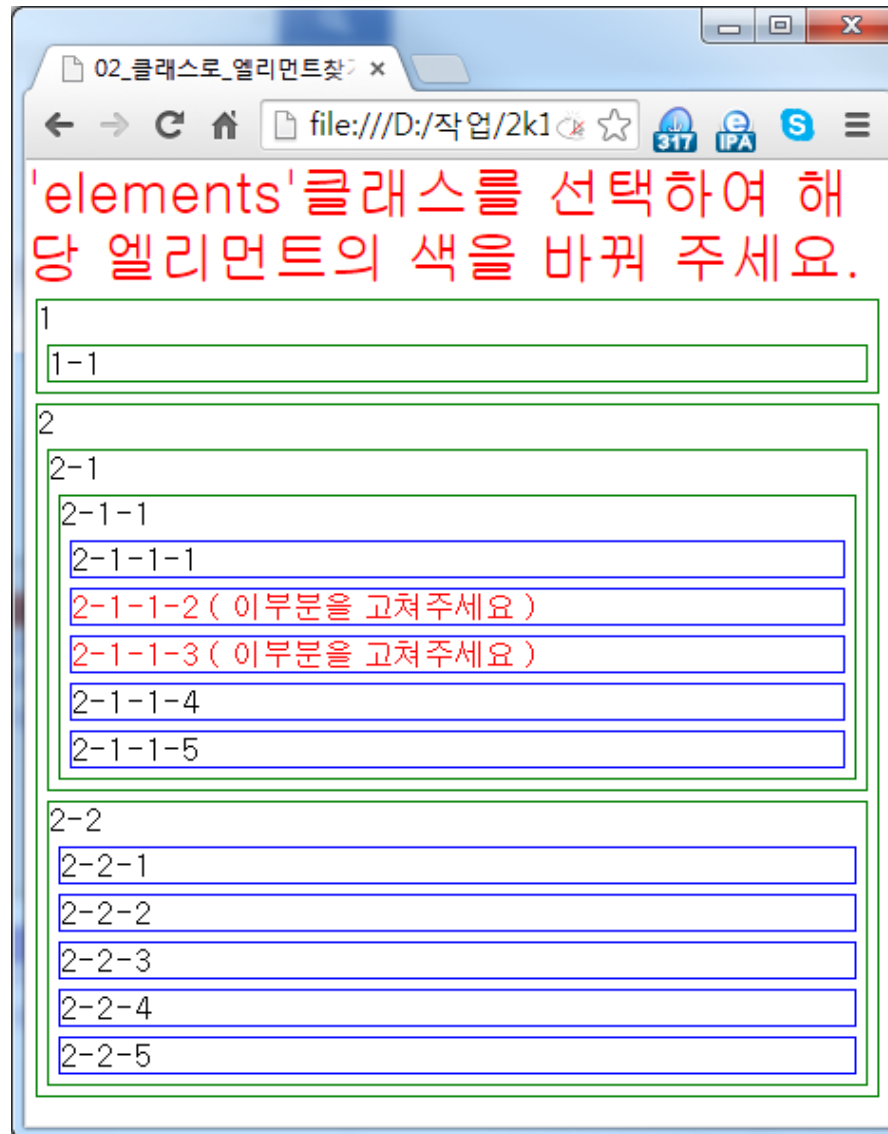
기본 CSS selector: id

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8"/>
5   <title>01_아이디로_엘리먼트찾기</title>
6   <script src="js/jquery-1.9.1.min.js"></script>
7   <style>
8     html * { margin:0; padding:0; }
9     body span { color:red; font-size: 30px;}
10    div { border:1px solid green; margin:5px; }
11    li { border:1px solid blue; list-style: none; margin:5px;}
12  </style>
13  <script>
14    $(document).ready(function(){
15      var el = $('#element');
16      $('#element').css('color', 'red');
17      console.log(el);
18    });
19  </script>
20 </head>
```

inspector 사용법



기본 CSS selector: class

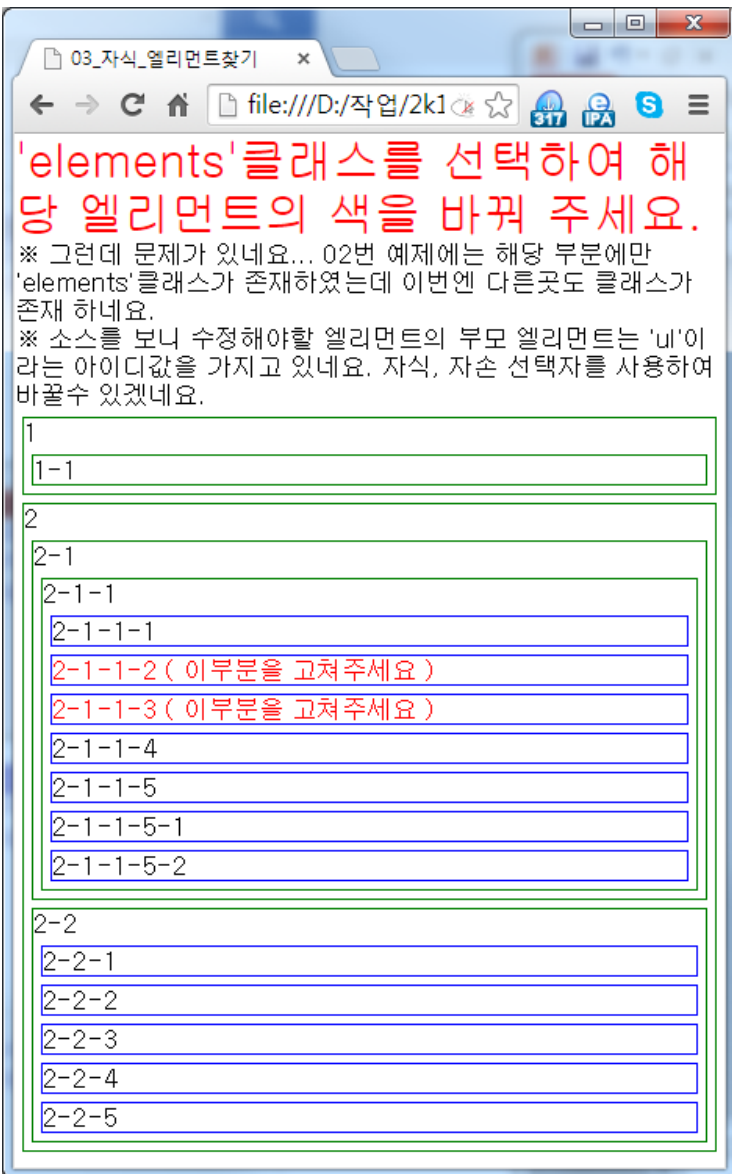


기본 CSS selector: class

```
33 <ul>
34   <li>2-1-1-1</li>
35   <li class="elements">2-1-1-2 ( 이부분을 고쳐주세요 )</li>
36   <li class="elements">2-1-1-3 ( 이부분을 고쳐주세요 )</li>
37   <li>2-1-1-4</li>
38   <li>2-1-1-5</li>
39 </ul>
```

```
13 <script>
14   $(document).ready(function(){
15     $('.elements').css('color', 'red');
16   });
17 </script>
```

기본 CSS selector: child



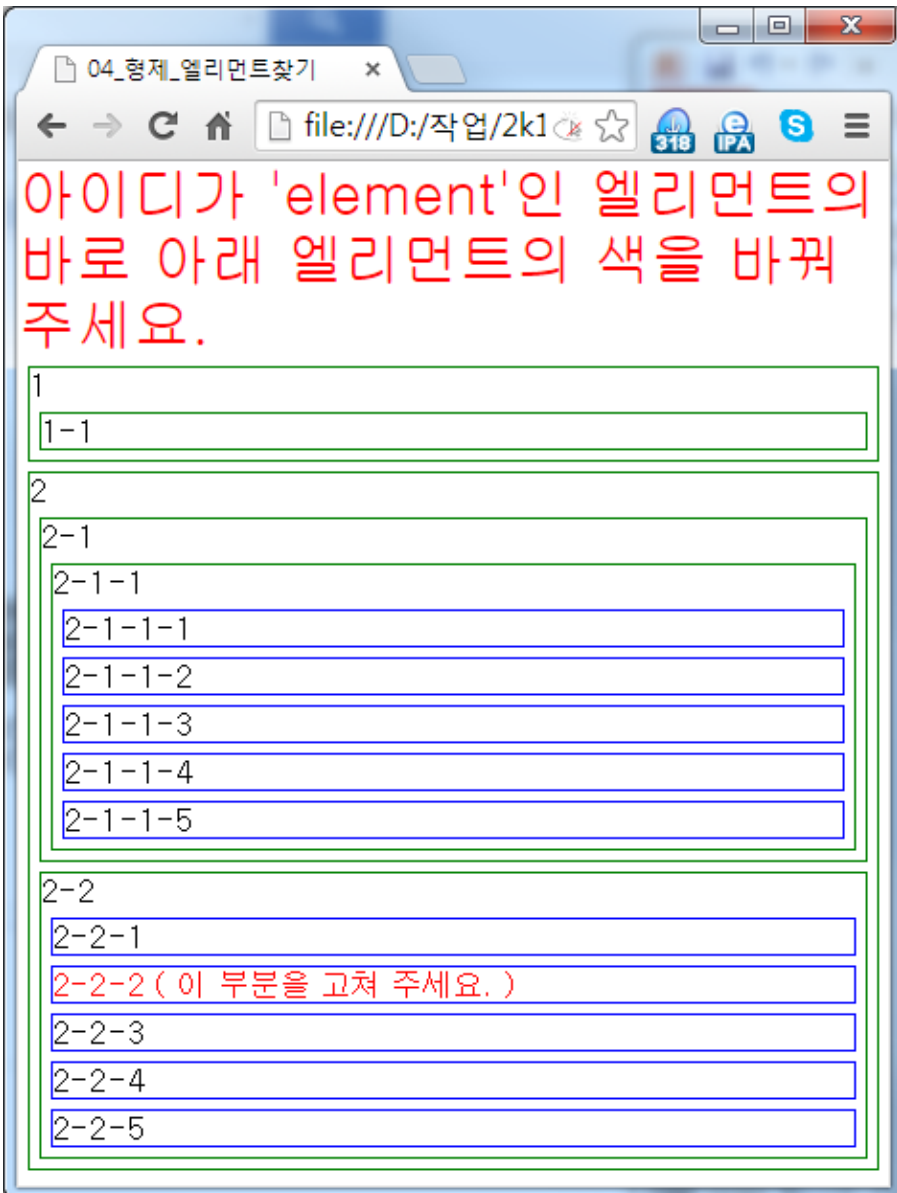
```
<ul id="ul">
  <li>2-1-1-1</li>
  <li class="elements">2-1-1-2 ( 이부분을 고쳐주세요 )</li>
  <li class="elements">2-1-1-3 ( 이부분을 고쳐주세요 )</li>
  <li>2-1-1-4</li>
  <li>2-1-1-5</li>
</ul>
  <li class="elements">2-1-1-5-1</li>
  <li class="elements">2-1-1-5-2</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    $('#ul > .elements').css('color', 'red');
  });
</script>
```

기본 CSS selector: child

- 부모 엘리먼트의 자식 엘리먼트 선택: **>** 사용
 - #ul > .elements
 - 아이디가 ul인 엘리먼트**의 자식들 중** class가 elements인 엘리먼트들 선택
 - #ul .elements
 - 아이디가 ul인 엘리먼트**의 자손들 중** class가 elements인 엘리먼트들 선택
 - #ul.elements
 - 아이디가 ul인 엘리먼트**이고** class가 elements인 엘리먼트들 선택

기본 CSS selector: sibling



```
<ul>
```

```
<li id="element">2-2-1</li>
```

```
<li>2-2-2 ( 이 부분을 고쳐 주세요. )</li>
```

```
<li>2-2-3</li>
```

```
<li>2-2-4</li>
```

```
<li>2-2-5</li>
```

```
</ul>
```

```
<script>
```

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

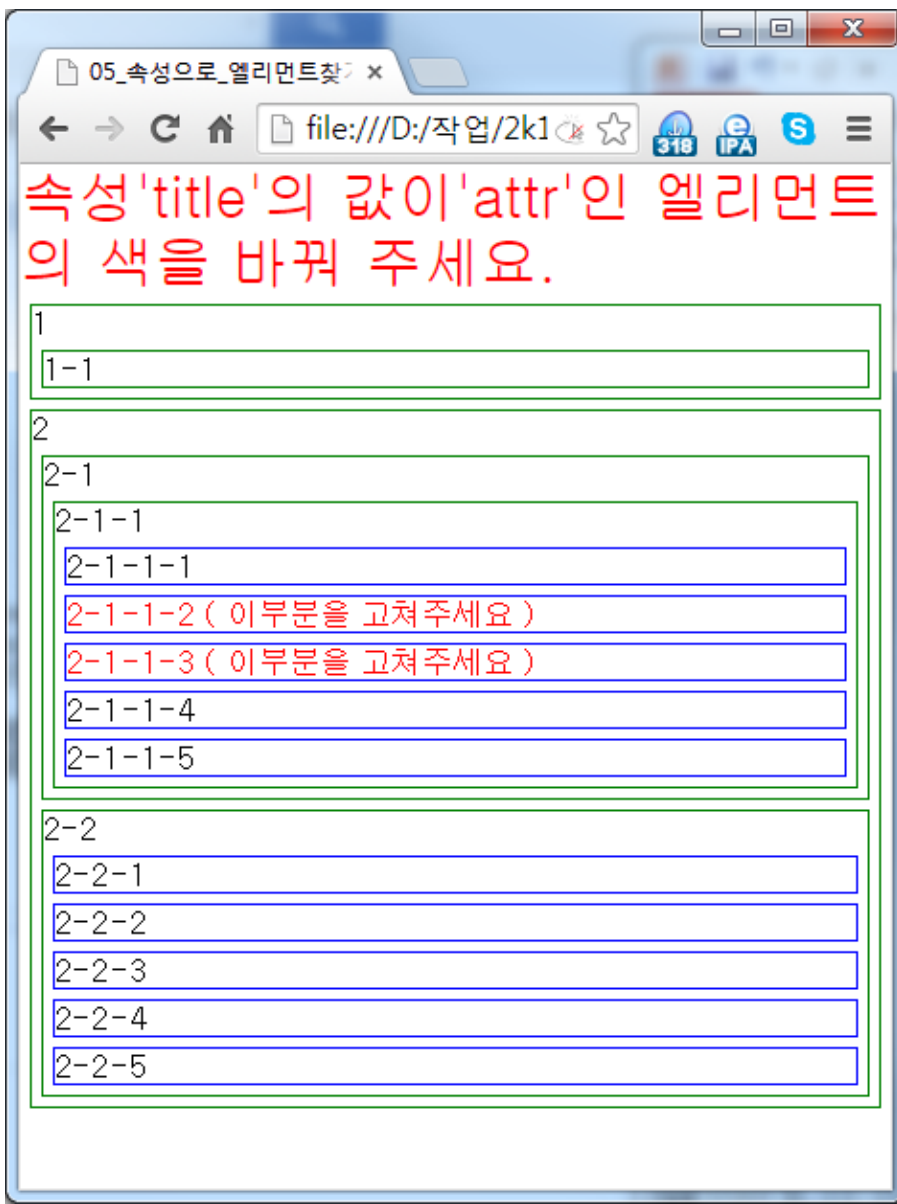
```
$('#element + li').css('color', 'red');
```

```
});
```

```
</script>
```

- #element + li
 - 아이디가 element인 엘리먼트의 바로 아래 태그명이 li인 엘리먼트
- #element ~ li
 - 아이디가 element인 엘리먼트의 형제들 중 태그명이 li인 엘리먼트

기본 CSS selector: attribute



```
<ul>
  <li>2-1-1-1</li>
  <li title="attr">2-1-1-2 ( 이부분을 고쳐주세요 )</li>
  <li title="attr">2-1-1-3 ( 이부분을 고쳐주세요 )</li>
  <li>2-1-1-4</li>
  <li>2-1-1-5</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    $(' [title=attr] ').css('color', 'red');
  });
</script>
```

위치기반 필터 selector

니다.

설명

셀렉터

첫 번째 일치하는 엘리먼트를 선택합니다.

:first

마지막으로 일치하는 엘리먼트를 선택합니다.

:last

각 리스트의 첫 번째 리스트 아이템을 반환합니다.

:first-child

각 리스트의 마지막 리스트 아이템을 반환합니다.

:last-child

부모의 형제가없는 자식인 모든 엘리먼트를 선택합니다.

:only-child

부모의 n번째 자식인 모든 엘리먼트를 선택합니다.

:nth-child(n)

:even

짝수번째 엘리먼트를 선택합니다.

:odd

홀수번째 엘리먼트를 선택합니다.

:eq(n)

일치하는 n번 index의 엘리먼트를 선택합니다.

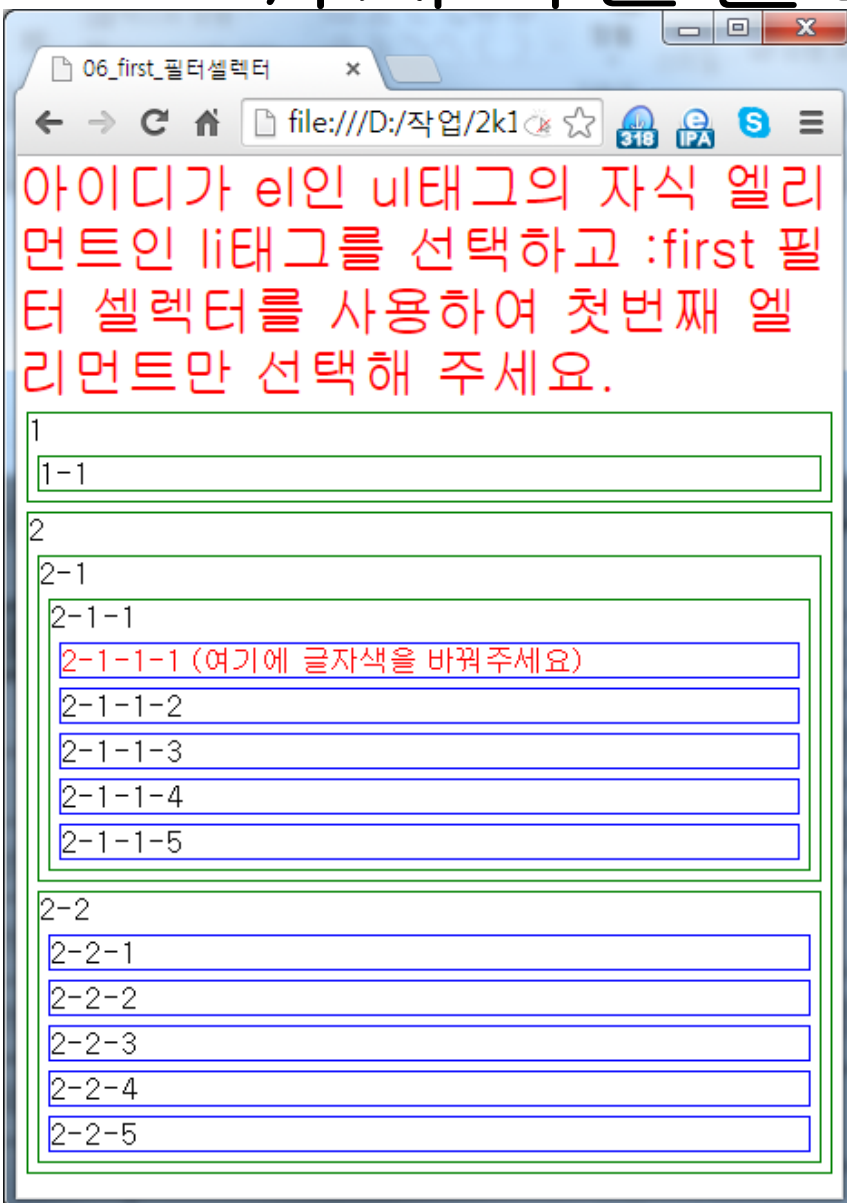
:gt(n)

n번 index를 포함하지 않고 이후의 엘리먼트와 일치하는 엘리먼트를 선택합니다.

:lt(n)

n번 index를 포함하지 않고 이전의 엘리먼트와 일치하는 엘리먼트를 선택합니다.

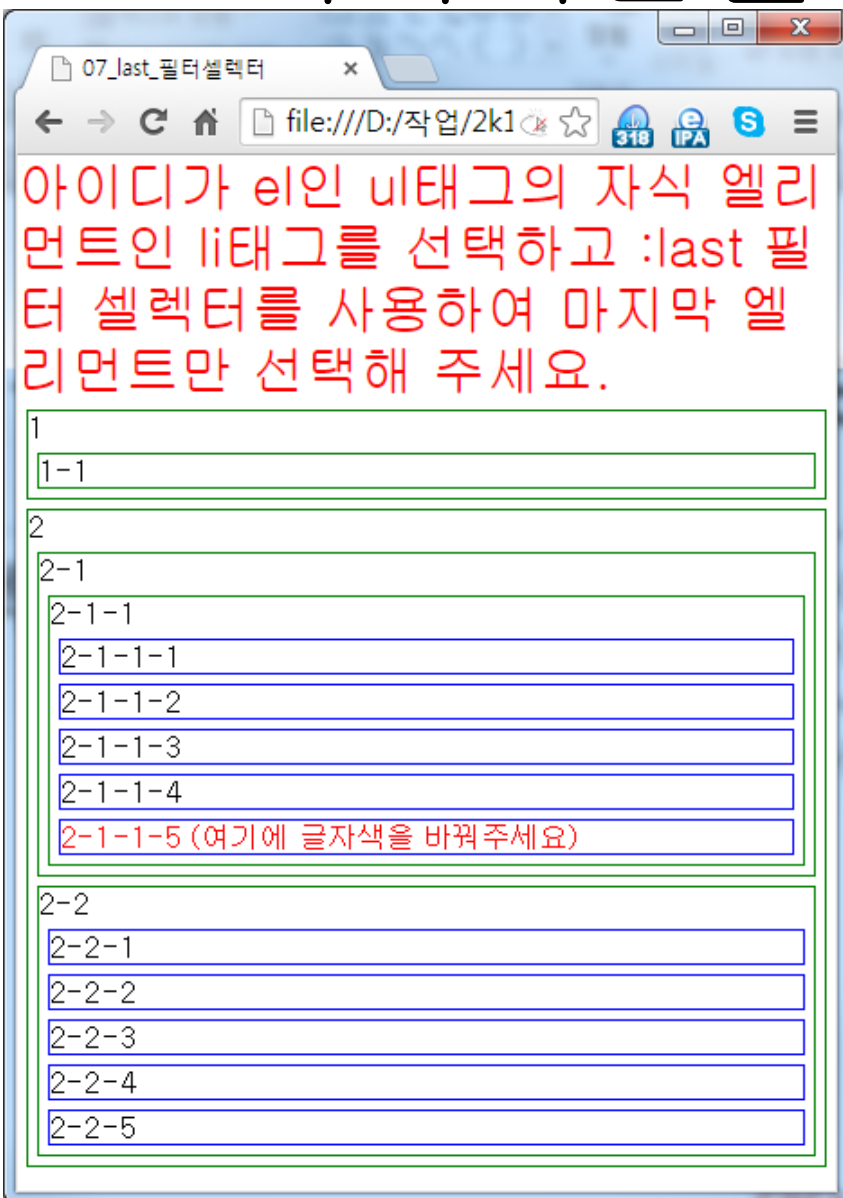
위치기반필터 selector: first



```
<ul id="el">
  <li>2-1-1-1 (여기에 글자색을 바꿔주세요)</li>
  <li>2-1-1-2</li>
  <li>2-1-1-3</li>
  <li>2-1-1-4</li>
  <li>2-1-1-5</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:first');
    el.css('color', 'red');
  });
</script>
```

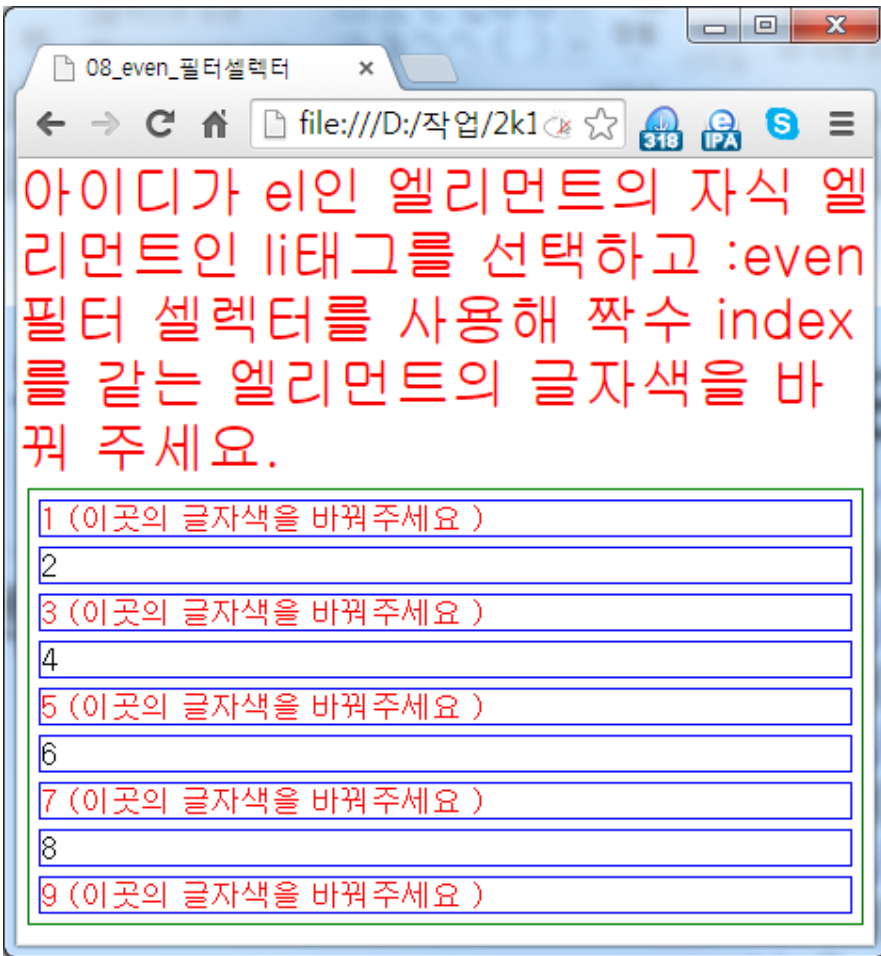
위치기반필터 selector: last



```
<ul id="el">
  <li>2-1-1-1</li>
  <li>2-1-1-2</li>
  <li>2-1-1-3</li>
  <li>2-1-1-4</li>
  <li>2-1-1-5 (여기에 글자색을 바꿔주세요)</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:last');
    el.css('color', 'red');
  });
</script>
```

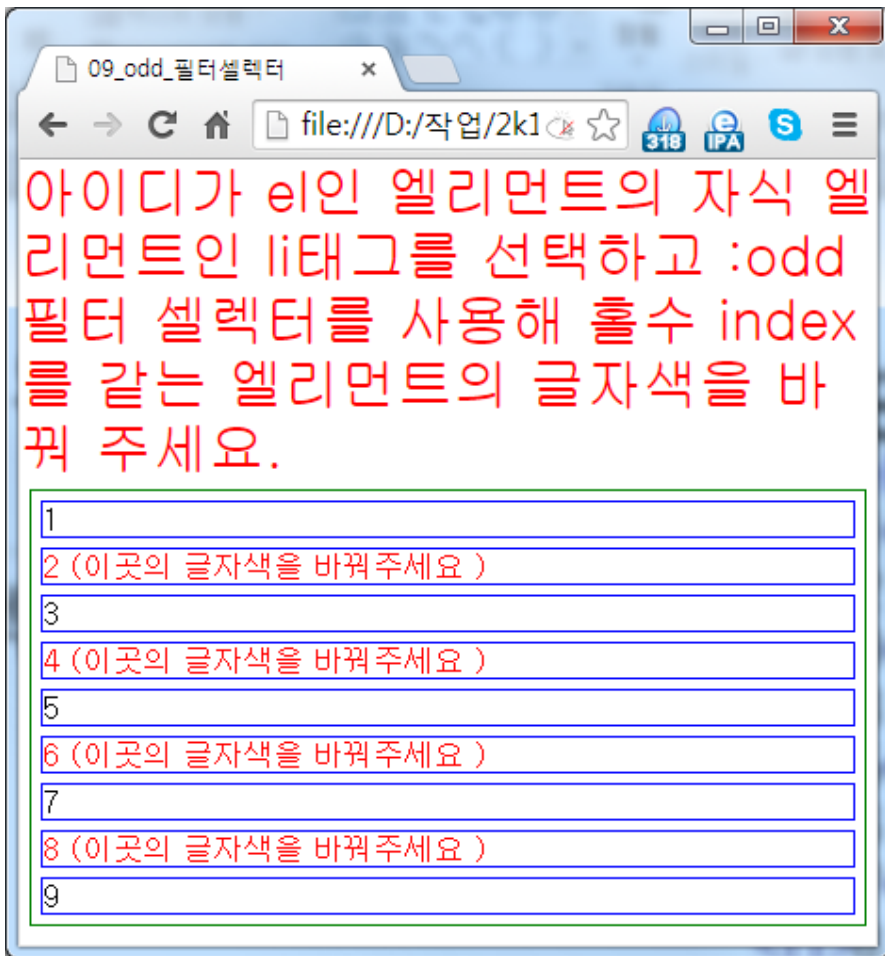
위치기반필터 selector: even



```
<ul id="el">
  <li>1 (이곳의 글자색을 바꿔주세요 )</li>
  <li>2</li>
  <li>3 (이곳의 글자색을 바꿔주세요 )</li>
  <li>4</li>
  <li>5 (이곳의 글자색을 바꿔주세요 )</li>
  <li>6</li>
  <li>7 (이곳의 글자색을 바꿔주세요 )</li>
  <li>8</li>
  <li>9 (이곳의 글자색을 바꿔주세요 )</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:even');
    el.css('color', 'red');
  });
</script>
```

위치기반필터 selector: odd

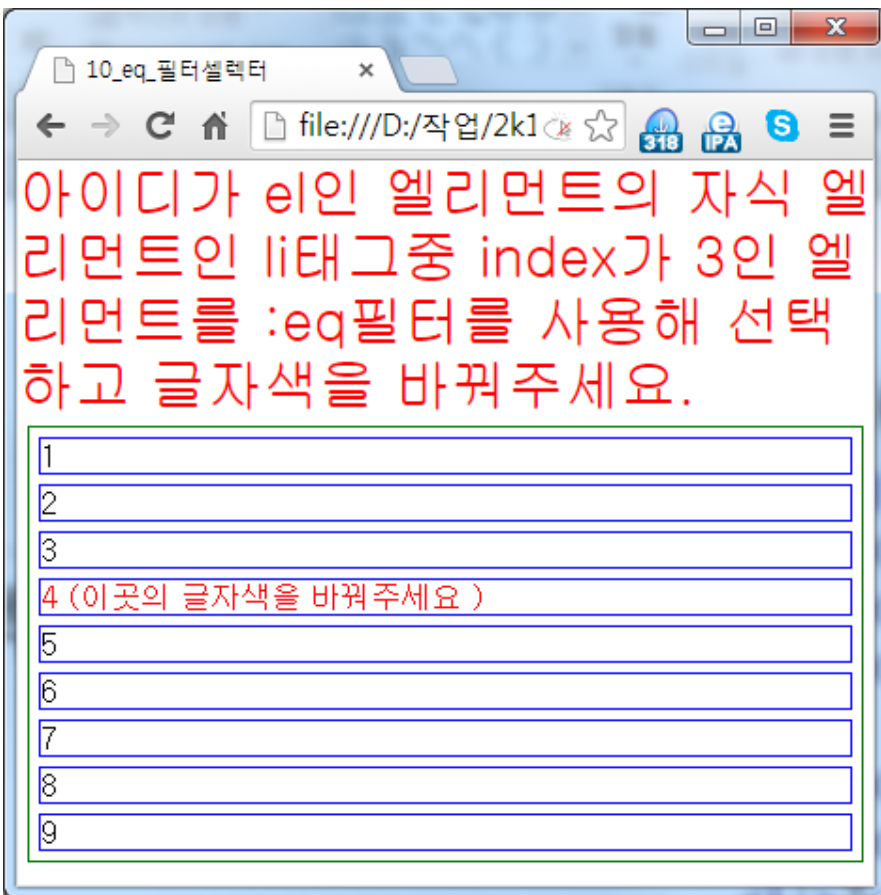


```
<ul id="el">
  <li>1</li>
  <li>2 (이곳의 글자색을 바꿔주세요 )</li>
  <li>3</li>
  <li>4 (이곳의 글자색을 바꿔주세요 )</li>
  <li>5</li>
  <li>6 (이곳의 글자색을 바꿔주세요 )</li>
  <li>7</li>
  <li>8 (이곳의 글자색을 바꿔주세요 )</li>
  <li>9</li>
```

```
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:odd');
    el.css('color', 'red');
  });
</script>
```

위치기반필터 selector: eq

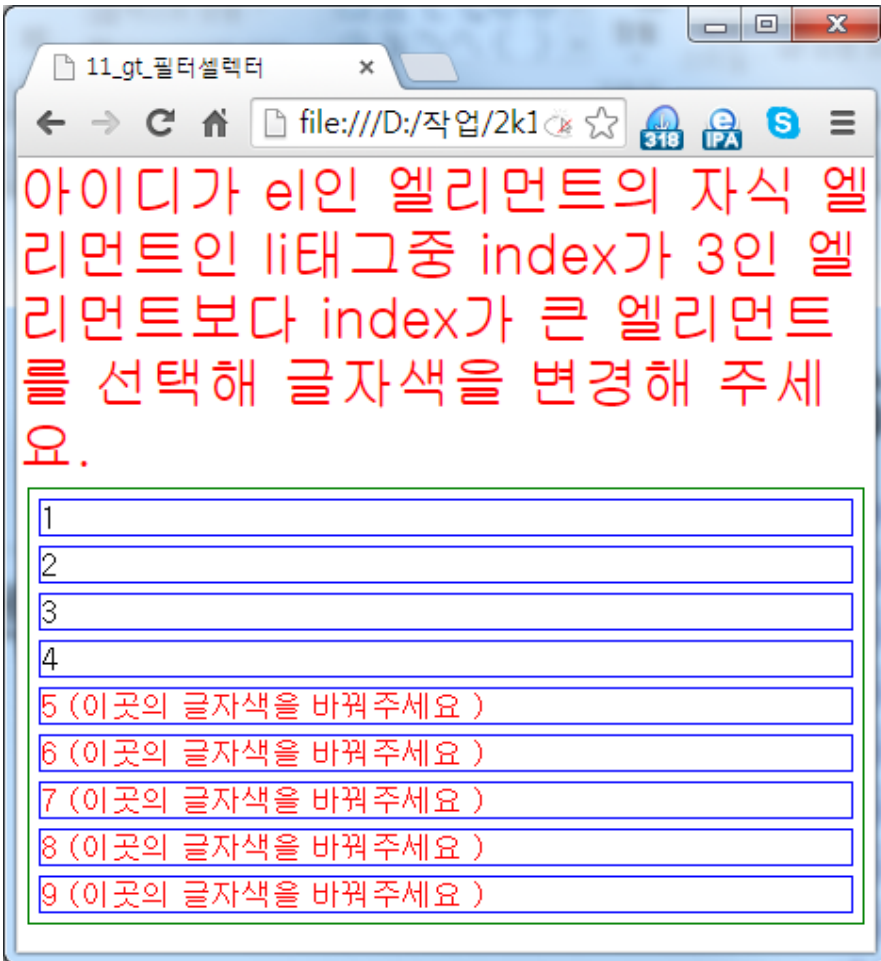


```
<ul id="el">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4 (이곳의 글자색을 바꿔주세요 )</li>
  <li>5</li>
  <li>6</li>
  <li>7</li>
  <li>8</li>
  <li>9</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:eq(3)');
    el.css('color', 'red');
  });
</script>
```

index번호

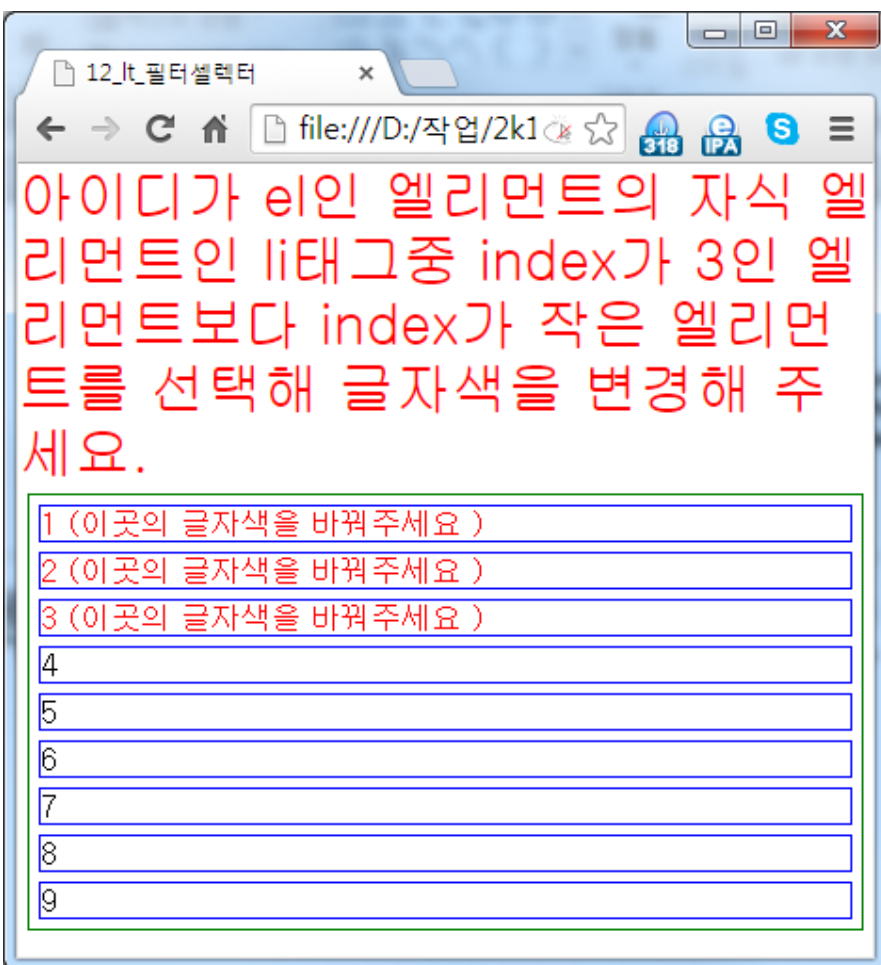
위치기반필터 selector: gt



```
<ul id="el">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5 (이곳의 글자색을 바꿔주세요 )</li>
  <li>6 (이곳의 글자색을 바꿔주세요 )</li>
  <li>7 (이곳의 글자색을 바꿔주세요 )</li>
  <li>8 (이곳의 글자색을 바꿔주세요 )</li>
  <li>9 (이곳의 글자색을 바꿔주세요 )</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:gt(3)');
    el.css('color', 'red');
  });
</script>
```


위치기반필터 selector: lt



```
<ul id="el">
  <li>1 (이곳의 글자색을 바꿔주세요 )</li>
  <li>2 (이곳의 글자색을 바꿔주세요 )</li>
  <li>3 (이곳의 글자색을 바꿔주세요 )</li>
  <li>4</li>
  <li>5</li>
  <li>6</li>
  <li>7</li>
  <li>8</li>
  <li>9</li>
```

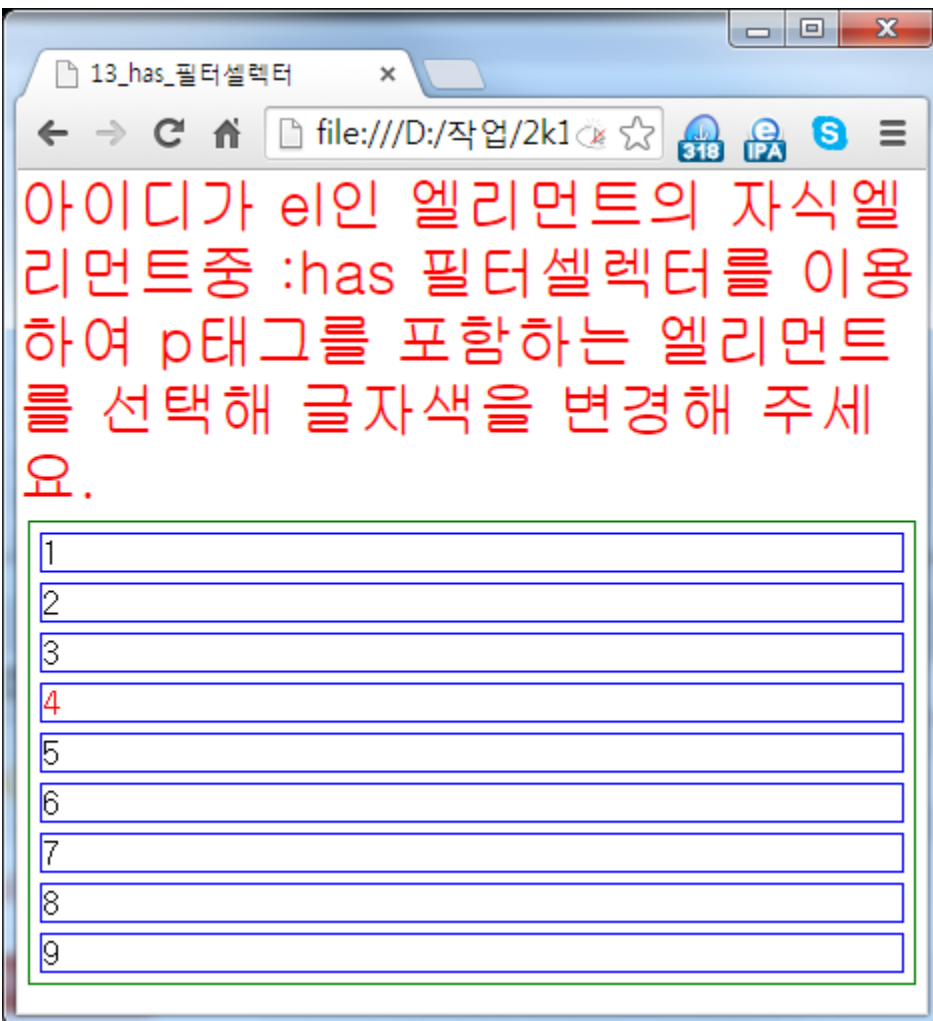
```
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:lt(3)');
    el.css('color', 'red');
  });
</script>
```

상태기반 selector

| 셀렉터 | 설명 |
|-----------------|---|
| .animated | 선택 상태가 실행되는 시점에서 진행 중인 애니메이션의 모든 엘리먼트를 선택합니다. |
| :button | 버튼 엘리먼트를 선택합니다. |
| :checkbox | 체크박스를 선택합니다. |
| :checked | 체크박스나 라디오 버튼 중 체크 또는 선택된 것들만 선택합니다. |
| :contains(text) | 지정된 텍스트를 포함하는 모든 엘리먼트를 선택합니다. |
| :disabled | 비활성화 상태인 엘리먼트를 선택합니다. |
| :enabled | 활성화 상태인 엘리먼트를 선택합니다. |
| :file | 파일타입 엘리먼트를 선택합니다. |
| :has(selector) | 셀렉터와 일치한 최소 한개 이상의 엘리먼트를 포함하는 엘리먼트만 선택합니다. |
| :header | H1, H2, H3 등과 같은 헤더 모든 요소를 선택합니다. |
| :hidden | 숨겨진 모든 요소를 선택합니다. |
| :image | 이미지 타입의 엘리먼트를 선택합니다. |
| :input | 모든 입력, 텍스트 영역을 선택합니다. |
| :not(selector) | 지정한 selector와 일치하지 않은 엘리먼트를 선택합니다. |
| :parent | 자식 엘리먼트를 가지는 엘리먼트를 선택합니다. |
| :password | 패스워드 엘리먼트를 선택합니다. |
| :radio | 라디오 버튼을 선택합니다. |
| :reset | 리셋 버튼을 선택합니다. |
| :selected | 선택된 모든 엘리먼트를 선택합니다. |
| :submit | 전송 버튼을 선택합니다. |
| :text | 텍스트 타입 엘리먼트를 선택합니다. |
| :visible | 보이는 엘리먼트를 선택합니다. |

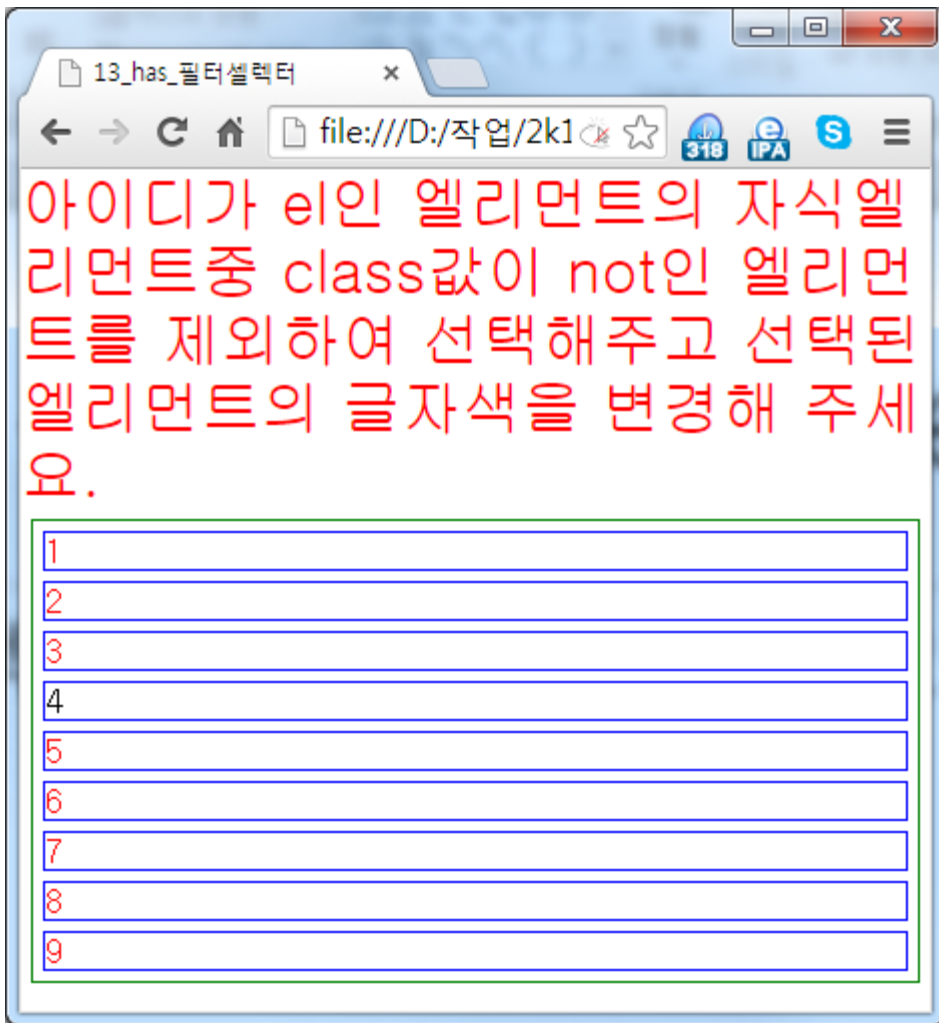
상태기반 selector: has



```
<ul id="el">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li><p>4</p></li>
  <li>5</li>
  <li>6</li>
  <li>7</li>
  <li>8</li>
  <li>9</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:has(p)');
    el.css('color', 'red');
  });
</script>
```

상태기반 selector: not



```
<ul id="el">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li class="not">4</li>
  <li>5</li>
  <li>6</li>
  <li>7</li>
  <li>8</li>
  <li>9</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var el = $('#el > li:not(.not)');
    el.css('color', 'red');
  });
</script>
```