jQuery DOM 탐색

재단법인 대덕인재개발원
Daedeok Human Resource Development Foundation

참조

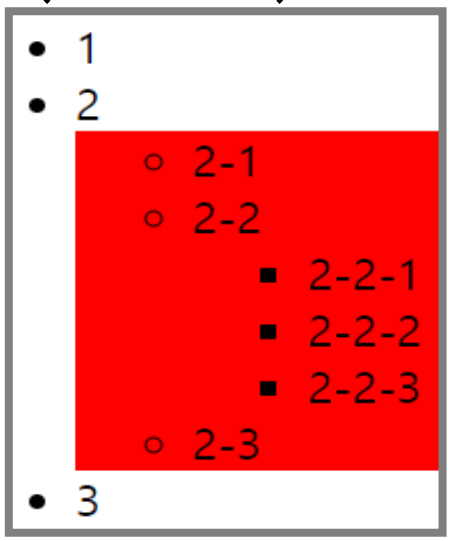
- [TUTORIALS] > [JavaScript] > [Learn jQuery] > [jQuery Traversing] > [jQuery Traversing] ~ [jQuery Filtering]
  - https://www.w3schools.com/jquery/jquery_traversing.asp
  - https://www.w3schools.com/jquery/jquery_traversing_ancestors.asp
  - https://www.w3schools.com/jquery/jquery_traversing_descendants.asp
  - https://www.w3schools.com/jquery/jquery_traversing_siblings.asp
  - https://www.w3schools.com/jquery/jquery_traversing_filtering.asp
- [TUTORIALS] > [JavaScript] > [Learn jQuery] > [jQuery Traversing] > [jQuery References] > [jQuery Traversing]
  - https://www.w3schools.com/jquery/jquery_ref_selectors.asp

DOM 탐색 메소드(1/4)

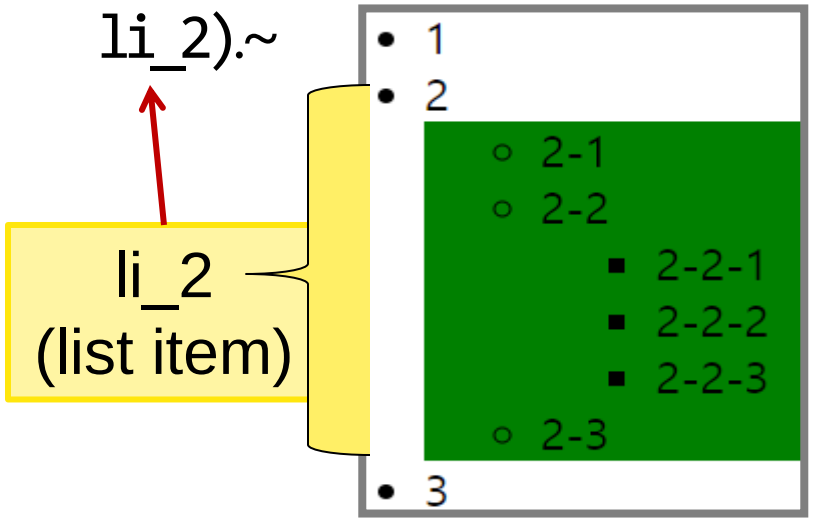
1. 상위 요소 찾기	
$(*selector*).parent(*filter*)	선택된 요소의 부모 요소를 반환 (*filter*를 옵션으로 줄 수 있음)
$(*selector*).parents(*filter*) $(*selector*).parents(*filter*, *filter*, …)	선택된 요소의 모든 상위 요소 반환(*filter* 옵션 가능) 최상위 요소 html 까지 반환함
$(*selector*).parentsUntil(*stop*, *filter*)	*selector*와 *stop* 사이의 요소를 반환(*selector*, *stop*은 불포함) *stop*이 명시되지 않으면 최상위 요소(html)까지 반환
$(*selector*).closest(*filter*, *context*)	*selector*에 가장 가까이에 있는 상위 요소를 반환(자신도 포함) *context*로 범위 지정가능 (*context* 불포함)

closest(filter) 메소드 예제(1/3)

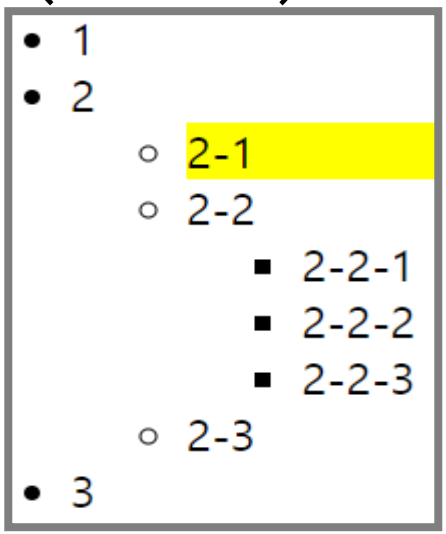
◆ $('li.2-1').closest('ul').~



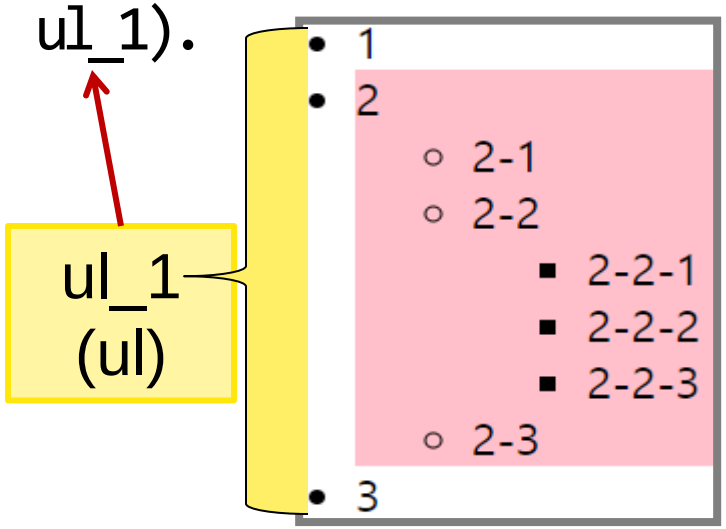
◆ $('li.2-1').closest('ul', li_2).~



◆ $('li.2-1').closest('li').~



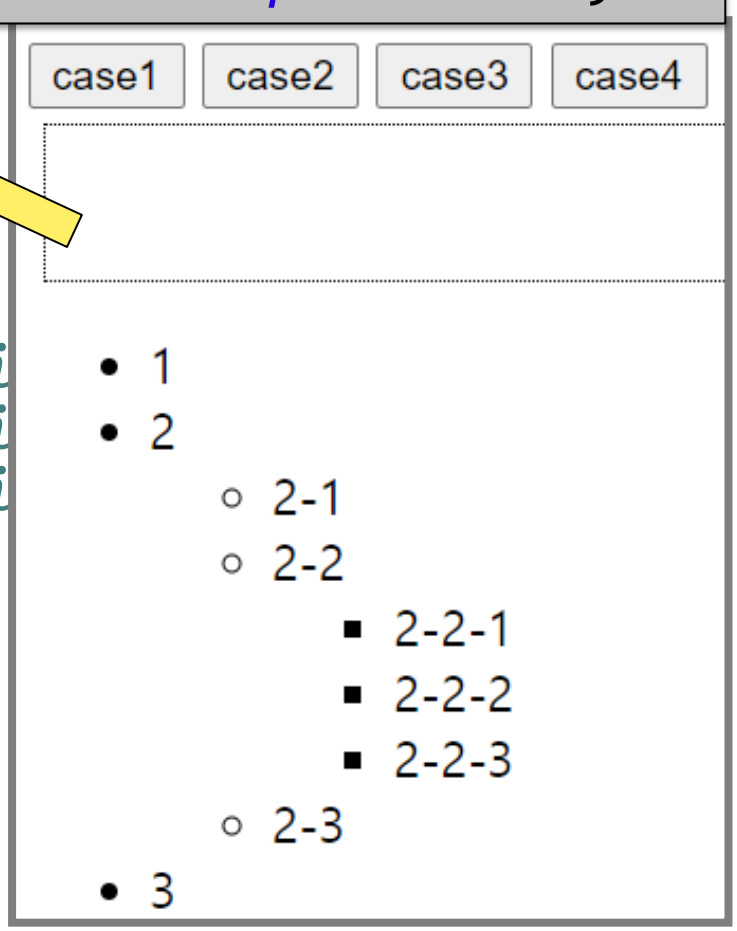
◆ $('ul.2-1').closest('li', ul_1).



closest(filter) 메소드 예제(2/3)

```
<button id="btnCase1">case1</button>
…<button id="btnCase4">case4</
<div id="divScript"></div>
<ul class="1" id="one">
  <li class="1">1</li>
  <li class="2" id="two">2
    <ul class="2-1">
      <li class="2-1">2-1</li>
      <li class="2-2">2-2
        <ul class="2-2-1">
          <li class="2-2-1">2-2-1</li
          <li class="2-2-2">2-2-2</li
          <li class="2-2-3">2-2-3</li
        </ul>
      </li>
      <li class="2-3">2-3</li>
    </ul>
  </li>
  <li class="3">3</li>
</ul>
```

```
div style
height:50px;margin:5px;
border:1px dotted;
```



closest(filter) 메소드 예제(3/3)

```
$(document).ready(function() {
  $("button").click(function() {
    $("ul, li").css('background-color', '');
    var strScript = "";

    var strId = $(this).attr("id");
    var li_2 = document.getElementById('two');//li 2

    if(strId == "btnCase1") {
      $('li.2-1').closest('ul').css('background-color', 'red');
      strScript = "$('li.2-1').closest('ul')";
    } else if(strId == "btnCase2") {
      $('li.2-1').closest('li').css('background-color', 'yellow');
      strScript = "$('li.2-1').closest('li')";
    } else if(strId == "btnCase3") {
      $('li.2-1').closest('ul', li_2).css('background-color', 'green');
      strScript = "$('li.2-1').closest('ul', li_2)";
    } else if(strId == "btnCase4") {
      var li_1 = document.getElementById('one');//li 2
      $('ul.2-1').closest('li', li_1).css('background-color', 'pink');
      $('ul.2-1').closest('li', li_2).css('background-color', 'lightblue');//listItem2의 하위요소까지만 검
    색.listItem2는 포함X
      strScript = "$('ul.2-1').closest('li', li_1)" + "<br>"
                + "$('ul.2-1').closest('li', li_2)";
    }

    $("#divScript").html(strScript);

  });
});
```

DOM 탐색 메소드(2/4)

2. 하위 요소 찾기	
$(*selector*) .children (*filter*)	선택한 요소의 자식 요소를 반환 (*filter*를 옵션으로 줄 수 있음)
$(*selector*).find(***filter***)	선택한 요소의 하위 요소에서 *filter*를 찾아서 모두 반환
3. 선택 요소 확장	
$(*selector*) .add(***element***, *context*)	***element***도 선택자로 추가 예) $("h1, p, #test").css("border", "1px"); ➔$("h1").add("p").add("#test").css("border", "1px");
4. 형제 요소 찾기	
$(*selector*) .siblings(*filter*)	선택한 요소의 모든 형제 요소를 반환(동일한 부모를 공유하는 요소)

DOM 탐색 메소드(3/4)

4. 형제 요소 찾기

$(*selector*) .next(*filter*)	-선택한 요소의 다음(뒤에 위치한) 형제 요소
$(*selector*) .nextAll(*filter*)	-선택한 요소의 다음 모든 형제 요소들
$(*selector*) .nextUntil(*stop*, *filter*)	-선택한 요소의 다음 모든 형제 요소. 단, *stop* 이전까지의 요소를 반환
$(*selector*) .prev(*filter*)	-선택한 요소의 이전(앞에 위치한) 형제 요소
$(*selector*) .prevAll(*filter*)	-선택한 요소의 이전 모든 형제 요소들(형제를 따라 역방향으로 탐색)
$(*selector*) .prevUntil(*stop*, *filter*)	-선택한 요소의 이전 모든 형제 요소. 단, *stop* 이전까지의 요소를 반환

DOM 탐색 메소드(4/4)

5. 필터링	
$(*selector*).first()	선택된 요소의 첫 번째 요소 반환
$(*selector*).last()	선택된 요소의 마지막 번째 요소
$(*selector*).eq(***index***)	선택된 요소 중 ***index*** 번째 요소
$(*selector*) .filter(*criteria*, *function(index)*)	선택된 요소 중 특정 조건(*criteria*)에 일치하는 요소들을 반환
$(*selector*) .not(*criteria*, *function(index)*)	선택된 요소 중 특정 조건(*criteria*)에 일치하는 요소를 제거하고 반환
$(*selector*) .is(***selectorElement***, *function(index,element)*)	선택된 요소 중 하나가 ***selectorElement***와 일치하는지 확인. true 또는 false를 리턴

is(*selector*) 메소드 예

```
<script>
$(document).ready(function(){
  $("p").css("background-color", "yellow");
  $("p").click(function(){
    if ($(this).parent().is("div")) {
      alert("클릭한 p의 부모는 div이다.");
    }
  });
});
</script>
```

```
<body>
  <p>Click 1</p>
  <div>
    <p>Click 2</p>
  </div>
</body>
```

찾기메소드(parent-is)

1) div로 감싸는 p태그와 그렇지 않은 p태그를 이용하여 문자열을 작성한다.
2) p태그를 클릭하면 부모가 div인지 판단하여, 맞다면 p태그를 복사하고 글자색을 빨강색으로 설정한 뒤 부모에 붙여넣는다 . (parent, is, clone, append)