



Chapter 06-3

탐색 선택자



탐색 선택자의 개요

탐색 선택자의 개요

■ 탐색 선택자

- 선택자에 의해 선택된 요소를 기준으로 새로운 요소를 찾아가는 방법
 - 탐색 선택자는 인접관계 선택자라고 할 수 있음
 - 선택자와 함께 메서드 형태로 사용됨
 - [예] `$('#div#container').prev().css('color', 'red')`
- 트리 구조 탐색을 수행함
 - 문서가 구성하는 DOM Tree를 기반으로 새로운 노드를 찾는 방식
 - 특정 노드를 기준으로 다른 노드를 검색함
- jQuery는 탐색을 위한 다수의 메서드를 제공함
 - 부모 요소, 형제 요소, 자식 요소를 탐색하기 위한 다양한 메서드 제공

탐색 선택자의 개요

■ 탐색 선택자의 종류

구분	메서드	설명
하위 요소 탐색	children()	선택된 요소의 자식 요소들 중 인자로 표현된 요소만을 선택 (자손 미포함)
	find()	선택된 요소의 자손 요소들 중 인자로 표현된 요소들을 모두 선택 (자손 자식 포함)
상위 요소 탐색	parent()	선택된 요소의 부모 요소를 선택
	parents()	선택된 요소의 부모 요소를 포함한 조상 요소들을 모두 선택
	closest()	선택된 요소를 포함해 인접한 부모 요소를 선택
형제 요소 탐색	next()	선택된 요소의 인접한 다음 형제 요소를 선택
	nextAll()	선택된 요소의 인접한 다음 형제 요소들을 모두 선택
	prev()	선택된 요소의 인접한 다음 이전 요소를 선택
	prevAll()	선택된 요소의 인접한 다음 형제 요소들을 모두 선택
	nextUntil(선택자)	인자로 가지는 조건에 만족할 때까지 다음 형제 요소들을 선택
	prevUntil()	인자로 가지는 조건에 만족할 때까지 이전 형제 요소들을 선택
	siblings()	선택된 요소의 형제 요소를 선택
	contents()	선택된 요소가 자식 요소를 가질 경우 해당 자식 요소를 선택

하위 요소 탐색 선택자

■ children() 메서드

- 선택된 요소의 자식 요소 중 인자로 지정된 자식 요소를 선택

```
$('요소').children( [자식 요소] )
```

- 인자를 가지지 않은 경우

- 모든 자식 요소를 선택 (자손 요소는 선택하지 않음)

- 인자를 가지는 경우

- 인자로 지정된 자식 요소만을 선택 (자손 요소는 선택하지 않음)

```
$('요소').children( 자식 요소 ) = $ ( ' 요소 > 자식 요소' )
```

```
<!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>
```

하위 요소 탐색 선택자

```
<body>

  <div id="wrap">

    <h1>children</h1>

    <p>내용1</p>

    <section>
      <h1>자식 요소 선택자</h1>
      <p>내용2</p>
    </section>
  </div>

</body>
</html>
```

children

내용1

자식 요소 선택자

내용2

하위 요소 탐색 선택자

■ find() 메서드

- 선택된 요소의 자식 요소 중 인자로 지정된 자손 요소를 선택

```
$('요소').find( [자손 요소] )
```

- 인자를 가지지 않은 경우
 - 자식 요소를 포함해 모든 자손 요소를 선택
- 인자를 가지는 경우
 - 자식 요소를 포함해 인자로 지정된 자손 요소들을 선택

하위 요소 탐색 선택자

6-search-find.htm (p.213)

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

    $(function(){
      $("#outer_wrap").find(".txt1")
        .css({"background-color":"#ff0"});
    });

  </script>
</head>
<body>
  <div id="outer_wrap">
    <h1>콘텐츠 탐색 선택자</h1>
    <section id="inner_1">
      <h2>find( )</h2>
      <p class="txt1">내용1</p>
      <p class="txt2">내용2</p>
    </section>
  </div>
</body>
</html>
```

콘텐츠 탐색 선택자

find()

내용1

내용2

하위 요소 탐색 선택자

6-search-children-find.htm

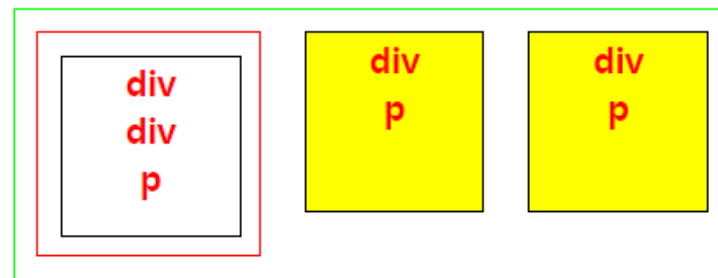
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <style type="text/css">
    div, p {
      float:left; margin:10px;
      background-color:#ffffff; border:#00ff00 thin solid;
    }
    div#sub { width:100px; height:100px; border:#ff0000 thin solid; }
    p { width:80px; height:80px ; border:#000000 thin solid; }
  </style>
  <script type="text/javascript">

    $(function(){
      $("div#container").children("p").css("backgroundColor","yellow");
      $("div#container").children().css("color","red");
      $("div#container").find("p").css("font-weight","bold");
    });

  </script>
</head>
```

하위 요소 탐색 선택자

```
<body>  
  
  <div id="container">  
  
    <div id = "sub">  
      <p align="center">div<br>div<br>p</p>  
    </div>  
  
    <p align="center">div<br>p</p>  
    <p align="center">div<br>p</p>  
  
  </div>  
  
</body>  
</html>
```



상위 요소 탐색 선택자

■ parent() 메서드와 parents() 메서드

• parent() 메서드

- 선택된 요소의 부모 요소를 선택하는 메서드

```
$('요소').parent( [요소] )
```

- 인자로 선택자를 포함할 수 있으나 잘 사용되지는 않음
 - 부모 요소가 인자로 지정된 요소와 동일하지 않을 경우 선택하지 않음
 - 선택된 요소는 자신은 선택에 포함되지 않음

• parents() 메서드

```
$('요소').parent2( [요소] )
```

- 선택된 요소의 부모 요소를 포함하는 조상 요소를 모두 선택하는 메서드
 - 인자로 선택자를 포함할 수 있으며, 이 경우 지정된 조상 요소를 선택
- 선택된 요소는 자신은 선택에 포함되지 않음

상위 요소 탐색 선택자

6-search-parent-parents htm

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

  <script type="text/javascript">
    $(function(){
      $("li#ch1-1").parent().css("backgroundColor","yellow");
      $("li#ch1-1").parents("ol").css("color","red");
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li id="ch">
        충청도
        <ol>
          <li id="ch1">충청북도
            <ul>
              <li id="ch1-1">청주시</li>
              <li id="ch1-2">충주시</li>
            </ul>
          </li>
        </ol>
      </li>
    </ul>
  </div>
</body>
</html>
```

상위 요소 탐색 선택자

```
<li id="ch2">충청남도
  <ul>
    <li id="ch2-1">대전시</li>
    <li id="ch2-2">천안시</li>
  </ul>
</li>
</ol>
</li>
<li>경기도</li>
</ul>
</div>
</body>
</html>
```

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

상위 요소 탐색 선택자

■ closet() 메서드

- 선택된 요소의 조상 요소 중에서 인자로 가지는 요소를 선택하되 가까운 요소 하나만을 선택

```
$('요소').closest( 요소 )
```

- parents() 메서드와 closest() 메서드의 차이점

- parents() 메서드

- 상위 요소들 중 인자로 지정된 모든 요소들을 선택

- closest() 메서드

- 상위 요소들 중 인자로 지정된 가장 가까운 요소 하나를 선택

상위 요소 탐색 선택자

6-search-closet-1.htm (p. 189)

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

    $(function(){
      $(".txt1").closest("div")
        .css({
          "border":"2px solid #f00"
        });
    });

  </script>
</head>
<body>
  <div>
    <h1 class="title">선택자</h1>
    <div>
      <p class="txt1">내용</p>
    </div>
  </div>
</body>
</html>
```

선택자

내용

상위 요소 탐색 선택자

6-search-closet-2.htm

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

  <script type="text/javascript">
    $(function(){
      $("li#ch1-1").closest('ul').css("color","red");
      $("li#ch1-1").parents('ul').css("backgroundColor","yellow");
    });
  </script>
</head>
<body>
  <div>
    <ul>
      <li id="ch">
        충청도
        <ol>
          <li id="ch1">충청북도
            <ul>
              <li id="ch1-1">청주시</li>
              <li id="ch1-2">충주시</li>
            </ul>
          </li>
        </ol>
      </li>
    </ul>
  </div>
</body>
</html>
```

상위 요소 탐색 선택자

```
<li id="ch2">충청남도
  <ul>
    <li id="ch2-1">대전시</li>
    <li id="ch2-2">천안시</li>
  </ul>
</li>
</ol>
</li>
<li>경기도</li>
</ul>
</div>
</body>
</html>
```

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

형제 요소 탐색 선택자

■ next() 메서드

- 선택된 요소의 바로 다음 형제 요소를 선택하는 메서드

```
$('요소').next( [요소] )
```

- 인자를 가지지 않는 경우
 - 요소의 종류와는 무관하게 선택된 요소의 바로 다음 형제 요소를 선택
- 인자로 선택자를 가지는 경우
 - 바로 다음에 인접한 형제 요소가 인자로 가지는 요소와 동일한 경우에만 선택

형제 요소 탐색 선택자

■ nextAll() 메서드

- 선택된 요소 다음의 모든 형제 요소들을 선택하는 메서드

```
$('요소'). nextAll( [요소] )
```

- 인자를 가지지 않는 경우
 - 요소의 종류와는 무관하게 선택된 요소의 다음의 형제 요소들을 모두 선택
- 인자로 선택자를 가지는 경우
 - 선택된 요소의 다음 형제 노드들 중 인자로 가지는 요소와 동일한 요소들을 모두 선택

형제 요소 탐색 선택자

6-search-next-nextAll.htm

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

    $(function(){
      $("p.part-1").next("div").css("backgroundColor","yellow");
      $("p.part-3").nextAll("div").css("backgroundColor", "gray");
      $("p.part-1").nextAll().css("color","red");
    });

  </script>
</head>
<body>
  <p class="part-1"> (p 요소) Hello part-1 </p>
  <div class="part-2"> (div 요소) Hello part-2 </div>
  <p class="part-3"> (p 요소) Hello part-3 </p>
  <p class="part-4"> (p 요소) Hello part-4 </p>
  <div class="part-5"> (div 요소) Hello part-5 </div>
  <p class="part-6"> (p 요소) Hello part-6 </p>
  <div class="part-7"> (div 요소) Hello part-7 </div>
</body>
</html>
```

(p 요소) Hello part-1

(div 요소) Hello part-2

(p 요소) Hello part-3

(p 요소) Hello part-4

(div 요소) Hello part-5

(p 요소) Hello part-6

(div 요소) Hello part-7

형제 요소 탐색 선택자

■ nextUntil() 메서드

- 선택된 요소를 기준으로 지정된 다음 형제 요소까지 만을 선택하는 메서드

```
$('요소').nextUntil( [요소1] [,요소2] )
```

- 선택된 요소 자신은 선택에 포함되지 않음
- [요소1]과 [요소2]는 생략할 수 있음
 - [요소1]과 [요소2]가 모두 지정된 경우
 - 선택된 요소를 기준으로 [요소1]까지의 범위 내에서 [요소2]의 요소만을 선택
 - [요소1]만 지정한 경우
 - 선택된 요소를 시작으로 [요소1] 직전까지의 요소를 반환 (요소1은 포함되지 않음)
 - [요소1]과 [요소2]가 모두 생략된 경우
 - nextAll() 메서드와 동일한 기능 수행

형제 요소 탐색 선택자

6-search-next-nextUntil.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#prov-1").nextUntil("dt#prov-2").css("backgroundColor", "yellow");
      $("#prov-1").nextUntil("dt#prov-2", "#stop").css("color", "red");
    });
  </script>
</head>
<body>
  <dl>
    <dt id="prov-1">충청북도</dt>
    <dd>청주시</dd>
    <dd id="stop">충주시</dd>
    <dd>제천시</dd>
    <dt id="prov-2">충청남도</dt>
    <dd>대전시</dd>
    <dd>천안시</dd>
  </dl>
</body>
</html>
```

충청북도
청주시
충주시
제천시
충청남도
대전시
천안시

형제 요소 탐색 선택자

■ prev() 메서드

- 선택된 요소의 바로 이전 형제 요소를 선택하는 메서드

```
$('요소').prev( [요소] )
```

- 인자를 가지지 않는 경우
 - 요소의 종류와는 무관하게 선택된 요소의 바로 이전 형제 요소를 선택
- 인자로 선택자를 가지는 경우
 - 바로 이전에 인접한 형제 요소가 인자로 가지는 요소와 동일한 경우에만 선택

형제 요소 탐색 선택자

■ prevAll() 메서드

- 선택된 요소 다음의 모든 이전 요소들을 선택하는 메서드

```
$('요소').prevAll( [요소] )
```

- 인자를 가지지 않는 경우
 - 요소의 종류와는 무관하게 선택된 요소의 이전 형제 요소들을 모두 선택
- 인자로 선택자를 가지는 경우
 - 선택된 요소의 이전 형제 노드들 중 인자로 가지는 요소와 동일한 요소들을 모두 선택

형제 요소 탐색 선택자

6-search-prev-prevAll.htm

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

    $(function(){
      $("p.part-6").prev("div").css("backgroundColor","yellow");
      $("p.part-6").prevAll("p").css("backgroundColor", "gray");
      $("p.part-6").prevAll().css("color","red");
    });

  </script>
</head>
<body>
  <p class="part-1"> (p 요소) Hello part-1 </p>
  <div class="part-2"> (div 요소) Hello part-2 </div>
  <p class="part-3"> (p 요소) Hello part-3 </p>
  <p class="part-4"> (p 요소) Hello part-4 </p>
  <div class="part-5"> (div 요소) Hello part-5 </div>
  <p class="part-6"> (p 요소) Hello part-6 </p>
  <div class="part-7"> (div 요소) Hello part-7 </div>
</body>
</html>
```

(p 요소) Hello part-1

(div 요소) Hello part-2

(p 요소) Hello part-3

(p 요소) Hello part-4

(div 요소) Hello part-5

(p 요소) Hello part-6

(div 요소) Hello part-7

형제 요소 탐색 선택자

■ prevUntil() 메서드

- 선택된 요소를 기준으로 지정된 이전 형제 요소까지 만을 선택하는 메서드
- 선택된 요소 자신은 선택에 포함되지 않음
- [요소1]과 [요소2]는 생략할 수 있음
 - [요소1]과 [요소2]가 모두 지정된 경우
 - 선택된 요소를 기준으로 [요소1]까지의 범위 내에서 [요소2]의 요소만을 선택
 - [요소1]만 지정한 경우
 - 선택된 요소를 시작으로 [요소1] 직전까지의 요소를 반환 (요소1은 포함되지 않음)
 - [요소1]과 [요소2]가 모두 생략된 경우
 - prevAll() 메서드와 동일한 기능 수행

형제 요소 탐색 선택자

6-search-prevUntil.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#prov-2").prevUntil("dt#prov-1").css("backgroundColor", "yellow");
      $("#prov-2").prevUntil("dt#prov-1", "#stop").css("color", "red");
    });
  </script>
</head>
<body>
  <dl>
    <dt id="prov-1">충청북도</dt>
    <dd>청주시</dd>
    <dd id="stop">충주시</dd>
    <dd>제천시</dd>
    <dt id="prov-2">충청남도</dt>
    <dd>대전시</dd>
    <dd>천안시</dd>
  </dl>
</body>
</html>
```

충청북도

청주시

충주시

제천시

충청남도

대전시

천안시

형제 요소 탐색 선택자

■ siblings() 메서드

- 선택된 요소 자신은 제외한 모든 형제 요소를 선택

`$("요소").siblings()`

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("#second").siblings().css("background", "red");
    });
  </script>
</head>
<body>
  <ul>
    <li id="first">내용 1</li>
    <li id="second">내용 2</li>
    <li id="third">내용 3</li>
  </ul>
</body>
</html>
```

6-search-sibling.htm

- 내용 1
- 내용 2
- 내용 3

형제 요소 탐색 선택자

■ contents() 메서드

- 선택된 요소가 자식 요소를 가질 경우 해당 자식 요소를 선택

```
$('요소').contents( [요소] )
```

- 자식 요소가 텍스트 만일 경우에는 선택되지 않음
- 요소를 생략할 경우
 - 선택된 요소의 자식요소들을 선택
- 요소를 생략하지 않을 경우
 - 선택된 요소의 자식요소들을 중 인자로 가지는 자식 요소만을 선택

형제 요소 탐색 선택자

6-search-contents.htm

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

      $('li').contents().css('backgroundColor','yellow');
      $('li').contents('span').css('color','red');

    });
  </script>
</head>
<body>
  <ul class="first">
    <li>list item 1</li>
    <li>
      <span><b>list item 1</b></span>
      <strong>list item 1-1</strong>
    </li>
    <li>list item 3</li>
  </ul>
</body>
</html>
```

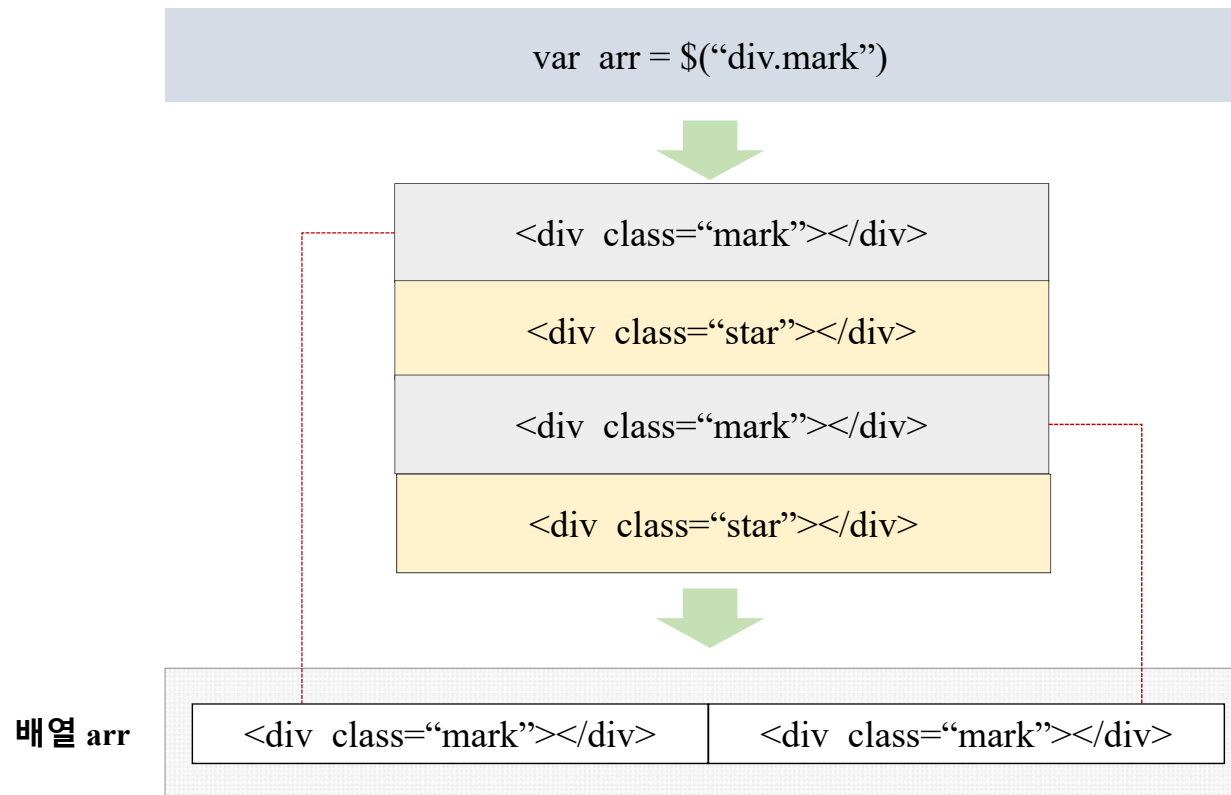
- list item 1
- **list item 1** **list item 1-1**
- list item 3

배열 관련 메서드

jQuery와 배열

■ jQuery와 배열

- 선택자에 의해 선택된 하나 이상의 요소들은 배열의 형태로 유지됨



배열 관련 메서드

■ 배열 관련 메서드

메서드	설명
each() / \$.each()	배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨
\$.map()	배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨 - 반환되는 데이터는 새로운 배열을 생성함
\$.grep()	배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨 - 반환 값이 TRUE인 경우에만 그에 해당하는 데이터가 새로운 배열을 생성함
\$.inArray()	배열 내에서 인자로 지정한 데이터를 검색해 그 인덱스를 반환
\$.isArray()	인자가 배열인 경우 true를 반환
\$.merge()	인자로 지정한 2개의 배열을 하나의 배열로 결합
index(요소)	인자로 지정된 선택자에 해당하는 요소를 찾아 그 index 값을 반환

배열 관련 메서드

■ each() 메서드

- 배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨

```
$("요소").each( callback function{ } )
```

- 선택자에 의해서 선택된 문서 객체들은 배열로 유지됨
- each() 메서드는 생성된 배열의 각 요소에 대해 동일한 연산을 순차적으로 수행
 - 배열의 각 요소에 저장된 문서 객체들에 대해 each() 메서드의 인자로 지정된 콜백 함수를 수행
 - 콜백 함수는 배열의 상위 요소부터 하위 요소 순서로 처리됨

배열 관련 메서드

6-array-each-1.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
    div { color:red; cursor:pointer; font-weight:bolder; cursor:pointer; }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $(document).click(function () {
        $("div").each(function () {
          if ( this.style.color != "blue" ) {
            this.style.color = "blue";
          } else {
            this.style.color = "red";
          }
        });
      });
    });
  </script>
</head>
```

배열 관련 메서드

```
<body>
```

```
  <div>KOREA</div>
```

```
  <div>USA</div>
```

```
  <div>JAPAN</div>
```

```
</body>
```

```
</html>
```

KOREA
USA
JAPAN

KOREA
USA
JAPAN

```
$(function(){
```

```
  $("div").click(function () {  
    if ( this.style.color != "blue" ) {  
      this.style.color = "blue";  
    } else {  
      this.style.color = "";  
    }  
  });  
});
```

```
});
```

KOREA
USA
JAPAN

KOREA
USA
JAPAN

배열 관련 메서드

■ each() 메서드의 콜백 함수가 index를 인자로 가질 경우

```
$("요소").each( callback function( index ) { } )
```

- each() 메서드에서 사용되는 콜백 함수는 인덱스(index)를 인자로 가질 수 있음
- index는 배열에서 첨자와 같은 역할을 수행함
 - index는 0부터 시작하여 배열이 크기보다 1이 작은 정도만큼 증가함
 - 하나의 요소에 콜백 함수가 적용될 때마다 1씩 증가함
- 선택된 요소들을 콜백 함수를 사용해 제어해야 하는 경우 사용

배열 관련 메서드

6-array-each-3.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
    div {
      width:60px; height:60px;
      margin:5px; float:left;
      border:2px blue solid;
    }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $("button").click(function() {
        $("div#class-a").each(function (index) {
          $(this).css("backgroundColor","yellow");
          $(this).html(index);
        });
      });
    });
  </script>
</head>
```

배열 관련 메서드

```
<body>
```

```
  <div id="class-a"></div>
```

```
  <div id="class-a"></div>
```

```
  <div id="class-b"></div>
```

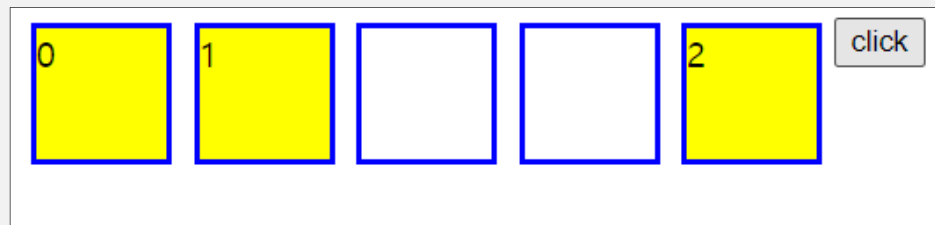
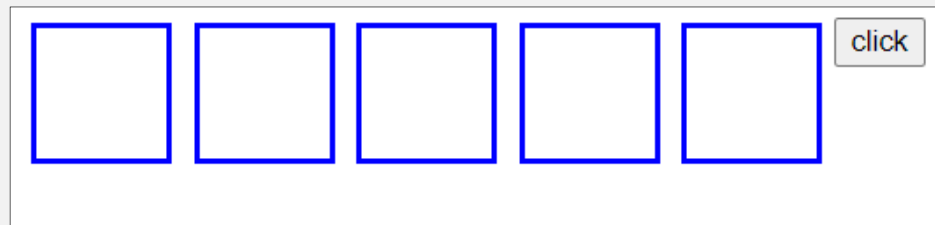
```
  <div id="class-b"></div>
```

```
  <div id="class-a"></div>
```

```
  <button>click</button>
```

```
</body>
```

```
</html>
```



배열 관련 메서드

■ each() 메서드의 콜백함수가 index와 element를 인자로 가질 경우

```
$("요소").each( callback function( index, item ) { } )
```

- 콜백 함수는 인덱스와 요소를 나타내는 인자를 함께 가질 수 있음
- index
 - 요소의 인덱스를 의미 (0 ~ (요소의 크기-1))
- item
 - 요소의 인덱스에 해당하는 요소를 의미
 - \$(this)로도 표현 가능

배열 관련 메서드

6-array-each-4.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
    div {
      width:40px; height:40px; margin:5px; float:left; border:2px blue solid; text-align:center;
    }
    span { color:red; }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $( "button" ).click(function () {
        $("div").each(function ( index, element ) {
          $(element).css( "backgroundColor", "yellow" );
          if ( $(this).is( "#stop" ) ) {
            $( "p" ).text( "Stopped at div index #" + index );
            return false;
          }
        });
      });
    });
  </script>
</head>
```

배열 관련 메서드

```
<body>
```

```
  <button>Change colors</button>
```

```
  <p> </p>
```

```
  <div>0</div>
```

```
  <div>1</div>
```

```
  <div>2</div>
```

```
  <div id="stop">Stop here</div>
```

```
  <div>4</div>
```

```
  <div>5</div>
```

```
  <div>6</div>
```

```
</body>
```

```
</html>
```

Change colors

0	1	2	Stop here	4	5	6
---	---	---	--------------	---	---	---

Change colors

Stopped at div index #3

0	1	2	Stop here	4	5	6
---	---	---	--------------	---	---	---

배열 관련 메서드

■ \$.each() 메서드

- 배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨

```
$.each( 요소, callback function( index, element ) { } )
```

- each() 메서드와 동일한 기능을 수행하나 사용 방법이 다름
- 첫 번째 인자인 요소에 대해 두 번째 인자인 콜백 함수를 적용
 - **요소**
 - jQuery 객체
 - 배열일 경우 각 요소에 대해 두 번째 인자인 함수(function)이 적용됨
 - **function(index, item)**
 - index - 객체의 인덱스를 의미 (0 ~ (객체의 크기-1))
 - item - 객체의 인덱스에 해당하는 요소를 의미 (\$(this)로도 표현 가능)

배열 관련 메서드

6-array-each-5.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
    div {
      width:40px; height:40px; margin:5px; float:left; border:2px blue solid; text-align:center;
    }
    span { color:red; }
  </style>
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      $( "button" ).click(function () {
        $.each( $('div'), function( index, element ) {
          $(this).css( "backgroundColor", "yellow" );
          if ( $(this).is( "#stop" ) ) {
            $( "p" ).text( "Stopped at div index #" + index );
            return false;
          }
        });
      });
    });
  </script>
</head>
```

배열 관련 메서드

```
<body>
```

```
  <button>Change colors</button>
```

```
  <p> </p>
```

```
  <div>0</div>
```

```
  <div>1</div>
```

```
  <div>2</div>
```

```
  <div id="stop">Stop here</div>
```

```
  <div>4</div>
```

```
  <div>5</div>
```

```
  <div>6</div>
```

```
</body>
```

```
</html>
```

Change colors

0	1	2	Stop here	4	5	6
---	---	---	--------------	---	---	---

Change colors

Stopped at div index #3

0	1	2	Stop here	4	5	6
---	---	---	--------------	---	---	---

배열 관련 메서드

■ \$.map() 메서드

- 배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨

```
var arr = $.map( 요소, callback function( element, index ) { } )
```

- 반환되는 데이터는 새로운 배열을 생성함

- map() 메서드도 동일한 기능을 수행

```
var arr = $(요소).map( callback function( element, index ) { } )
```

배열 관련 메서드

6-array-map.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      var arr1 = [
        {area:"서울",name:"무과장"},
        {area:"부산",name:"홍과장"},
        {area:"대전",name:"박사장"},
        {area:"서울",name:"빅마마"},
      ]

      var arr2 = $.map(arr1, function(item, index){
        if(item.area=="서울") {
          return item;
        }
      });

      $.each( arr2, function( index, item ) {
        var output = item.area + '-' + item.name + "<br>";
        document.body.innerHTML += output;
      });
    });
  </script>
</head>
<body> </body>
</html>
```

서울-무과장
서울-빅마마

배열 관련 메서드

■ \$.grep() 메서드

- 배열에 저장된 각 요소에 대해 인자로 지정된 콜백 함수가 반복하여 수행됨

```
var arr = $.map( 요소, callback function( element, index ) { } )
```

- 반환 값이 TRUE인 경우에만 그에 해당하는 데이터가 새로운 배열을 생성함

배열 관련 메서드

6-array-grep.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      var arr1 = [
        {area:"서울",name:"무과장"},
        {area:"부산",name:"홍과장"},
        {area:"대전",name:"박사장"},
        {area:"서울",name:"빅마마"},
      ]

      var arr2 = $.grep(arr1, function(item, index){
        if(item.area=="서울") {
          return true;
        }
      });

      $.each( arr2, function( index, item ) {
        var output = item.area + '-' + item.name + "<br>";
        document.body.innerHTML += output;
      });
    });
  </script>
</head>
<body> </body>
</html>
```

서울-무과장
서울-빅마마

배열 관련 메서드

■ \$.isArray() 메서드와 \$.isArray() 메서드

• \$.isArray() 메서드와

- 배열 내에서 인자로 지정한 데이터를 검색해 그 인덱스를 반환

```
var arr = $.isArray( 데이터, 배열, 시작 index )
```

- 데이터: 찾고자 하는 데이터
- 배열: 검색 대상인 배열
- 시작 index: 0 부터 시작하는 검색의 시작 index

• \$.isArray() 메서드

- 인자가 배열인 경우 true를 반환

```
var arr = $.isArray( 배열 )
```

배열 관련 메서드

■ \$.merge(배열1, 배열2)

- 인자로 가지는 배열을 결합해 하나의 배열을 생성

```
var arr = $.merge( 배열1, 배열2 )
```

- 배열1과 배열2를 결합해 새로운 배열 arr을 생성

■ index() 메서드

- 인자로 지정된 선택자가 나타내는 요소의 index 값을 반환

```
$( '요소' ).index( 선택자 )
```

배열 관련 메서드

6-array-inArray-isArray.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script src="script/jquery-3.5.1.js"></script>
  <script type="text/javascript">
    $(function(){
      var arr1 = ['서울', '부산', '대전', '전주'];
      var arr2 = ['한국', '미국', '일본', '중국'];

      if( $.isArray(arr1) ) {
        var num1 = $.inArray('부산',arr1);
        if( num1 != -1 ) {
          alert('부산의 인덱스는' + num1 + '입니다.')
        }
      }

      var arr3 = $.merge(arr1, arr2);
      alert(JSON.stringify(arr3));

    });
  </script>
</head>
<body> </body>
</html>
```

localhost:8080 내용:

부산의 인덱스는1입니다.

확인

localhost:8080 내용:

["서울","부산","대전","전주","한국","미국","일본","중국"]

확인

배열 관련 메서드

6-array-index.htm

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

      var num = $('li').index( $('#list3') );
      alert(num);

    });
  </script>
</head>
<body>
  <ul>
    <li>내용1</li>
    <li>내용2</li>
    <li id='list3'>내용3</li>
    <li>내용4</li>
  </ul>
</body>
</html>
```

- 내용1
- 내용2
- 내용3
- 내용4

localhost:8080 내용:

2

확인

수고하셨습니다