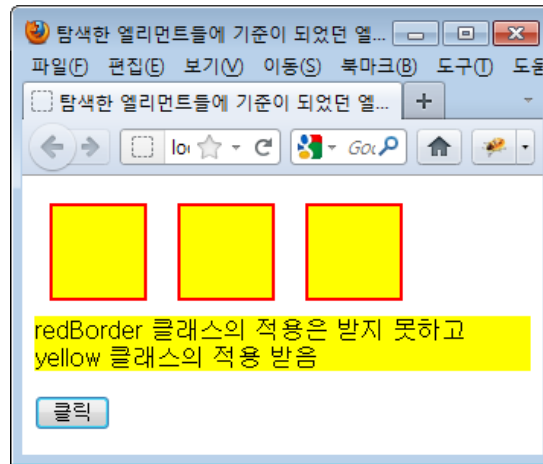
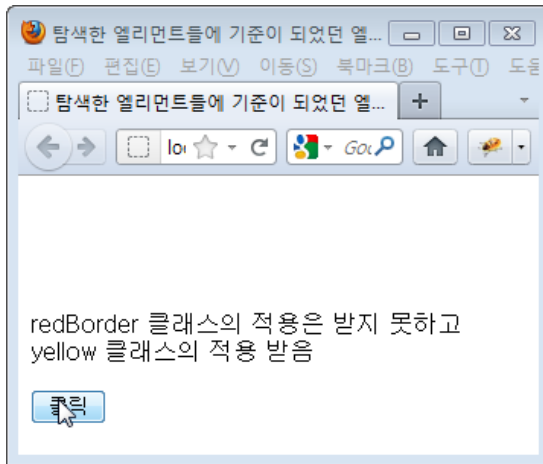


DOM 탐색하기

강사 : 강병준

엘리먼트들 추가하기

```
$("#div").addClass('redBorder')  
  .add("p").addClass('yellow');
```



- div 엘리먼트에 redBorder 클래스를 적용하고 p 엘리먼트를 더하여 yellow 클래스를 적용한다.
- 결과는 div 엘리먼트에는 redBorder, yellow 두개의 클래스가 적용되고, p 엘리먼트에는 yellow 클래스만 적용이 된다.

엘리먼트들 추가하기

■ add(expression!)

add(selector 혹은 elements 혹은 htmlString)	
설명	jQuery 확장 객체 집합에 새로운 엘리먼트를 추가한다.
매개변수	selector : jQuery 확장 객체 집합에 추가할 엘리먼트를 찾는 셀렉터 표현식 elements : jQuery 확장 객체 집합에 추가할 엘리먼트 htmlString : 새로 만든 후 추가할 엘리먼트의 HTML 코드
반환값	jQuery 확장 객체 집합

엘리먼트들 추가하기

```
<style>
```

```
div { width: 60px; height: 60px; margin: 10px; float: left;}
```

```
p{ clear:both;}
```

```
.redBorder { border: 2px solid red}
```

```
.yellow { background: yellow}
```

```
</style>
```

```
<script src="../js/jquery.js" type="text/javascript"></script>
```

```
<script type="text/javascript">
```

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

```
  $("button").click(function() {
```

```
    $("div").addClass('redBorder')
```

```
    .add("p")
```

```
    .addClass('yellow');
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

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

```
  <p> redBorder 클래스의 적용은 받지 못하고 yellow 클래스의 적용 받음</p>
```

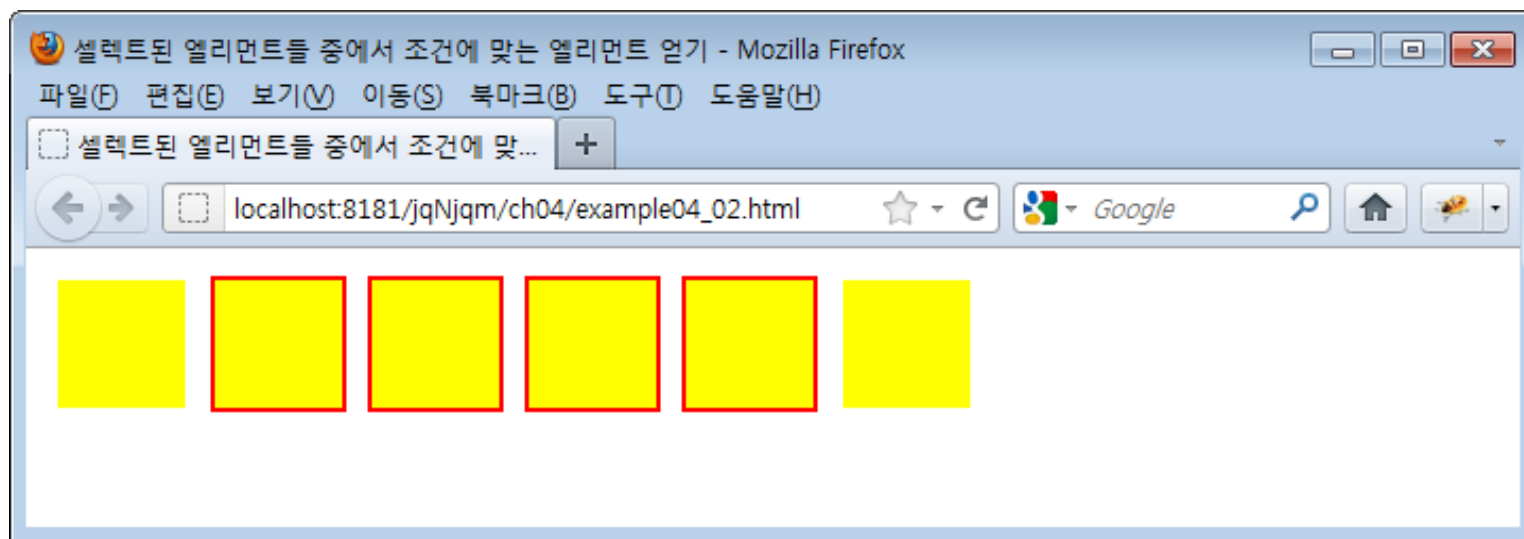
```
  <button> 클릭 </button>
```

```
</body>
```

```
</html>
```

엘리먼트 얻기

```
$("#div").css("background", "#c8ebcc")  
    .filter(".middle") .css("border-color", "red");
```



- 모든 div 엘리먼트에 "background", "#c8ebcc" 클래스를 적용하고 나서, div 엘리먼트 중에 클래스명이 ".middle"인 것만 골라내서 "border-color", "red" 클래스를 적용한다.

엘리먼트 얻기

- **filter(fn)**

filter(selector 혹은 function)	
설명	매개변수에 전달된 셀렉터 표현식 혹은 함수에 만족하는 엘리먼트를 추출(필터링)한다.
매개변수	selector : 엘리먼트를 추출하기 위한 셀렉터 표현식 function : 엘리먼트를 추출하기 위한 함수
반환값	jQuery 확장 객체 집합

엘리먼트 얻기

```
<style>
div {    width: 60px; height: 60px;
        margin: 5px; float: left;
        border: 2px white solid;
}
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $("div").css("background", "yellow")
        .filter(".middle")
        .css("border-color", "red");
    });
</script></head><body>
    <div id="first"></div>

    <div id="second" class="middle"></div>
    <div id="third" class="middle"></div>
    <div id="fourth" class="middle"></div>
    <div id="fifth" class="middle"></div>

    <div id="sixth"></div>

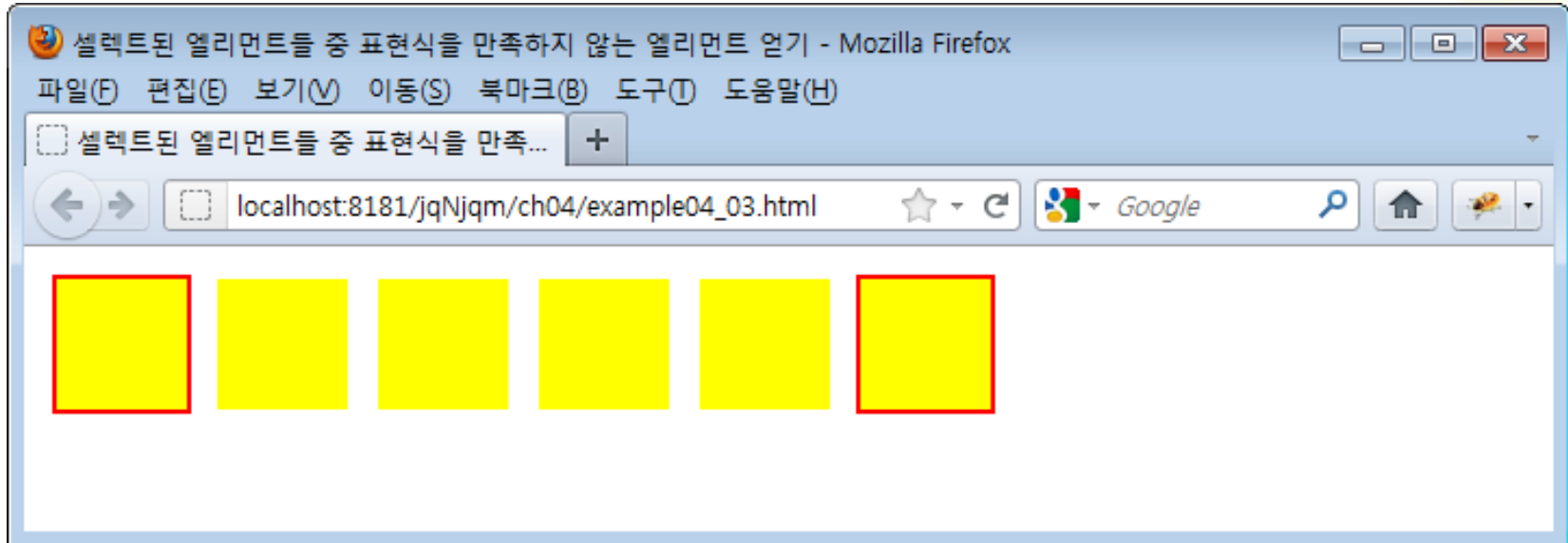
</body>
</html>
```

엘리먼트 얻기

```
<style>
div {      width: 60px; height: 60px;
           margin: 5px; float: left;
           border: 2px white solid; }
</style>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $("div").css("background", "yellow")
        .filter(function(index) {
            return index == 1 || $(this).attr("id") == "fourth";
        })
        .css("border-color", "red");
    });
</script></head>
<body>
    <div id="first"></div>
    <div id="second" class="middle"></div><div id="third" class="middle"></div>
    <div id="fourth" class="middle"></div><div id="fifth" class="middle"></div>
    <div id="sixth"></div>
</body></html>
```


엘리먼트 얻기

```
$(function() {  
    $("div").css("background", "#c8ebcc")  
    .not(".middle")  
    .css("border-color", "red");  
});
```



엘리먼트 얻기

not(expression!)

not(selector 혹은 elements 혹은 function)	
설명	매개변수에 전달된 셀렉터 표현식 혹은 함수에 만족하는 엘리먼트를 제거한다.
매개변수	<p>selector : 제거할 엘리먼트를 찾기 위한 셀렉터 표현식</p> <p>elements : 제거할 엘리먼트나 엘리먼트 집합</p> <p>function : 제거할 엘리먼트를 찾기 위한 함수</p>
반환값	jQuery 확장 객체 집합

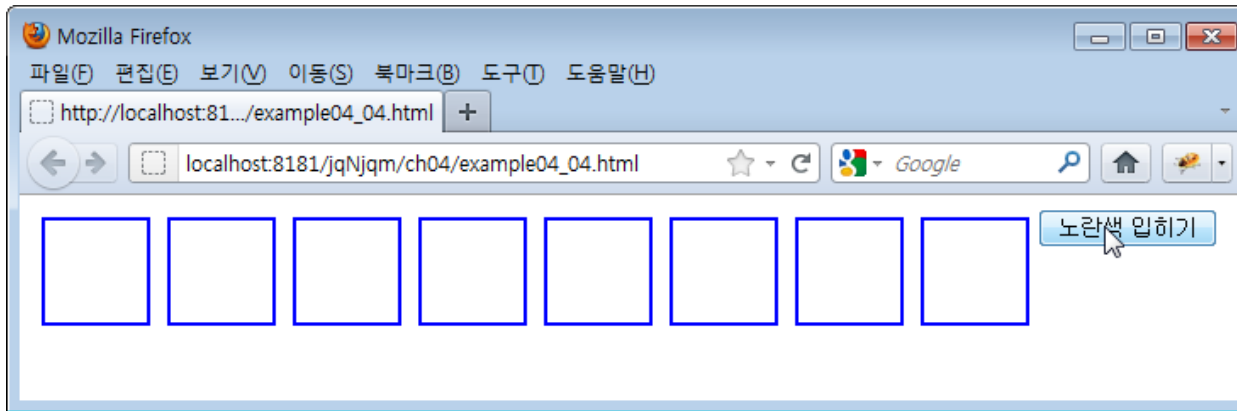
엘리먼트 얻기

```
<style>
div {      width: 60px; height: 60px;
           margin: 5px; float: left;
           border: 2px white solid;
}
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $("div").css("background", "yellow")
        .not('.middle')
        .css("border-color", "red");
    });
</script></head>
<body>
    <div id="first"></div>
    <div id="second" class="middle"></div>
    <div id="third" class="middle"></div>
    <div id="fourth" class="middle"></div>
    <div id="fifth" class="middle"></div>
    <div id="sixth"></div>

</body>
</html>
```

엘리먼트만 반환하기

```
$div.slice(start, end).css("background", "yellow");  
$div.slice(start).css("background", "yellow");
```



엘리먼트만 반환하기

slice(begin, end)

■ 확장집합의 특정 위치의 집합 반환

- begin 부터 시작하여 end 까지의 집합을 잘라낸다. 잘라낸 집합을 반환한다.
- begin 이상 end 미만으로 생각하면 쉽다.
- 매개변수
 - begin(number) : 첫 엘리먼트의 위치, 0부터 시작
 - end(number) : 마지막 엘리먼트의 위치, 생략한다면 마지막까지 포함한다.
- 반환값
 - 확장집합

엘리먼트만 반환하기

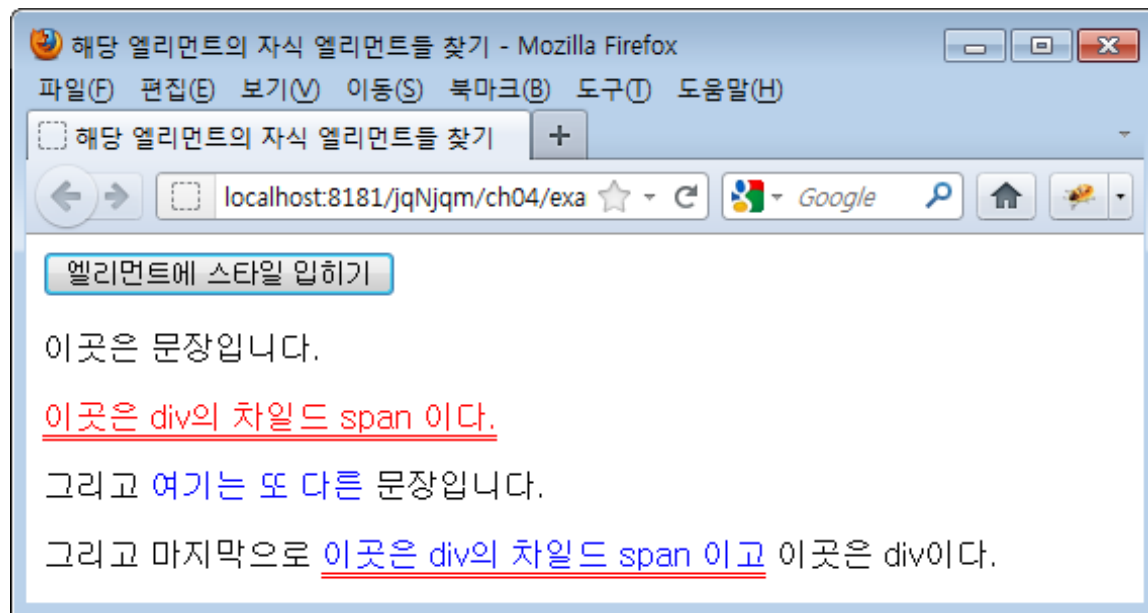
```
<style>
div {      width: 60px; height: 60px;
           margin: 5px; float: left;
           border: 2px blue solid; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $('button').click(function(){
            //셀렉터로 얻어낸 확장 집합 객체가 여러 번 사용될 경우에는 변수에 저장해 둔다.
            var $div = $('div');
            //div 엘리먼트 중 2 이상 3미만, 인덱스가 0부터 시작하기에 세번째 엘리먼트만 반환
            $div.slice(2, 3).css('background', 'yellow');
            //end가 생략된 상태에서 begin만 기술하면 begin부터 끝까지
            $div.slice(4).css('background', 'pink');
        });
    });
</script></head>
<body>
    <div></div>      <div></div>      <div></div>      <div></div>
    <div></div>      <div></div>      <div></div>      <div></div>
    <button> 노란색 입히기</button>
</body></html>
```

해당 엘리먼트의 자식 엘리먼트들 찾기

children()

```
$("#div").children().css("border-bottom", "3px double red");
```

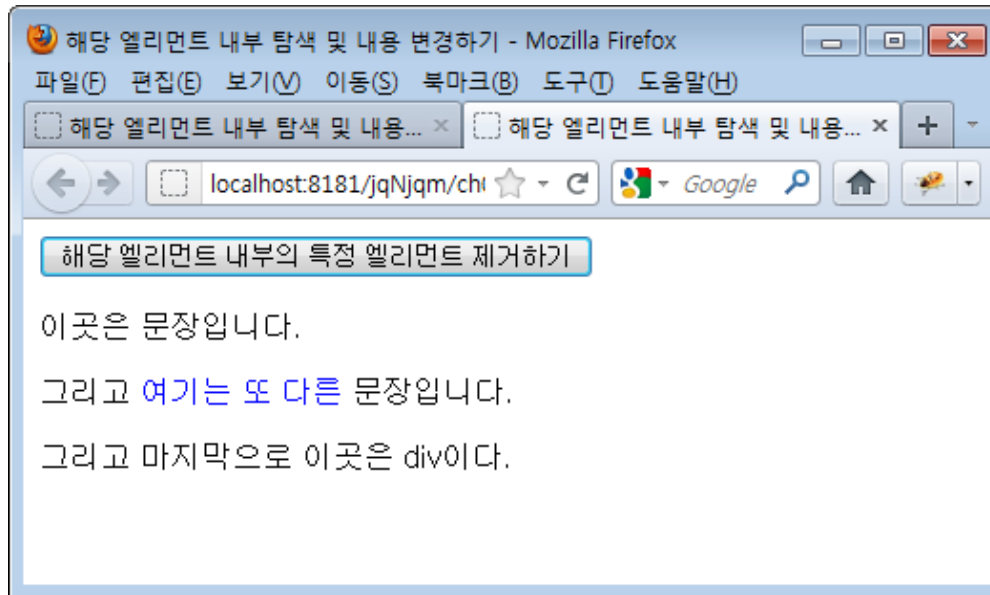
```
$("#div").children(".selected").css("color", "blue");
```



```
<!DOCTYPE html><html>
  <head><meta charset="UTF-8">
<title> 해당 엘리먼트의 자식 엘리먼트들 찾기 </title>
<style>  span { color: blue; } </style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
  $(function() {
    $('button').click(function(){
      $("div").children().css("border-bottom", "3px double red");
      $("div").children(".selected").css("color", "red");
    });
  });
</script></head>
<body>
  <button> 엘리먼트에 스타일 입히기</button>
  <p> 이곳은 문장입니다. </p>
  <div> <span class="selected"> 이곳은 div의 차일드 span 이다.</span></div>
  <p> 그리고 <span> 여기는 또 다른 </span> 문장입니다.</p>
  <div> 그리고 마지막으로
    <span> 이곳은 div의 차일드 span 이고</span>
    이곳은 div이다.
  </div>
</body>
</html>
```


해당 엘리먼트 내부 탐색 및 내용 변경하기

```
$("#div").contents().remove("span");
```



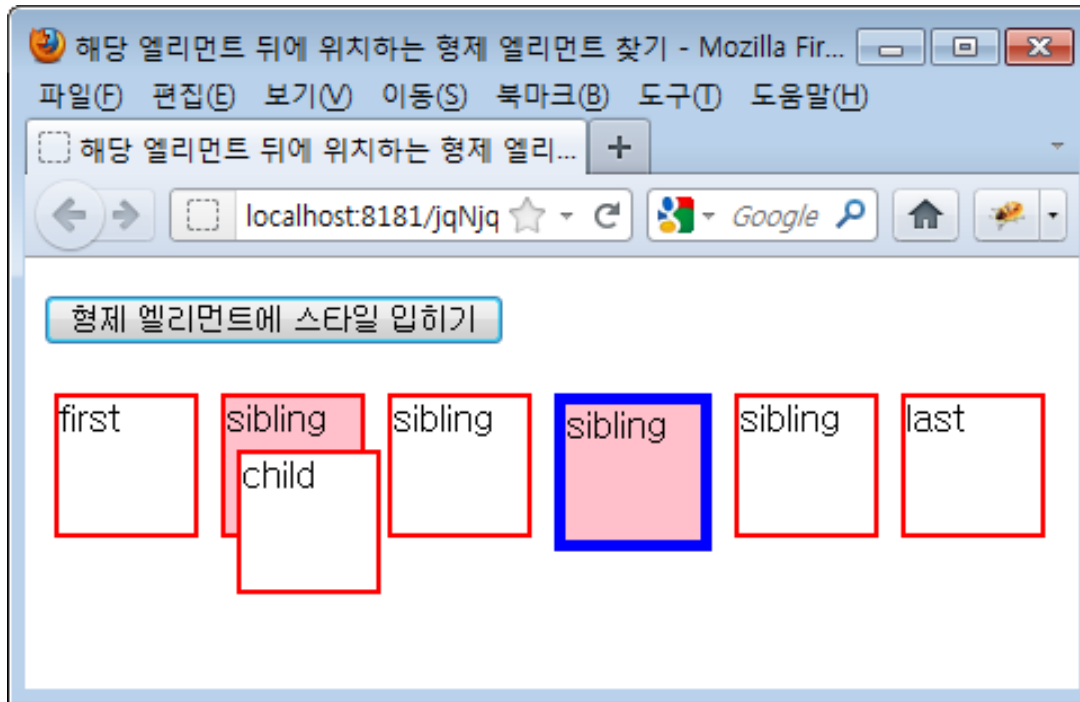
■ contents()

- text 노드를 포함한 모든 자손 엘리먼트를 가져 온다.
- div 엘리먼트의 text 노드와 자손 엘리먼트를 가져온 후에 span 엘리먼트만 삭제한다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title> 해당 엘리먼트 내부 탐색 및 내용 변경하기 </title>
<style> span { color: blue; }</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $('button').click(function(){
            $("div").contents().remove("span");
        });
    });
</script></head>
<body>
    <button> 해당 엘리먼트 내부의 특정 엘리먼트 제거하기</button>
    <p> 이곳은 문장입니다. </p>
    <div>
        <span class="selected"> 이곳은 div의 차일드 span 이다.</span>
    </div>
    <p>
        그리고 <span> 여기는 또 다른 </span> 문장입니다.
    </p>
    <div> 그리고 마지막으로
        <span> 이곳은 div의 차일드 span 이고</span>
        이곳은 div이다.
    </div>
</body>
</html>
```

해당 엘리먼트 뒤에 위치하는 형제 엘리먼트 찾기

```
$("#p").next().css("border-bottom", "3px double red");  
$("#p").next(".selected").css("background", "yellow");
```



해당 엘리먼트 뒤에 위치하는 형제 엘리먼트 찾기

- **`$("p").next()`**

- p 엘리먼트와 같은 노드 위치에 있는 인접한 엘리먼트 중 현재 엘리먼트 다음에 존재하는 엘리먼트를 찾는다.

- **`$("p").next(".selected")`**

- p 엘리먼트와 같은 노드 위치하며 바로 클래스명이 "selected"인 엘리먼트를 찾는다.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>해당 엘리먼트 뒤에 위치하는 형제 엘리먼트 찾기</title>
```

```
<style>
```

```
div, span {
```

```
    width: 60px; height: 60px;
```

```
    margin: 5px; float: left;
```

```
    border: 2px red solid;
```

```
    background: white;
```

```
}
```

```
.after { background: pink }
```

```
.selected {border: 5px blue solid;}
```

```
</style>
```

```
<script src="../js/jquery.js" type="text/javascript"></script>
```

```
<script type="text/javascript">
```

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

```
        $('button').click(function(){
```

```
            $("div").next().addClass("after");
```

```
            $("div").next("#fourth").addClass("selected");
```

```
        });
```

```
    });
```

```
</script>
```

```
</head>
```

<body>

<p> <button> 형제 엘리먼트에 스타일 입히기 </button> </p>

<div id="first">first</div>

**
 sibling
 <div id="child">child</div>**

<div id="third" class="middle">sibling</div>

sibling

sibling

<div id="sixth">last</div>

</body>

</html>

해당 엘리먼트 뒤에 있는 모든 엘리먼트 찾기

```
$("#div:first").nextAll().addClass("after");
```

■ nextAll()

- 대상 엘리먼트를 기준으로 현재 노드와 같거나 하위 엘리먼트를 찾는다.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>해당 엘리먼트 뒤에 위치하는 형제 엘리먼트 찾기</title>
```

```
<style>
```

```
div, span {
```

```
    width: 60px; height: 60px;
```

```
    margin: 5px; float: left;
```

```
    border: 2px red solid;
```

```
    background: white;
```

```
}
```

```
.after { background: pink }
```

```
.selected {border: 5px blue solid;}
```

```
</style>
```

```
<script src="../js/jquery.js" type="text/javascript"></script>
```

```
<script type="text/javascript">
```

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

```
        $('button').click(function(){
```

```
            $("div").nextAll().addClass("after");
```

```
            $("div").nextAll("span").addClass("selected");
```

```
        });
```

```
    });
```

```
</script>
```

```
</head>
```


<body>

<p> <button> 형제 엘리먼트에 스타일 입히기 </button> </p>

<div id="first">first</div>

**
sibling**

<div id="child">child</div>

<div id="third" class="middle">sibling</div>

sibling

sibling

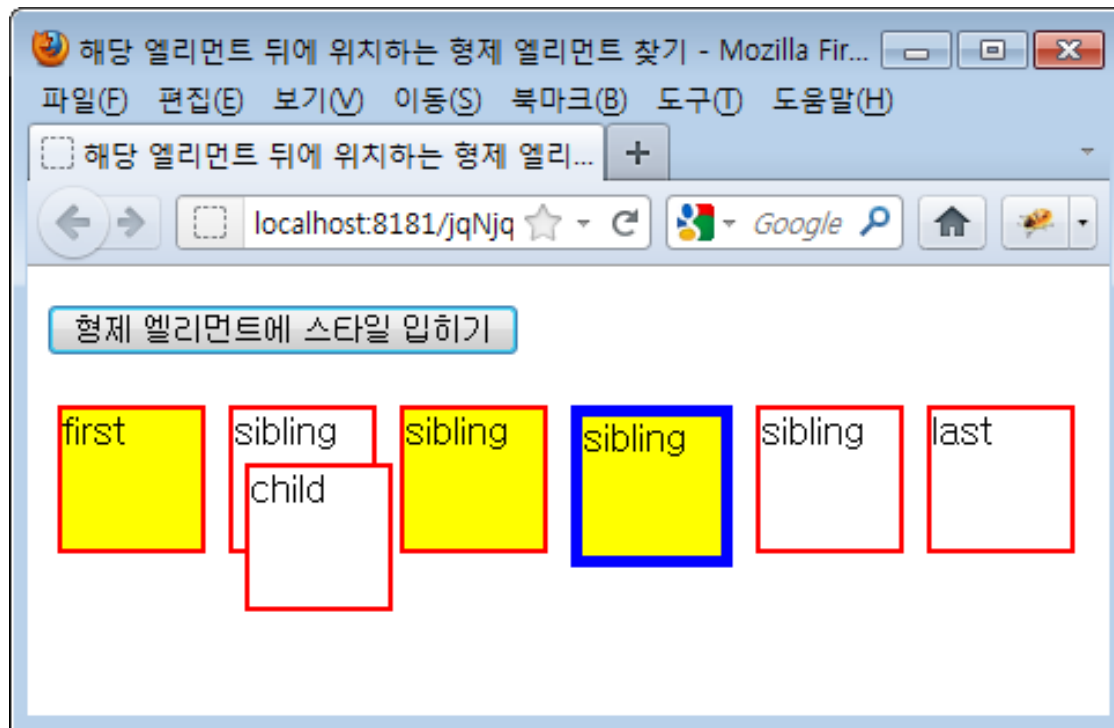
<div id="sixth">last</div>

</body>

</html>

해당 엘리먼트 앞에 있는 형제 엘리먼트 찾기

```
$("#p").prev().css("background", "yellow");  
$("#p").prev(".selected").css("background", "yellow");
```



해당 엘리먼트 앞에 있는 형제 엘리먼트 찾기

- **prev()**

- 대상이 되는 엘리먼트를 기준으로 동일한 노드의 현재 위치 이전의 인접한 엘리먼트를 찾는다.

- **\$("p").prev()**

- p 엘리먼트와 같은 노드에 존재하며 바로 이전에 존재하는 엘리먼트를 찾는다.
\$("p").prev(".selected")는 p 엘리먼트 이전에 존재하면서 class 속성값이 selected 인 엘리먼트를 찾는다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>해당 엘리먼트 뒤에 위치하는 형제 엘리먼트 찾기</title>
<style>
div, span {
    width: 60px; height: 60px;
    margin: 5px; float: left;
    border: 2px red solid;
    background: white;
}
.before { background: yellow}
.selected {border: 5px blue solid;}
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $('button').click(function(){
        $("span").prev().addClass("before");
        $("span").prev("#fourth").addClass("selected");
    });
});
</script>
</head>
```



```
<body>
```

```
  <p> <button> 형제 엘리먼트에 스타일 입히기 </button> </p>
```

```
  <div id="first">first</div>
```

```
  <span id="second" class="middle">
```

```
    sibling
```

```
    <div id="child">child</div>
```

```
  </span>
```

```
  <div id="third" class="middle">sibling</div>
```

```
  <span id="fourth" class="middle">sibling</span>
```

```
  <span id="fifth" class="middle">sibling</span>
```

```
  <div id="sixth">last</div>
```

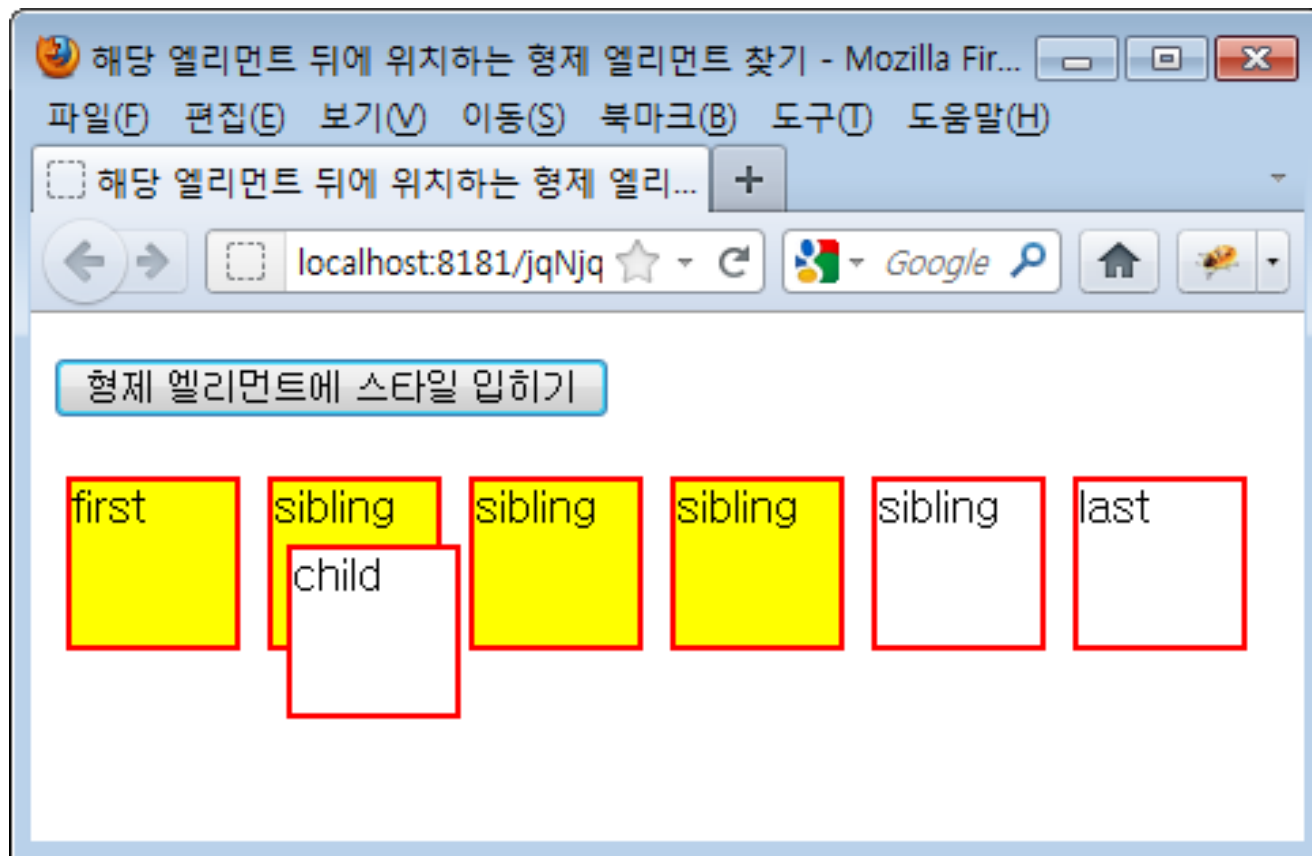
```
</body>
```

```
</html>
```



해당 엘리먼트 앞에 있는 모든 형제 엘리먼트 찾기

```
$("#div:last").prevAll('span, div').addClass("before");
```



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>해당 엘리먼트 뒤에 위치하는 형제 엘리먼트 찾기</title>
```

```
<style>
```

```
div, span {
```

```
    width: 60px; height: 60px;
```

```
    margin: 5px; float: left;
```

```
    border: 2px red solid;
```

```
    background: white;
```

```
}
```

```
.before { background: yellow}
```

```
.selected {border: 5px blue solid;}
```

```
</style>
```

```
<script src="../js/jquery.js" type="text/javascript"></script>
```

```
<script type="text/javascript">
```

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

```
        $('button').click(function(){
```

```
            $("span").prevAll("span, div").addClass("before");
```

```
        });
```

```
    });
```

```
</script>
```

```
</head>
```

<body>

<p> <button> 형제 엘리먼트에 스타일 입히기 </button> </p>

<div id="first">first</div>

sibling

<div id="child">child</div>

<div id="third" class="middle">sibling</div>

sibling

sibling

<div id="sixth">last</div>

</body>

</html>

해당 엘리먼트와 같은 노드 위치의 모든 엘리먼트 찾기

siblings()

```
var len = $(".hilite").siblings().css("color", "red").length;
```

```
<ul>
```

```
  <li>One</li>
```

```
  <li>Two</li>
```

```
  <li class="hilite">Three</li>
```

```
  <li>Four</li>
```

```
</ul>
```

```
<ul>
```

```
  <li>Five</li>
```

```
  <li>Six</li>
```

```
  <li>Seven</li>
```

```
</ul>
```

```
<ul>
```

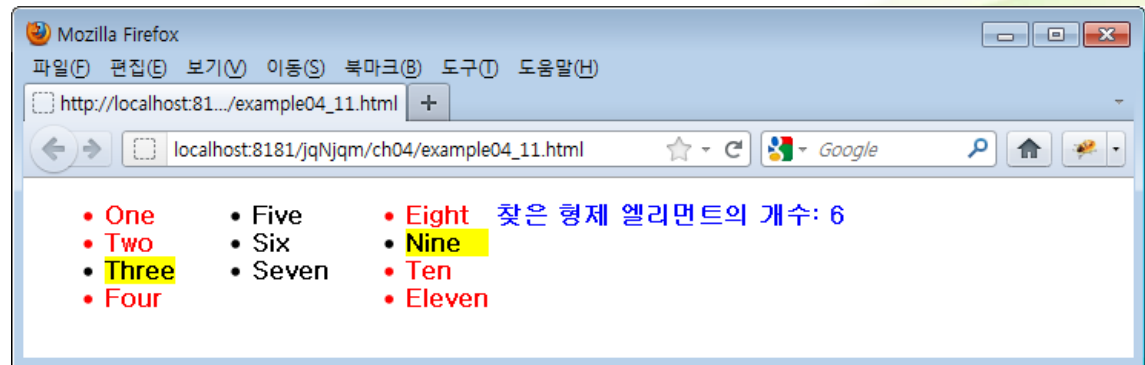
```
  <li>Eight</li>
```

```
  <li class="hilite">Nine</li>
```

```
  <li>Ten</li>
```

```
  <li>Eleven</li>
```

```
</ul>
```

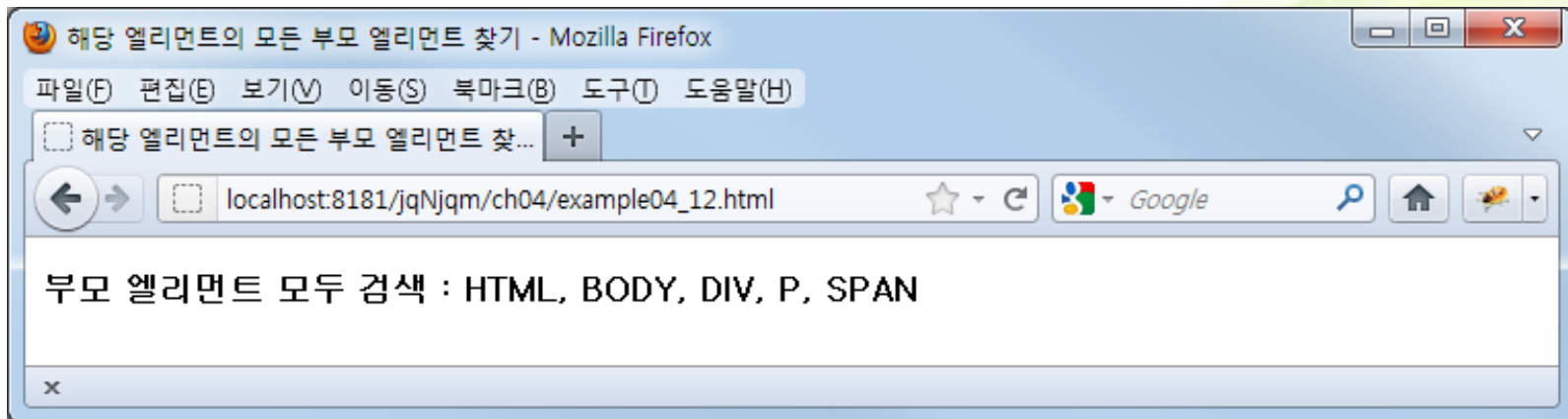


```
<!DOCTYPE html><html><head>
<meta charset="UTF-8">
<title></title>
<style>
  ul { float:left; margin:5px; font-size:16px; font-weight:bold; }
  p { color:blue; margin:10px 20px; font-size:16px; padding:5px;
    font-weight:bolder; }
  .yellow { background:yellow; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
  $(function() {
    var len = $(".yellow").siblings()
      .css("color", "red")
      .length;
    $("b").text(len);
  });
</script></head>
<body>
  <ul> <li>One</li>   <li>Two</li>   <li class="yellow">Three</li>   <li>Four</li> </ul>
  <ul>   <li>Five</li>   <li>Six</li>   <li>Seven</li> </ul>
  <ul> <li>Eight</li>   <li class="yellow">Nine</li>   <li>Ten</li>   <li>Eleven</li> </ul>
  <p>   찾은 형제 엘리먼트의 개수: <b></b> </p>
</body>
</html>
```

해당 엘리먼트의 모든 부모 엘리먼트 찾기

parents()

```
var result=new Array();
$("b").parents().each(function(i) {
    result[i]=this.tagName;
});
$("b").append(result.reverse().join(", "));
```



```
<!DOCTYPE html><html><head>
<meta charset="UTF-8">
<title>해당 엘리먼트의 모든 부모 엘리먼트 찾기</title>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    var result=new Array();
    $("b").parents()
        .each(function(i) {
            result[i]=this.tagName;
        });

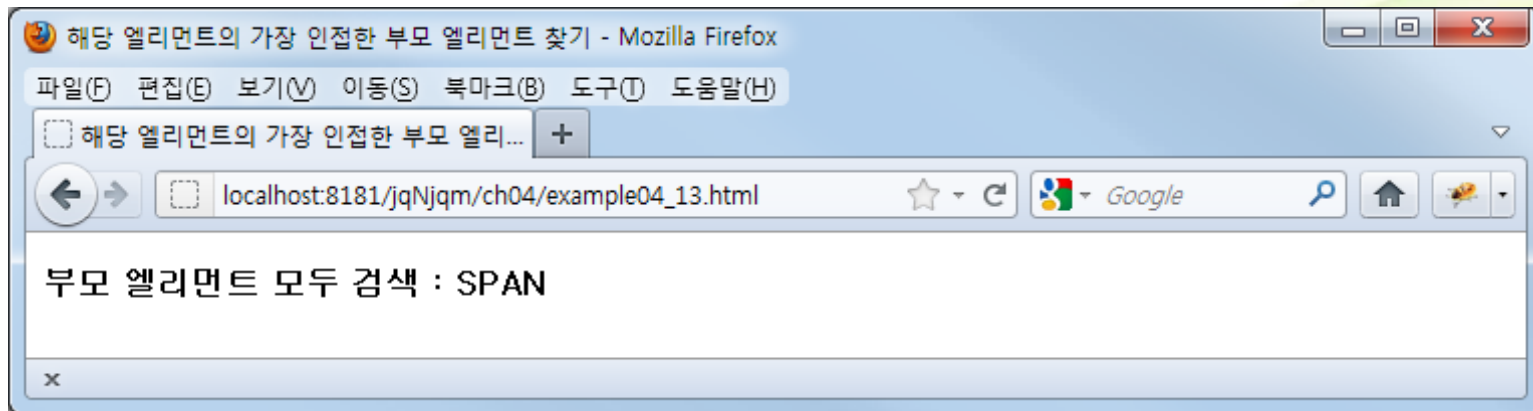
    $("b").append(result.reverse().join(", "));
});
</script>
</head>
<body>
<div>
    <p>    <span> <b> 부모 엘리먼트 모두 검색 : </b> </span>    </p>
</div>
</body>
</html>
```

해당 엘리먼트의 가장 인접한 부모 엘리먼트 찾기

parent()

```
$("#p").parent( ).css("background", "yellow");
```

```
$("#p").parent(".selected").css("background", "yellow");
```



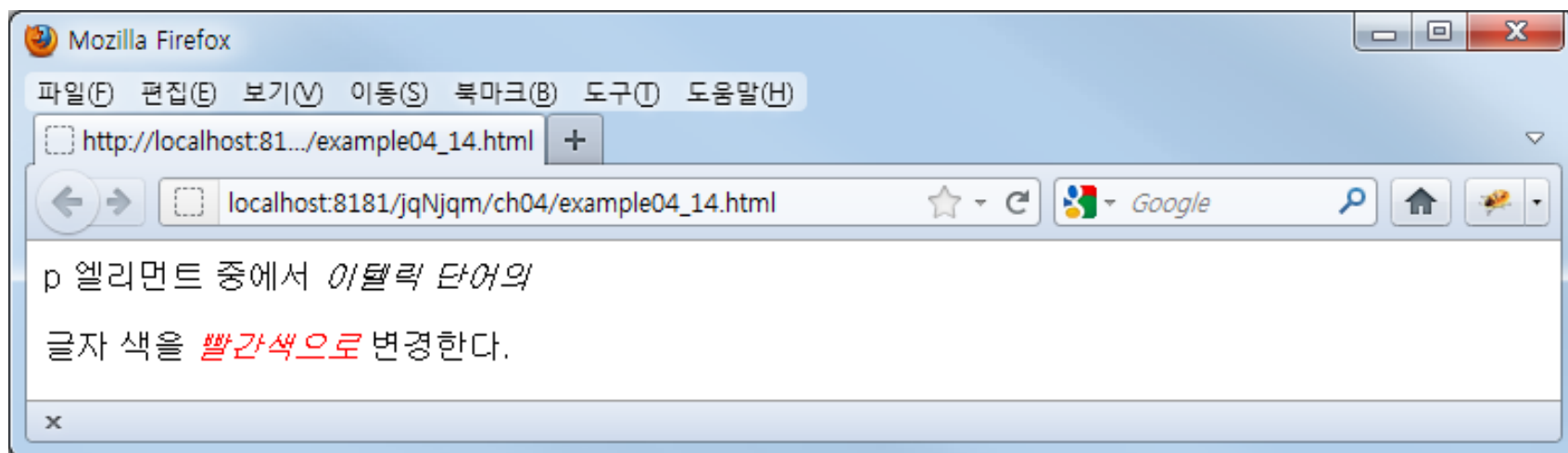
```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>해당 엘리먼트의 가장 인접한 부모 엘리먼트 찾기</title>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        var result=new Array();
        $("b").parent()
            .each(function(i) {
                result[i]=this.tagName;
            });

        $("b").append(result.reverse().join(", "));
    });
</script>
</head>
<body>
    <div>
        <p>
            <span> <b> 부모 엘리먼트 모두 검색 : </b> </span>
        </p>
    </div>
</body>
</html>
```

선택된 엘리먼트들 중에서 표현식을 통해 원하는 엘리먼트 얻기

find()

```
$("#p").find("span").css('color','red');
```



선택된 엘리먼트들 중에서 표현식을 통해 원하는 엘리먼트 얻기

find()

find(selector 혹은 elements 혹은 jQuery object)	
설명	현재 선택된 객체 집합의 컨텍스트와 그 자손들을 기반으로 새로운 객체 집합을 구성한다.
매개변수	selector : 엘리먼트를 선택할 셀렉터 표현식 elements : 자손 엘리먼트를 검색할 엘리먼트 jQuery object : 자손 엘리먼트를 검색할 jQuery 객체
반환값	jQuery 확장 객체 집합

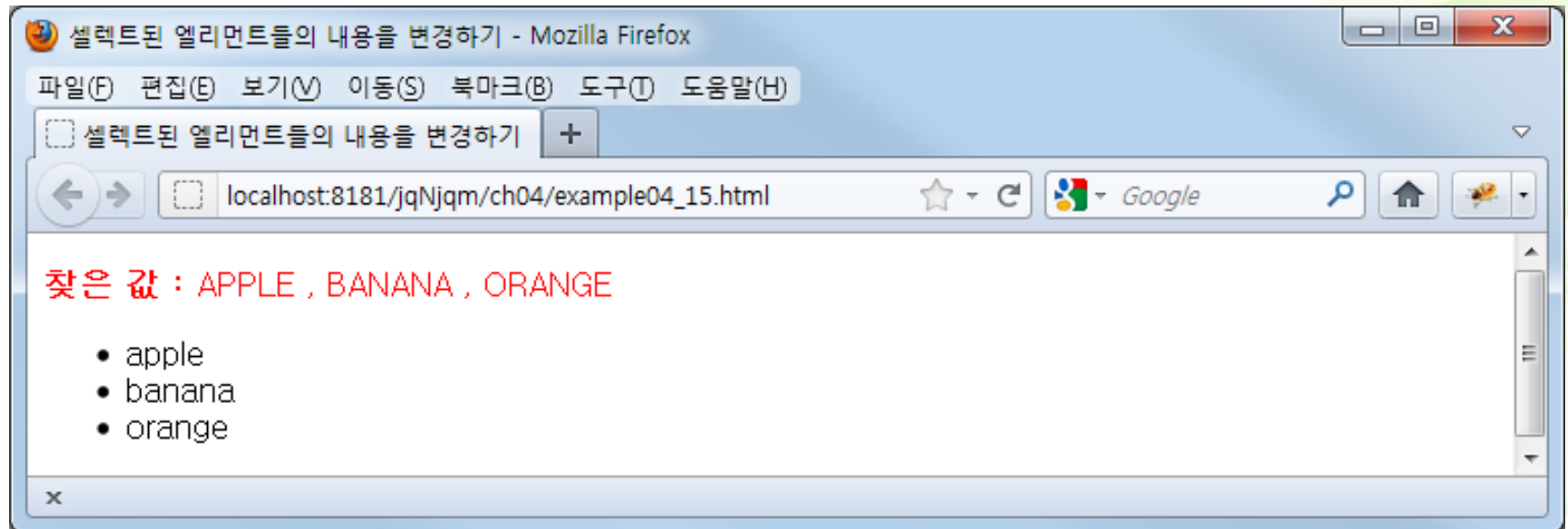

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $("p").find("em").css('color','red');
        //$("em", "p").css('color','red');
        //$("p em").css('color','red');
    });
</script>
</head>
<body>
    <div> p 엘리먼트 중에서 <em> 이텔릭 단어의 </em> </div>
    <p> 글자 색을 <em> 빨간색으로 </em> 변경한다.</p>
</body>
</html>
```

선택된 엘리먼트들의 내용을 변경하기

```
$("#p").append($("#li").map(function() {  
    return $(this).text().toUpperCase();  
}).get().join(", "));
```

■ map(fn)

- 리스트(배열)과 같은 성격을 가진 엘리먼트의 변경에 용이하다.



```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>선택된 엘리먼트들의 내용을 변경하기</title>
<style>
p {
  color: red;
}
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
  $(function() {
    alert("");
    $("p").append($("li").map(function() {
      return $(this).text().toUpperCase();
    }).get().join(", "));
  });
</script></head>
<body>
  <p>
    <b> 찾은 값 : </b>
  </p>
  <ul>
    <li> apple </li>    <li> banana </li>    <li> orange </li>
  </ul>
</body>
</html>
```

탐색한 엘리먼트들에 기준이 되었던 엘리먼트들 추가하기

andSelf()는 jQuery3.0에서 addBack()으로 변경됨

```
$("#div").find("p").addClass("background")  
    .andSelf().addClass("border");
```

■ andSelf()

- 최초 대상이 되었던 엘리먼트를 포함해서 되돌린다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title></title>
<style>
  p, div { margin:5px; padding:5px; }
  .border { border: 2px solid red; }
  .background { background:yellow; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
  $(function() {
    $("div").find("p")
      .addClass("background")
      .andSelf()
      .addClass("border");
  });
</script></head>
<body>
  <div>
    <p>첫째 문장</p>
    <p>둘째 문장</p>
  </div>
</body>
</html>
```

메소드 체이닝(CHANING) 중에 최초 엘리먼트로 돌아가기

end()

```
$("#div").find("p").addClass("background")  
    .end().addClass("border");
```

■ end()

- jQuery는 체인을 통해 확장집합을 생성할 경우 가장 상위의 확장집합으로 되돌리는 역할

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>메소드 체이닝(chaining) 중에 최초 엘리먼트로 돌아가기</title>
<style>
  p, div { margin:5px; padding:5px; }
  .border { border: 2px solid red; }
  .background { background:yellow; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
  $(function() {
    $("div").find("p")
      .addClass("background")
      .end()
      .addClass("border");
  });
</script></head>
<body>
  <div>
    <p>첫째 문장</p>
    <p>둘째 문장</p>
  </div>
</body>
</html>
```

해당 엘리먼트에 자식 엘리먼트로 추가하기

append(content [, content])

```
$("#span").append("<br> <strong> 사과 </strong> <br>");
```


내가 좋아하는 과일

내가 좋아하는 과일

 사과

■ append()

- 인자로 전달된 내용을 대상 엘리먼트 뒤에 마지막 자식 엘리먼트로 추가한다.

해당 엘리먼트에 자식 엘리먼트로 추가하기

appendTo('셀렉터')

```
$("<br> <strong> 사과 </strong> <br>").appendTo("span");
```


내가 좋아하는 과일

내가 좋아하는 과일

 사과

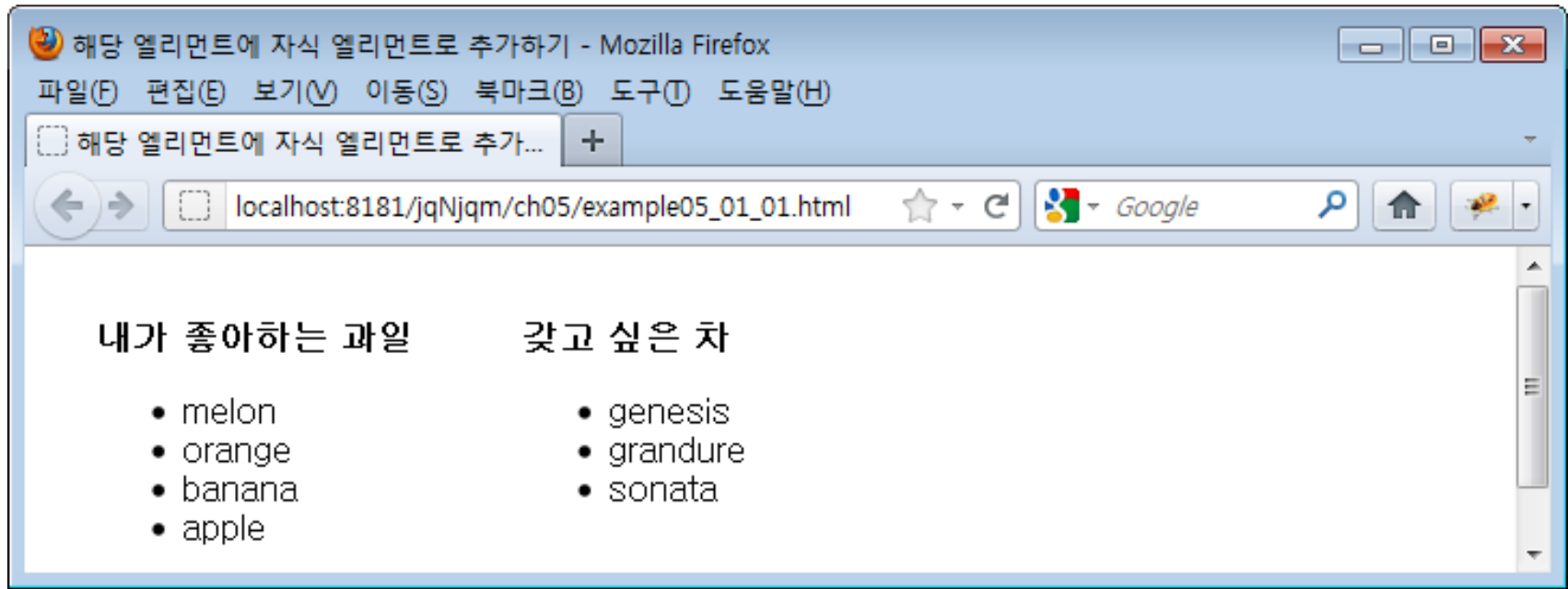
■ appendTo()

- append()와 같은 동작을 수행한다. 차이점은 대상과 콘텐츠를 배치하는 순서에 있다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>해당 엘리먼트에 자식 엘리먼트로 추가하기</title>
<style type="text/css">
div { float:left; margin:25px; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        var fruits=["apple", "banana", "orange", "melon"];
        var cars=["sonata", "grandure", "genesis"];
        $.each(fruits, function(index, value){
            $("#fruit_list").append($("<li>" + value + "</li>"));
        });
        $.each(cars, function(index, value){
            $("<li>" + value + "</li>").appendTo("#car_list");
        });
    });
</script></head>
<body>
<div>
    <b> 내가 좋아하는 과일 </b>    <ul id="fruit_list"></ul>
</div>
<div>
    <b> 갖고 싶은 차 </b>    <ul id="car_list"></ul>
</div>
</body></html>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>해당 엘리먼트에 자식 엘리먼트로 추가하기</title>
<style type="text/css">
span { display:block; }
</style>
<script src="../../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $('span').append($('b'));
    });
</script>
</head>
<body>
    <b> 사과 </b>
    <span> 갖고 싶은 과일 : </span>
    <span> 먹고 싶은 과일 : </span>
</body>
</html>
```

해당 엘리먼트에 자식 엘리먼트로 추가하기



`$(target).prepend(새로운엘리먼트)==$(새로운엘리먼트).prependTo(target)`

해당 엘리먼트에 자식 엘리먼트로 추가하기

prepend(content [, content])

```
$("span").prepend("<br> <strong> 사과 </strong> <br>");
```


내가 좋아하는 과일

 사과

내가 좋아하는 과일

■ prepend()

- 인자로 전달된 내용을 대상 엘리먼트 앞에 첫 번째 자식 엘리먼트로 추가한다.

해당 엘리먼트에 자식 엘리먼트로 추가하기

prependTo('셀렉터')

```
$("<br> <strong> 사과 </strong> <br>"). prependTo("span");
```


내가 좋아하는 과일

 사과

내가 좋아하는 과일

■ prependTo()

- prepend()와 같은 동작을 수행한다. 차이점은 대상과 콘텐츠를 배치하는 순서에 있다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
<title>해당 엘리먼트에 자식 엘리먼트로 추가하기</title>
<style type="text/css">
div { float:left; margin:25px; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        var fruits=["apple", "banana", "orange", "melon"];
        var cars=["sonata", "grandure", "genesis"];
        $.each(fruits, function(index, value){
            $("#fruit_list").prepend($("#li">"+value+"</li>"));
        });
        $.each(cars, function(index, value){
            $("#li">"+value+"</li>").prependTo("#car_list");
        });
    });
</script></head>
<body>
<div> <b> 내가 좋아하는 과일 </b> <ul id="fruit_list"></ul> </div>
<div> <b> 갖고 싶은 차 </b> <ul id="car_list"></ul> </div>
</body>
</html>
```

해당 엘리먼트의 뒤에 원하는 내용을 추가하기

after(), insertAfter()

```
$('#header').after($('<div> 좋아하는 과일 : 사과 </div>'));  
$('<div> 싫어하는 과일 : 바나나 </div>').insertAfter($('#header'));
```

■ insertAfter()

- 특정 노드 뒷부분에 요소를 삽입하기 위해 사용한다. 선택자를 사용하는 위치가 뒷부분이다.

■ after()

- 선택자를 사용하는 위치가 앞부분이다.


```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>해당 엘리먼트에 형제 엘리먼트로 추가하기</title>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $('header').after($('<div> 좋아하는 과일 : 사과 </div>'));
        $('<div> 싫어하는 과일 : 바나나 </div>').insertAfter($('header'));
    });
</script>
</head>
<body>
    <header> <h1> 과일 </h1></header>

    <footer> <h3> <i> copyright </i> </h3> </footer>
</body>
</html>
```

해당 엘리먼트의 앞에 원하는 내용을 추가하기

before(), insertBefore()

```
$('#footer').before($('<div> 좋아하는 과일 : 사과 </div>'));  
$('<div> 싫어하는 과일 : 바나나 </div>').insertBefore($('#footer'));
```

■ before()

- 특정 노드 앞부분에 요소를 삽입하기 위해 사용한다. 선택자를 사용하는 위치가 앞부분이다.

■ insertBefore()

- 선택자를 사용하는 위치가 뒷부분이다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>해당 엘리먼트에 형제 엘리먼트로 추가하기</title>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $('footer').before($('<div> 좋아하는 과일 : 사과 </div>'));
        $('<div> 싫어하는 과일 : 바나나 </div>').insertBefore($('footer'));
    });
</script>
</head>
<body>
    <header> <h1> 과일 </h1></header>

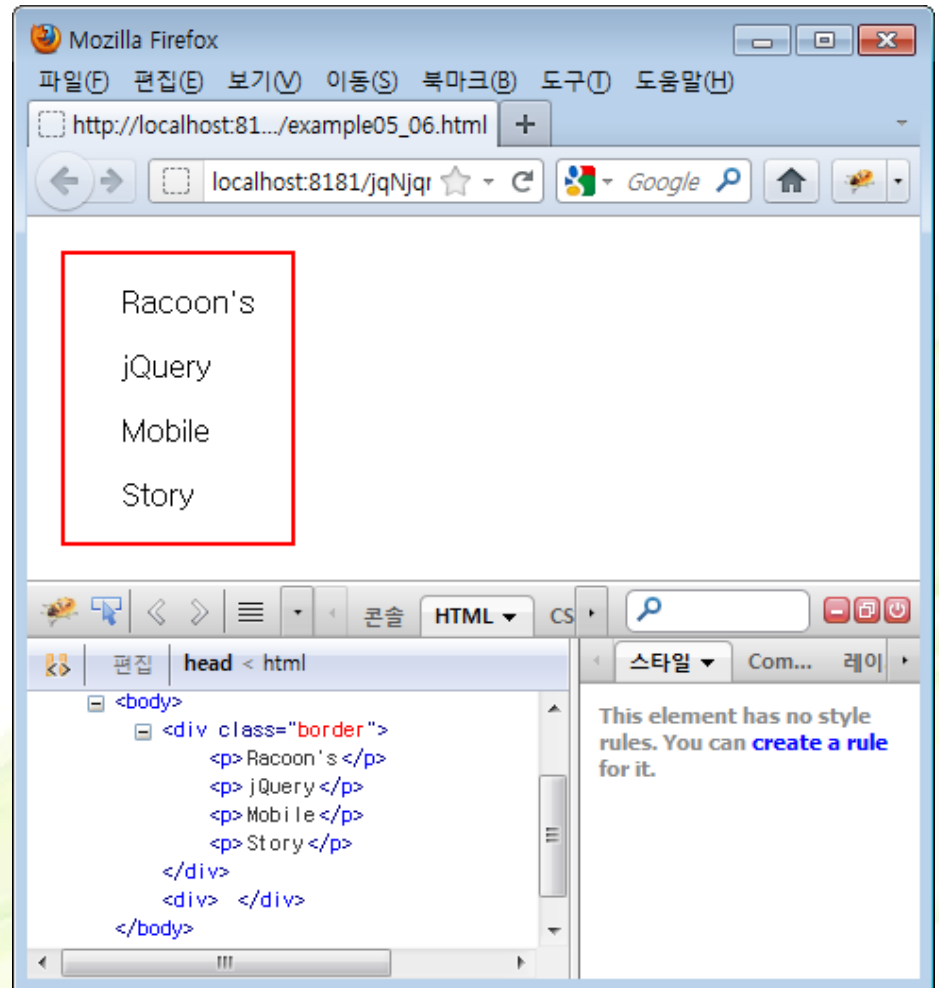
    <footer> <h3> <i> copyright </i> </h3> </footer>
</body>
</html>
```

해당 엘리먼트를 원하는 엘리먼트로 감싸기

```
$("#p").wrapAll("<div class='border'></div>");
```

■ wrapAll()

- 엘리먼트 집합의 모든 엘리먼트 주위를 감싼다.

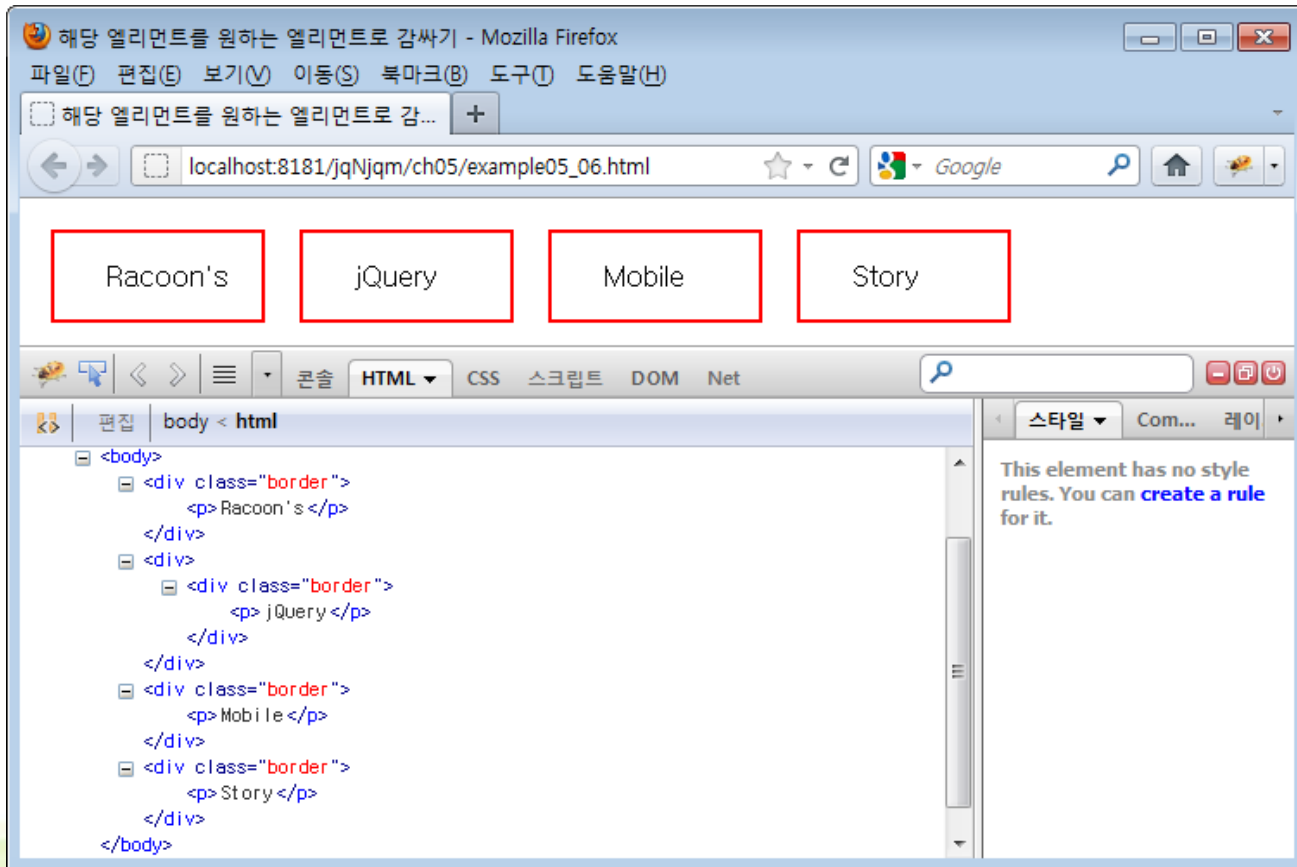


해당 엘리먼트를 원하는 엘리먼트로 감싸기

```
$("#p").wrap("<div class='border'></div>");
```

■ wrap()

- 엘리먼트 집합의 각 엘리먼트 주위를 개별적 감싼다.



```
<!DOCTYPE html><html><head>
<meta charset="UTF-8">
<title>해당 엘리먼트를 원하는 엘리먼트로 감싸기</title>
<style>
.border { border: 2px solid red;
          margin:10px;
          padding:0px 30px;
          float :left;
          width: 60px;
        }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $("p").wrap("<div class='border'></div>");
        //$("p").wrapAll("<div class='border'></div>");
    });
</script></head>
<body>
    <p>Racoon's</p>
    <div>    <p>jQuery</p>    </div>
    <p>Mobile</p>
    <p>Story</p>
</body>
</html>
```

해당 엘리먼트를 원하는 엘리먼트로 바꾸기

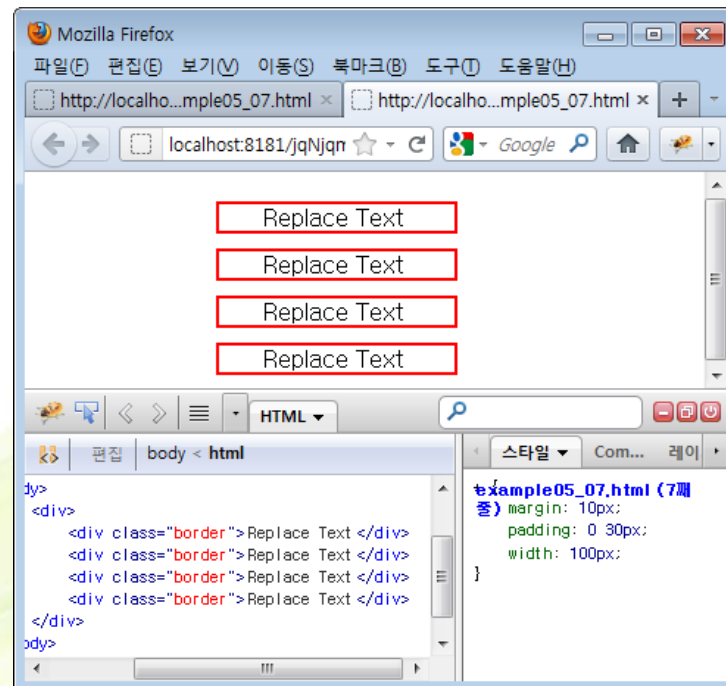
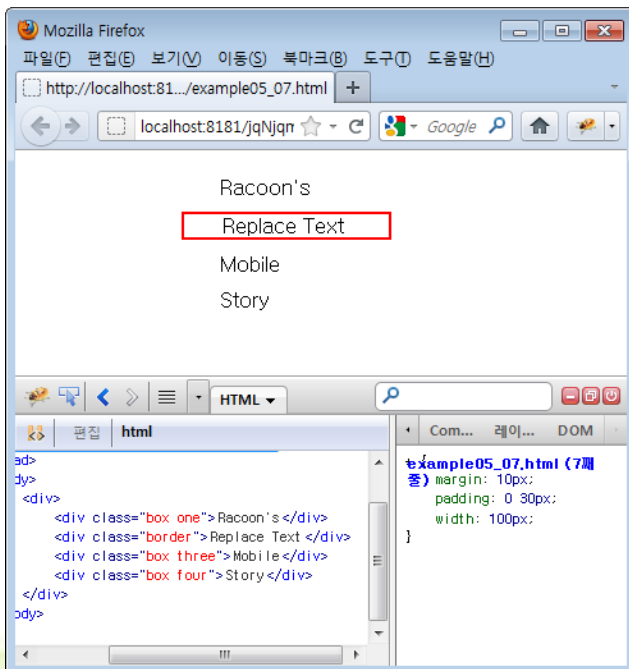
```
$('#div.two').replaceWith("<div class='border'>Replace Text</div>");  
$('#div.box').replaceWith("<div class='border'>Replace Text</div>");
```

■ replaceWith()

- 엘리먼트 집합에 속한 각 엘리먼트를 지정한 콘텐츠로 대체한다.

■ replaceAll()

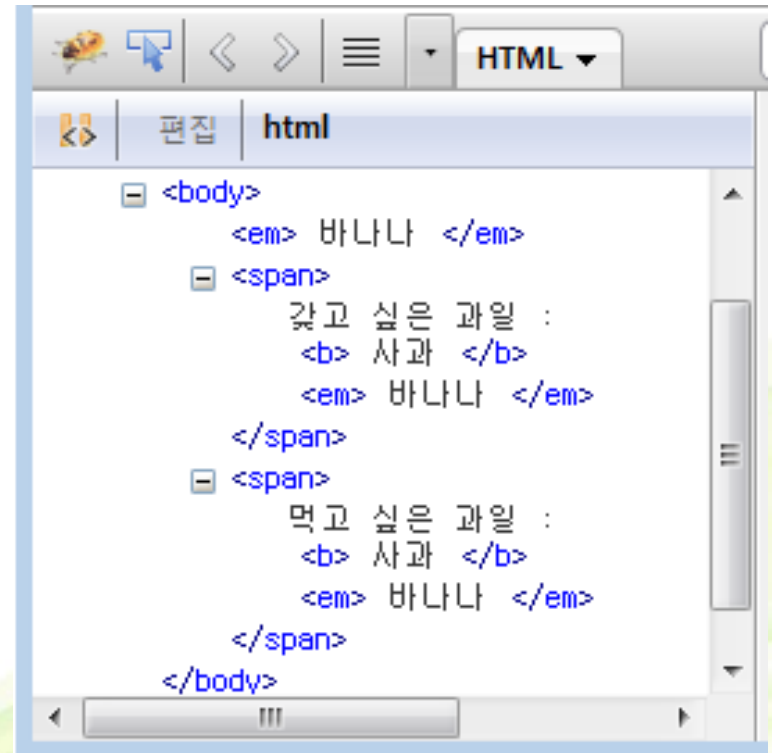
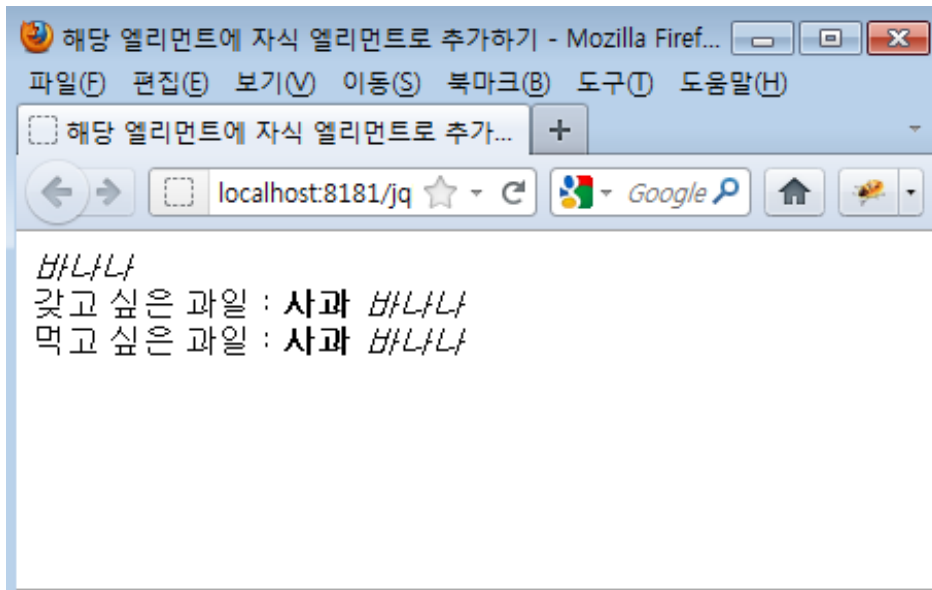
- 지정한 엘리먼트를 매개변수에 기술한 엘리먼트로 대체한다.



```
<!DOCTYPE html><html><head>
<meta charset="UTF-8">
<title>해당 엘리먼트를 원하는 엘리먼트로 바꾸기</title>
<style>
*{ margin:10px; padding:0px 30px; width: 100px; }
.border { border: 2px solid red; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(function() {
    $('div.two').replaceWith("<div class='border'>Replace Text</div>");
    // $('div.box').replaceWith("<div class='border'>Replace Text</div>");
    // $('div.three').replaceWith($('div.one'));
    // $('div.one').replaceAll($('div.three'));
});
</script></head>
<body>
<div>
    <div class="box one">Racoon's</div>
    <div class="box two">jQuery</div>
    <div class="box three">Mobile</div>
    <div class="box four">Story</div>
</div>
</body>
</html>
```


해당 엘리먼트를 복제하기

```
$('#em').clone().appendTo('span');
```



- **clone()**
 - 엘리먼트 집합을 복제한다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>해당 엘리먼트에 자식 엘리먼트로 추가하기</title>
<style type="text/css">
span { display:block; }
</style>
<script src="../js/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
    $(function() {
        $('b').appendTo('span');
        $('em').clone().appendTo('span');
    });
</script>
</head>
<body>
<b> 사과 </b>
<em> 바나나 </em>
<span> 갖고 싶은 과일 : </span>
<span> 먹고 싶은 과일 : </span>
</body>
</html>
```

this와 \$(this)

[this - Javascript]

Javascript의 경우에는 이벤트가 발생한 태그 요소가 표시됩니다!

[\$(this) - JQuery]

Jquery의 경우에는 이벤트가 발생한 요소의 정보들이 Object로 표시됩니다!

아래와 같이 코드를 통해 위에 설명을 확인해봤습니다 !!

```
<!DOCTYPE html><html><head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
<script>
```

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

```
    $("#div1").click(function(){
```

```
        console.log(this);    console.log("-----");
```

```
        console.log($(this));
```

```
        $("#test1").text($(this));
```

```
        $("#test2").text(this);
```

```
    });
```

```
});
```

```
</script> </head><body>
```

```
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-
color:yellow;"></div>
```

```
<div id="test1"></div> <div id="test2"></div>
```

```
</body></html>
```

this와 \$(this)

* 자바스크립트 this 를 제이쿼리를 통해 표현하고 싶다면 ???

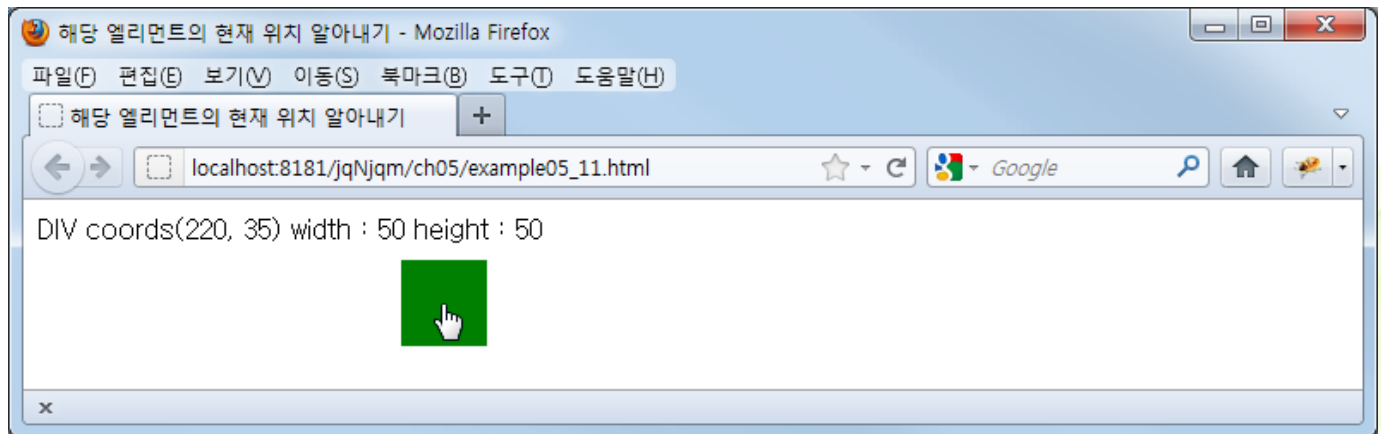
자바스크립트의 this는 jQuery의 \$(this)[0] 으로 동일 결과를 얻어낼 수 있습니다.

아래의 코드를 통해 확인할 수 있습니다

```
<!DOCTYPE html><html><head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#div1").click(function(){
        console.log(this)
        console.log("----- this -----")
        console.log("-----[this 는 $(this)[0]와 같다!!!]-----")
        console.log("----- $(this)[0] -----")
        console.log($(this)[0])
    });
});
</script></head><body>
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-
color:yellow;">
</body></html>
```

해당 엘리먼트의 현재 위치와 너비 높이를 알아내기

```
var offset=$(this).offset();
$("#result").text(this.tagName+
    " coords("+offset.left+ ", " +offset.top+")" +
    " width : "+ $(this).width()+" height : "+ $(this).height());
```



- **offset()**
 - 도큐먼트 내의 엘리먼트의 위치를 알려준다.
- **width()**
 - 엘리먼트의 너비를 알려준다.
- **height()**
 - 엘리먼트의 높이를 알려준다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
```

```
<title>해당 엘리먼트의 현재 위치 알아내기</title>
```

```
<style type="text/css">
```

```
div.abs{ width: 50px; height: 50px; position: absolute;
    left:220px; top:35px; background-color: green;
    cursor: pointer;
}
```

```
</style>
```

```
<script type="text/javascript" src="../js/jquery.js"></script>
```

```
<script type="text/javascript">
```

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

```
    $('div.abs').click( function(event){
```

```
        //대상 엘리먼트의 x좌표(left) y좌표(top)를 알려줌
```

```
        var offset=$(this).offset();
```

```
        $("#result").text(this.tagName+
```

```
            " coords("+offset.left+
```

```
                " , " +offset.top+" )" +
```

```
            " width : "+ $(this).width()+ //대상 엘리먼트의 가로길이
```

```
            " height : "+ $(this).height());//대상 엘리먼트의 세로길이
```

```
        }); //click(
```

```
    }); //ready(
```

```
</script></head>
```

```
<body>
```

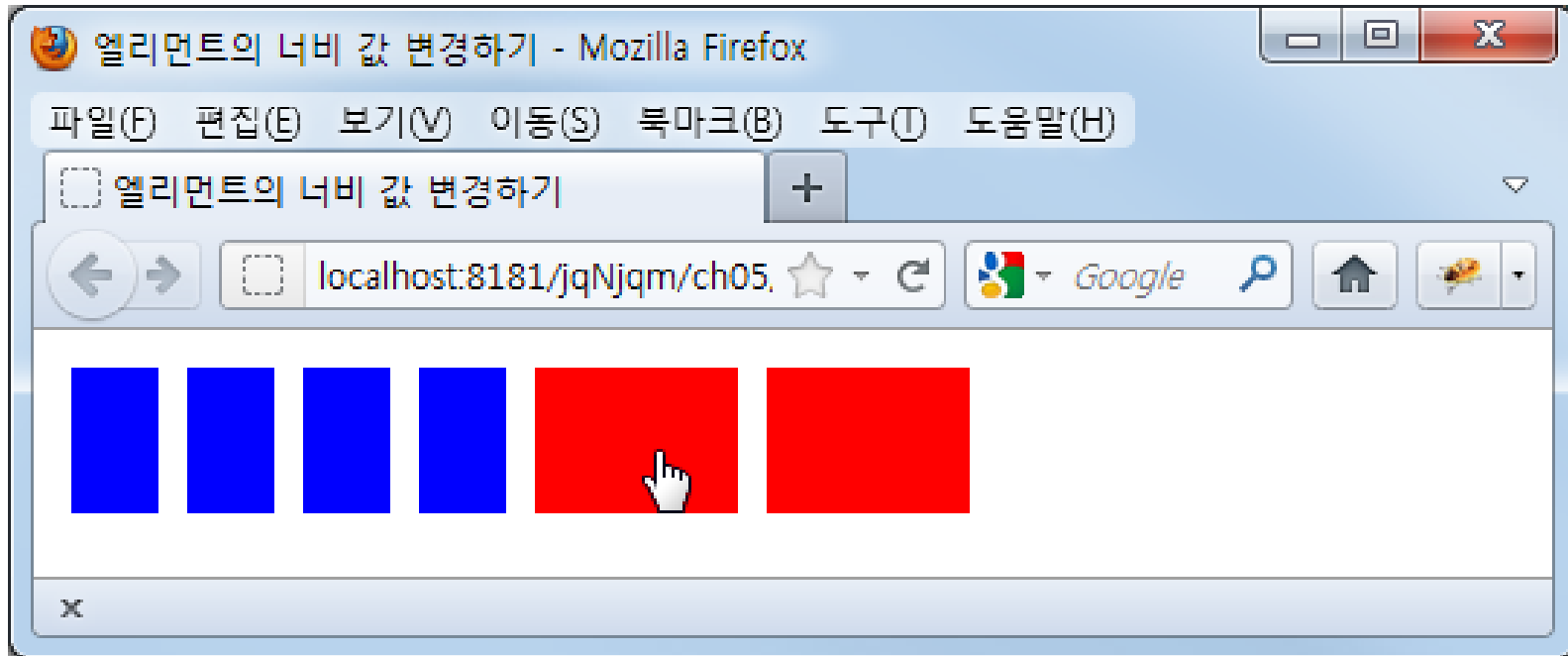
```
<div id="result"></div><div class="abs"></div>
```

```
</body>
```

```
</html>
```

엘리먼트의 너비 값 변경하기

```
$(this).width(30);
```



- **width(value)**
 - 엘리먼트의 너비를 변경한다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
```

```
<title>엘리먼트의 너비 값 변경하기</title>
```

```
<style type="text/css">
```

```
div{ width: 70px; height: 50px; float:left;  
    background-color: red; cursor: pointer;  
    margin: 5px
```

```
}
```

```
</style>
```

```
<script type="text/javascript" src="../js/jquery.js"></script>
```

```
<script type="text/javascript">
```

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

```
    //클릭 한번 처리하려면
```

```
    $('div').one('click', function(){
```

```
        // width() <- 인자 없이 사용되면 가로 길이를 알려줌
```

```
        $(this).width(30)
```

```
        .css( {cursor:"auto", backgroundColor:"blue"});
```

```
    });
```

```
});//ready(
```

```
</script></head>
```

```
<body>
```

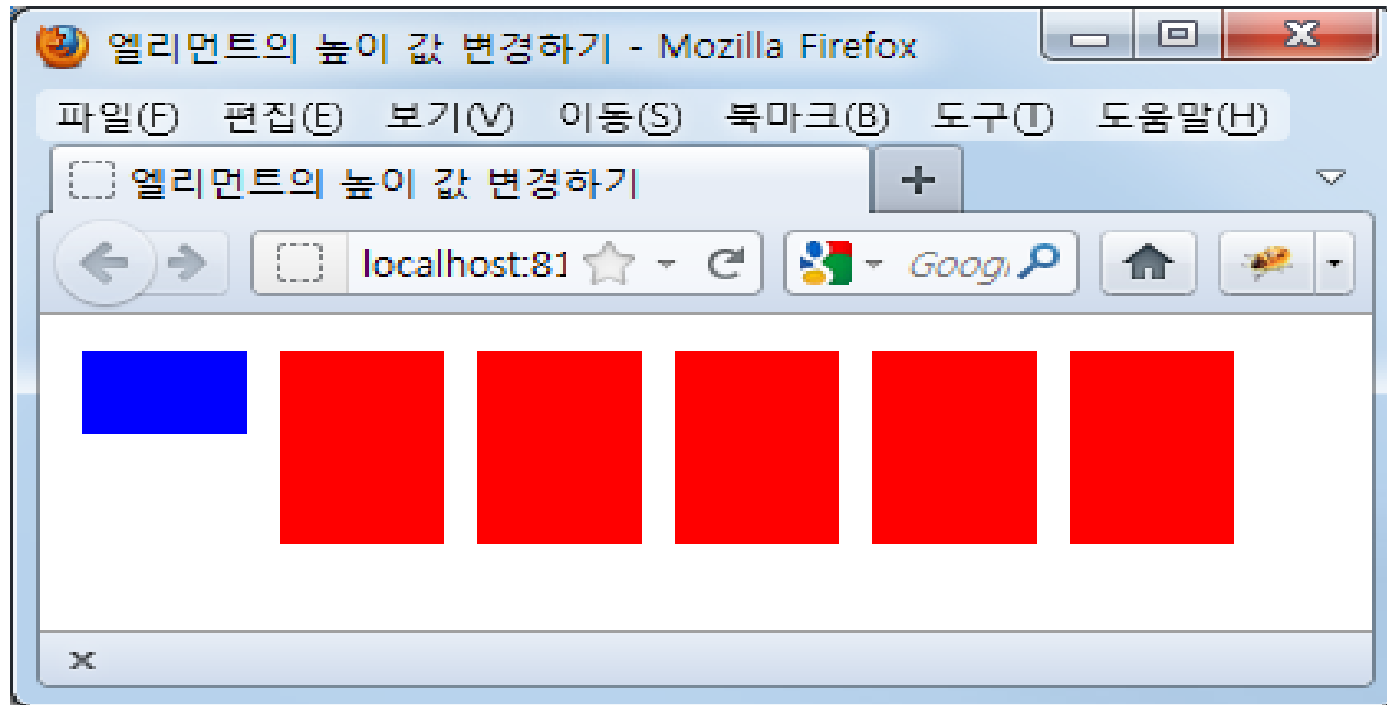
```
<div></div> <div></div> <div></div> <div></div><div></div> <div></div>
```

```
</body>
```

```
</html>
```


엘리먼트의 높이 값 변경하기

```
$(this).width(30);
```



- **width(value)**
 - 엘리먼트의 너비를 변경한다.

```
<!DOCTYPE html><html><head><meta charset="UTF-8">
```

```
<title>엘리먼트의 너비 값 변경하기</title>
```

```
<style type="text/css">
```

```
div{ width: 70px; height: 50px; float:left;
      background-color: red; cursor: pointer;
      margin: 5px
```

```
}
```

```
</style>
```

```
<script type="text/javascript" src="../js/jquery.js"></script>
```

```
<script type="text/javascript">
```

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

```
    //클릭 한번 처리하려면
```

```
    $('div').one('click', function(){
```

```
        // width() <- 인자 없이 사용되면 가로 길이를 알려줌
```

```
        $(this).height(30)
```

```
        .css( {cursor:"auto", backgroundColor:"blue"});
```

```
    });
```

```
});//ready(
```

```
</script></head>
```

```
<body>
```

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

```
</body>
```

```
</html>
```