jQuery

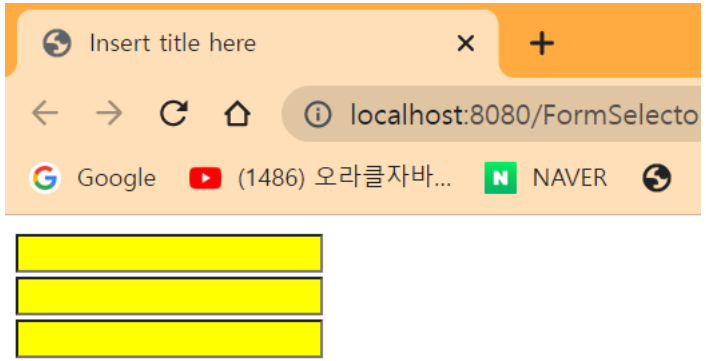
Event 처리 및 DOM 탐색

Form 태그 선택자

- Form 태그 내부의 입력에 관련된 태그들에 대한 선택자
- :input : 모든 입력에 관련된 태그들을 선택

:input

```
$(function(){
        $(":input").css("background-color", "yellow");
});
```

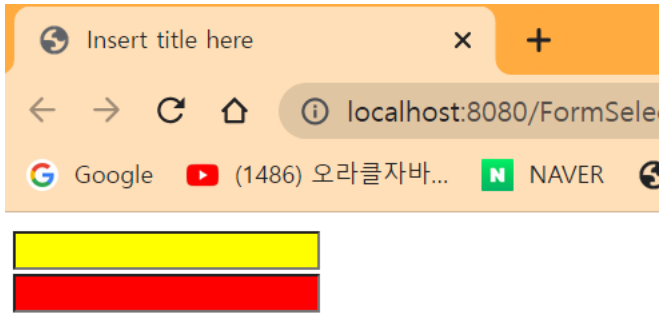


Form 태그 선택자

- :text : type 속성이 text인 input 태그를 선택
- :password : type 속성이 password인 input 태그를 선택
- :radio : type 속성이 radio인 input 태그를 선택
- :checkbox : type 속성이 submit인 input 태그를 선택
- :submit : type 속성이 submit인 input 태그를 선택
- :reset : type 속성이 rese 인 input 태그를 선택
- :button : type 속성이 button인 input 태그를 선택
- :image : type 속성이 image인 input 태그를 선택
- :file : type 속성이 file인 input 태그를 선택

:text, :password

```
$(function(){
    $(":text").css("background-color", "yellow");
    $(":password").css("background-color", "red");
});
```

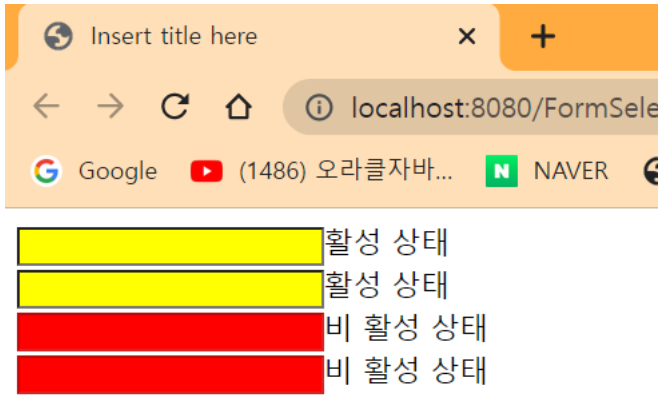


Form 태그 선택자

- :enabled : 활성 상태인 input 태그가 선택
- :disabled : 비 활성 상태인 input 태그가 선택
- :selected : select 태그 내의 option 태그 중 현재 선택되어 있는 태그를 선택
- :checked : checked나 radio에서 현재 check 되어 있는 태그를 선택

:enabled, :disabled

```
$(function(){
    $(":enabled").css("background-color", "yellow");
    $(":disabled").css("background-color", "red");
});
```



정리

- Form 태그 선택자를 활용하면 input 태그에 접근 할 수 있다.

jQuery 이벤트 함수

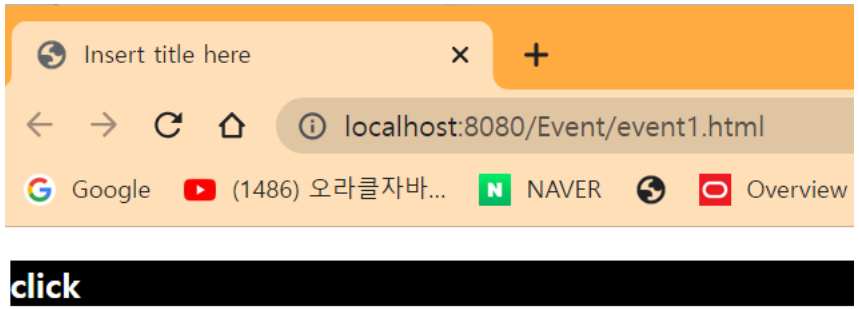
- jQuery는 여러 이벤트에 대해 이벤트 처리를 할 수 있는 함수를 제공
- 각 함수는 해당 이벤트가 발생됐을 때 등록된 함수를 자동으로 호출
- https://www.w3schools.com/jquery/jquery_ref_events.asp

jQuery 이벤트 함수

- click : 클릭
- Dblclick : 더블 클릭
- Mouseenter : 마우스 커서가 들어왔을 때
- Mouseleave : 마우스 커서가 나갔을 때
- Mousedown : 마우스 키를 눌렀을 때
- Mouseup : 마우스 키를 떼었을 때
- Hover : 마우스 커서가 들어왔을 때와 나갔을 때

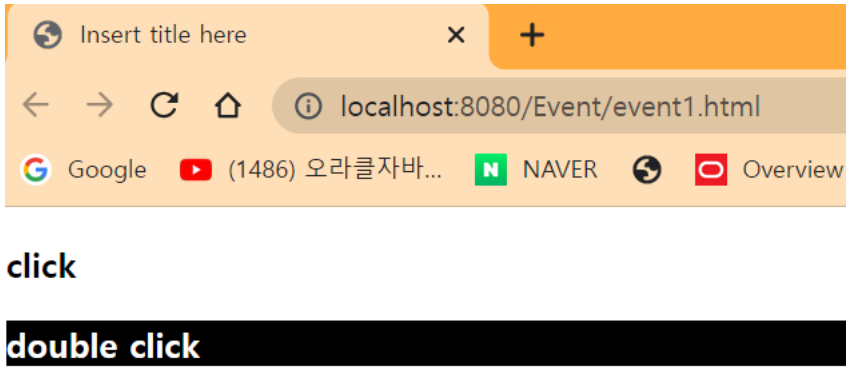
click

```
$(function(){
    $("#a1").click(function(){
        $("#a1").css("background-color", "black");
        $("#a1").css("color", "white");
    });
});
```



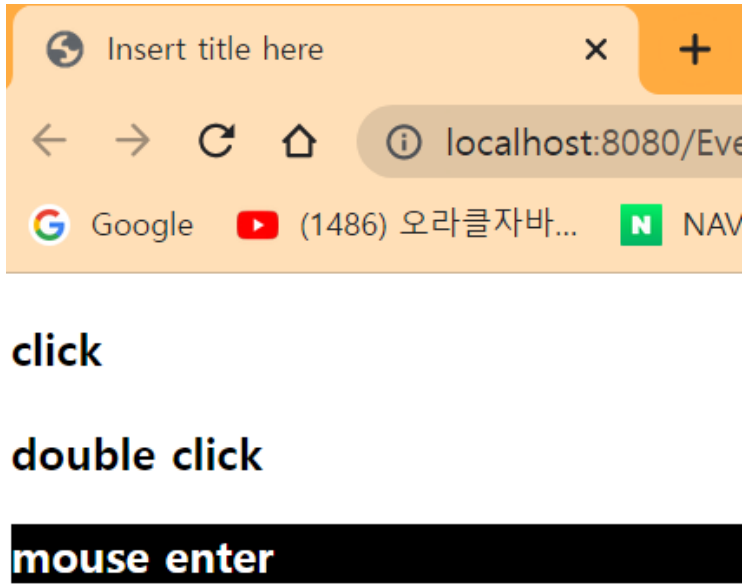
dblclick

```
$(function(){
    $("#a2").dblclick(function(){
        $("#a2").css("background-color", "black");
        $("#a2").css("color", "white");
    });
});
```



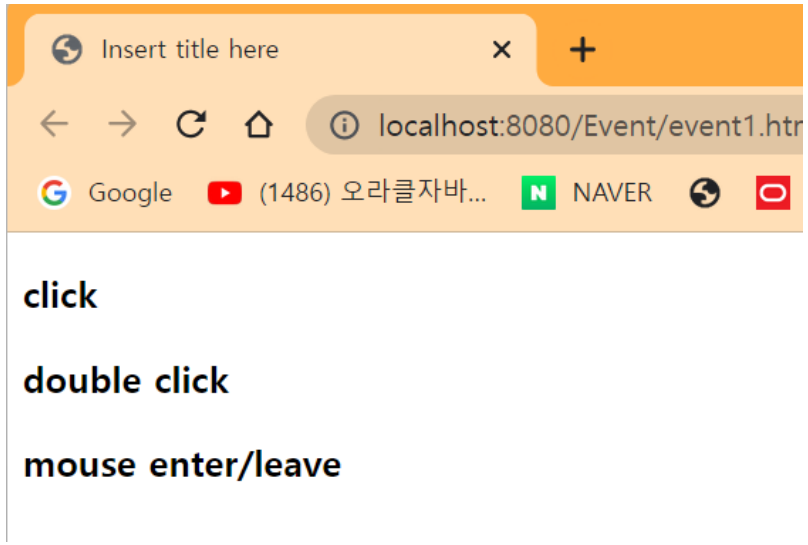
mouseenter

```
$(function(){
    $("#a3").mouseenter(function(){
        $("#a3").css("background-color", "black");
        $("#a3").css("color", "white");
    });
});
```



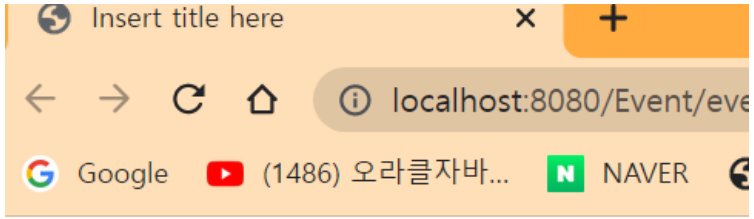
mouseleave

```
$(function(){
    $("#a3").mouseleave(function(){
        $("#a3").css("background-color", "white");
        $("#a3").css("color", "black");
    });
});
```



mousedown/up

```
$(function(){
    $("#a4").mousedown(function(){
        $("#a4").css("background-color", "black");
        $("#a4").css("color", "white");
    });

    $("#a4").mouseup(function(){
        $("#a4").css("background-color", "white");
        $("#a4").css("color", "black");
    });
});
```



click

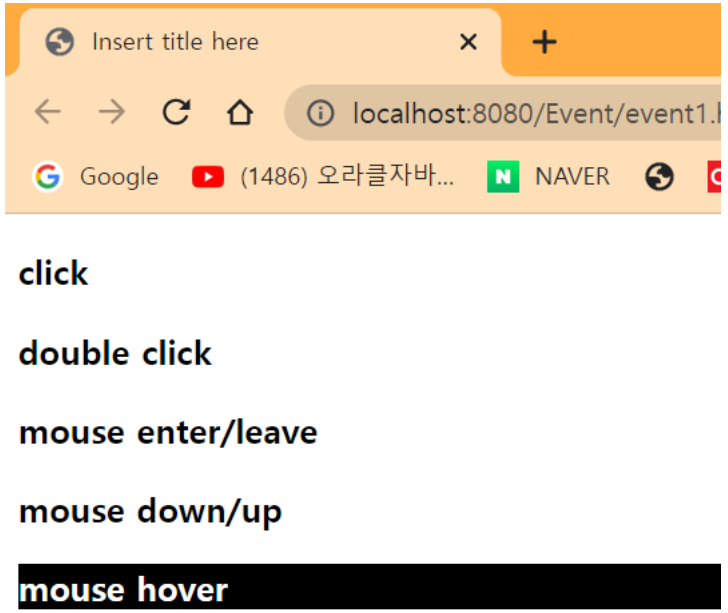
double click

mouse enter/leave

mouse down/up

hover

```
$(function(){
    $("#a5").hover(function(){
        $("#a5").css("background-color", "black");
        $("#a5").css("color", "white");
    }, function(){
        $("#a5").css("background-color", "white");
        $("#a5").css("color", "black");
    });
});
```

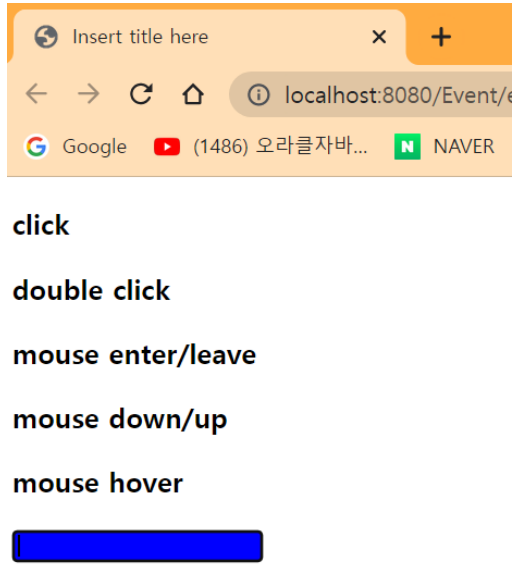


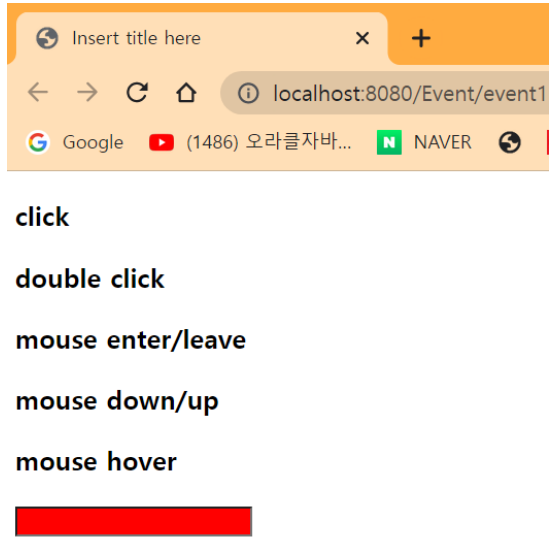
jQuery 이벤트 함수

- focus : 포커스가 주어졌을 때
- Blur : 포커스를 잃었을 때

focus/blur

```
$(function(){
    $("#a6").focus(function(){
        $("#a6").css("background-color", "blue");
    });

    $("#a6").blur(function(){
        $("#a6").css("background-color", "red");
    });
});
```



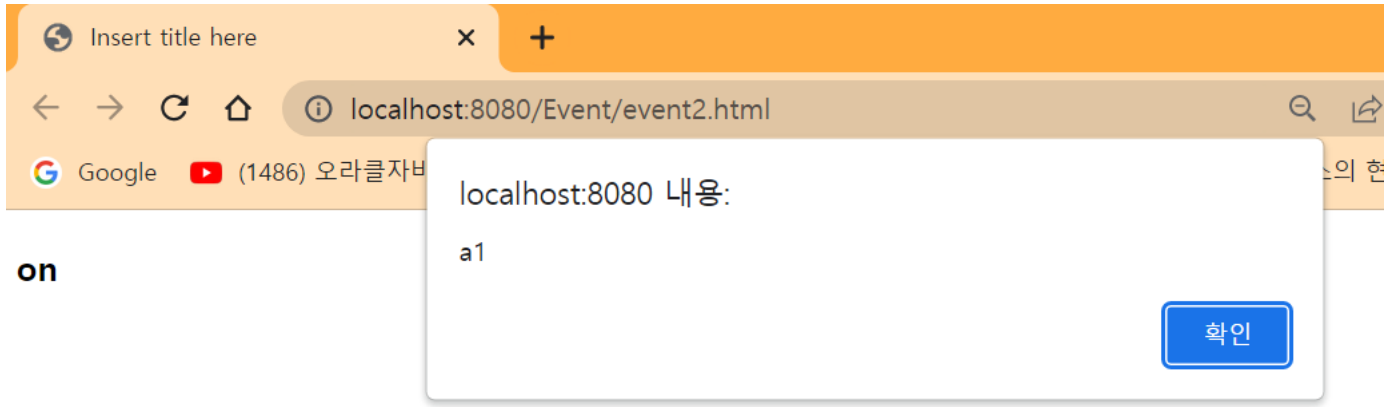


jQuery 이벤트 함수

- on : 이벤트를 설정하는 함수
- off : 설정된 이벤트를 제거하는 함수
- one : 이벤트를 설정하고 이벤트가 발생했을 때 자동으로 제거

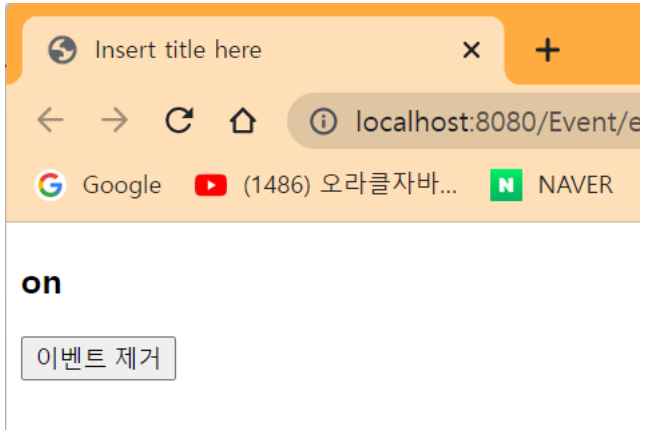
on

```
$(function(){
    $("#a1").on("click", function(){
        alert('a1');
    });
});
```



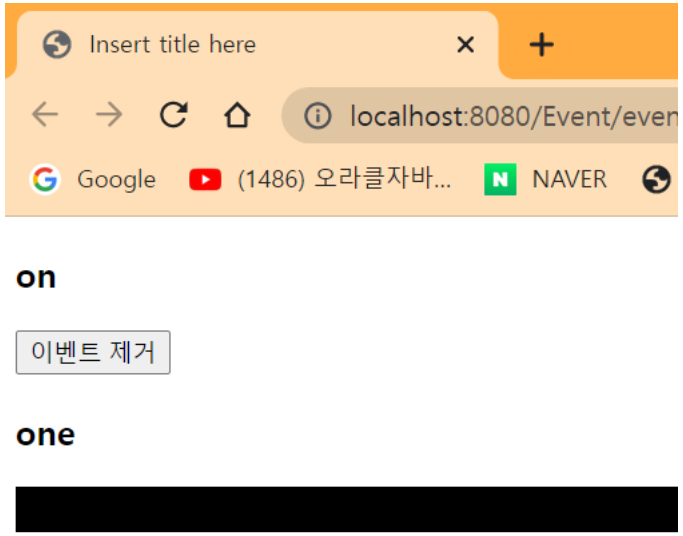
off

```
function remove_event(){
        $("#a1").off("click");
};
```



one

```
$(function(){
    $("#a2").one("click", function(){
        alert('a2');
    });

    $("#a3").on({
        click : function(){
        alert('click');
    },
        mouseenter : function(){
                $("#a3").css("background-color","black");
        },
        mouseleave : function(){
                $("#a3").css("background-color","white");
        }
    });
});
```



정리

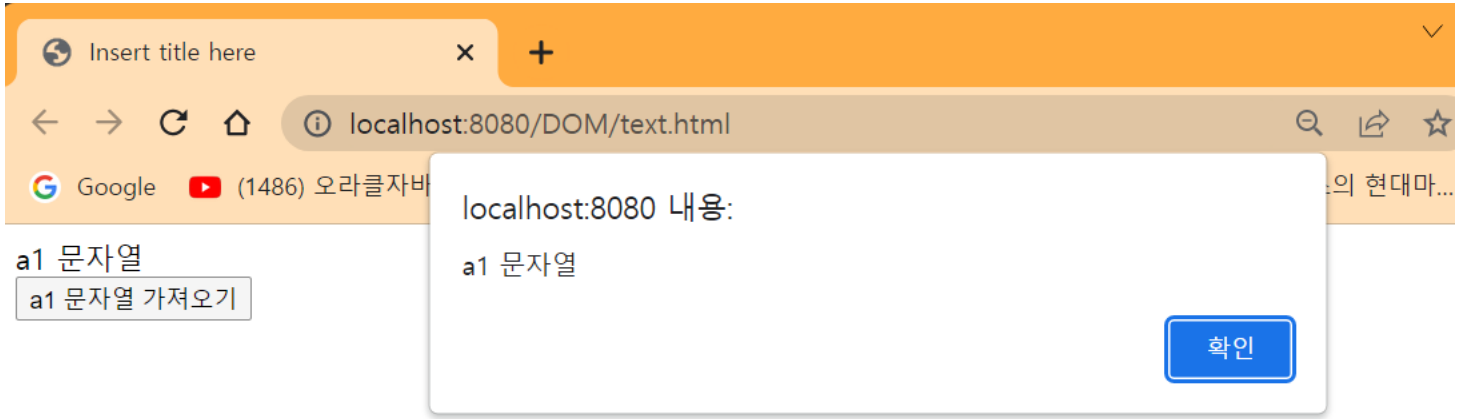
- jQuery는 다양한 이벤트를 처리할 수 있도록 함수를 제공한다.
- 이벤트의 이름과 동일한 함수를 이용해 이벤트를 처리할 수 있다.
- on을 사용하면 지정된 이벤트를 등록할 수 있다.
- off를 사용하면 지정된 이벤트를 제거할 수 있다.
- one을 사용하면 1회성 이벤트 드록이 가능하다.

DOM

- Document Object Model
- text : 태그 사이의 문자열을 제어
- html : 태그 내부의 html을 제어
- val : 입력 도구들의 value 속성값을 제어
- attr : 태그의 속성을 제어

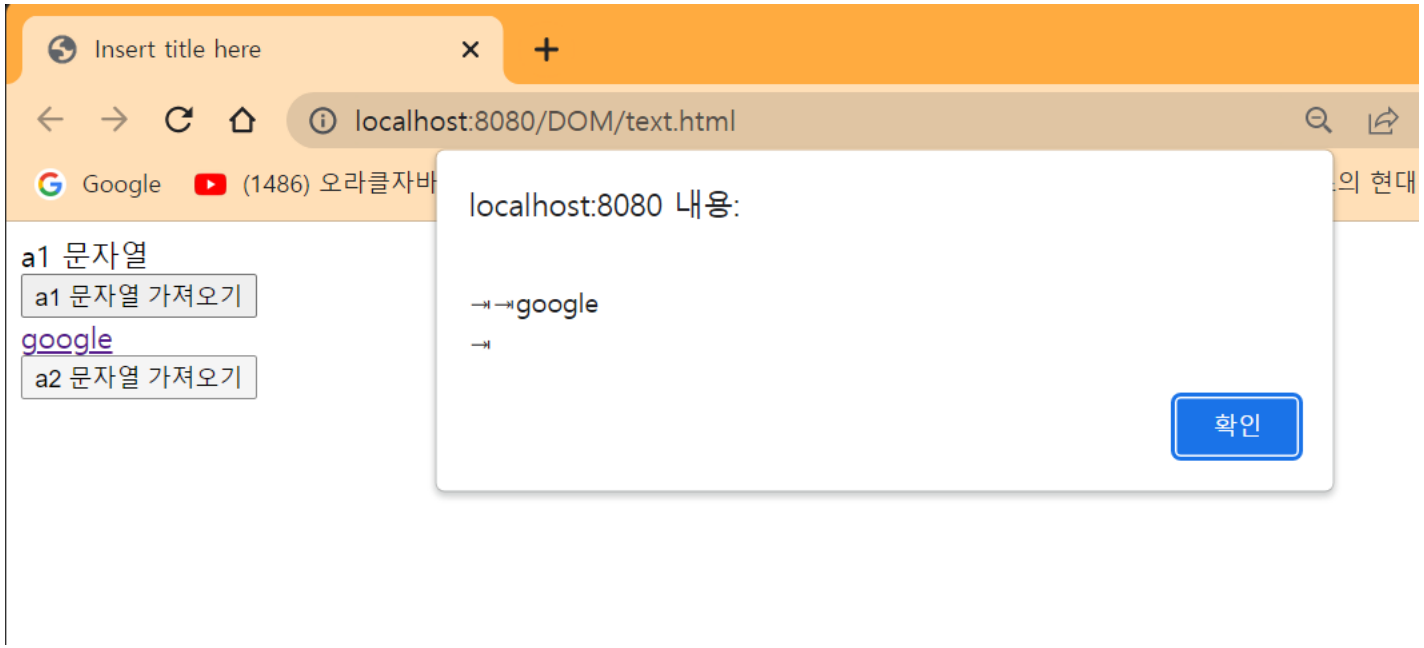
text

```
function getA1(){
    var str = $("#a1").text();
    alert(str);
};
```



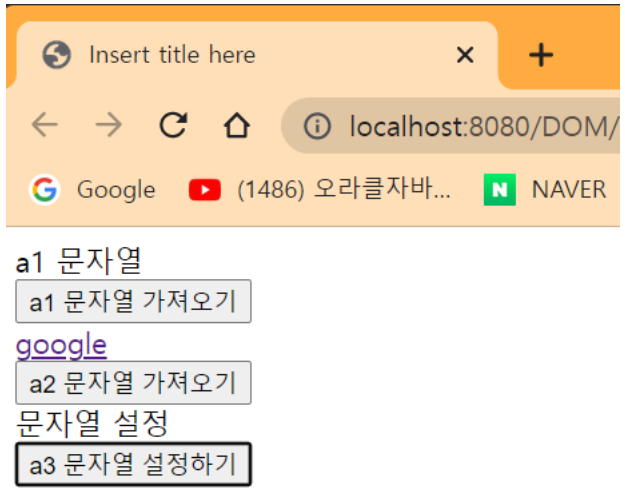
text

```
function getA2(){
    var str = $("#a2").text();
    alert(str);
};
```



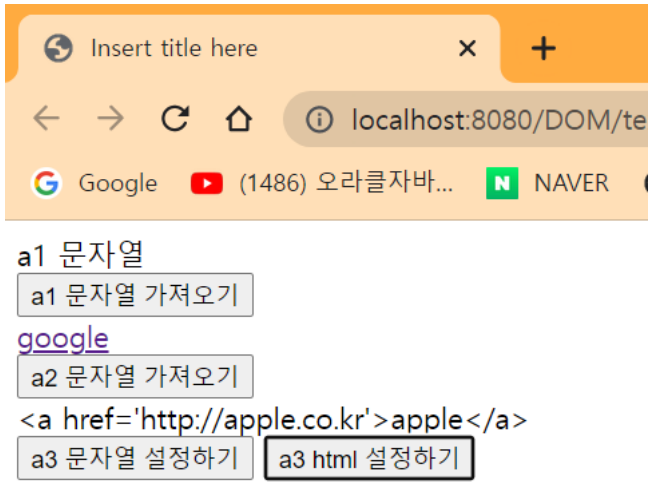
text

```
function setA3(){
    $("#a3").text("문자열 설정");
};
```



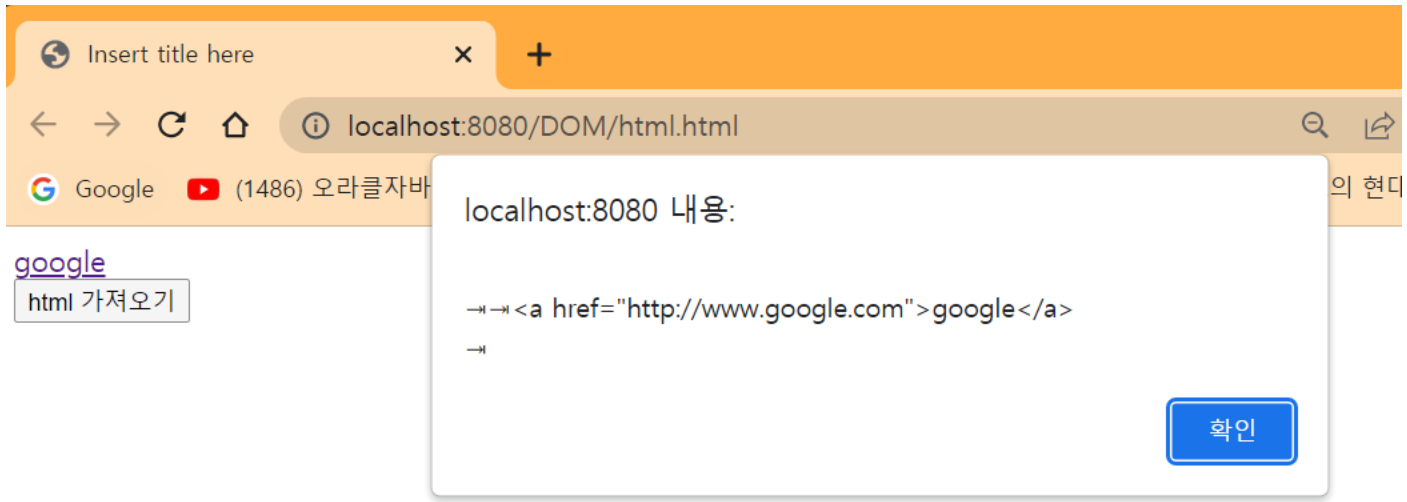
text

```
function setHtml(){
        $("#a3").text("<a href='http://apple.co.kr'>apple</a>");
};
```



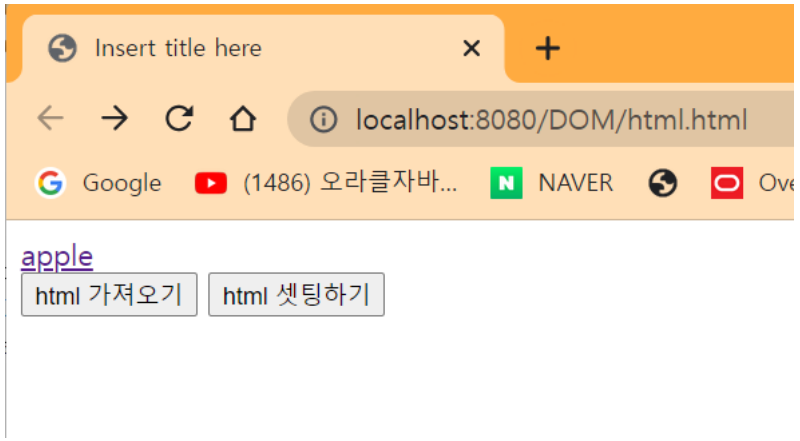
html

```
function getHtml(){
    var html = $("#a1").html();
    alert(html);
};
```



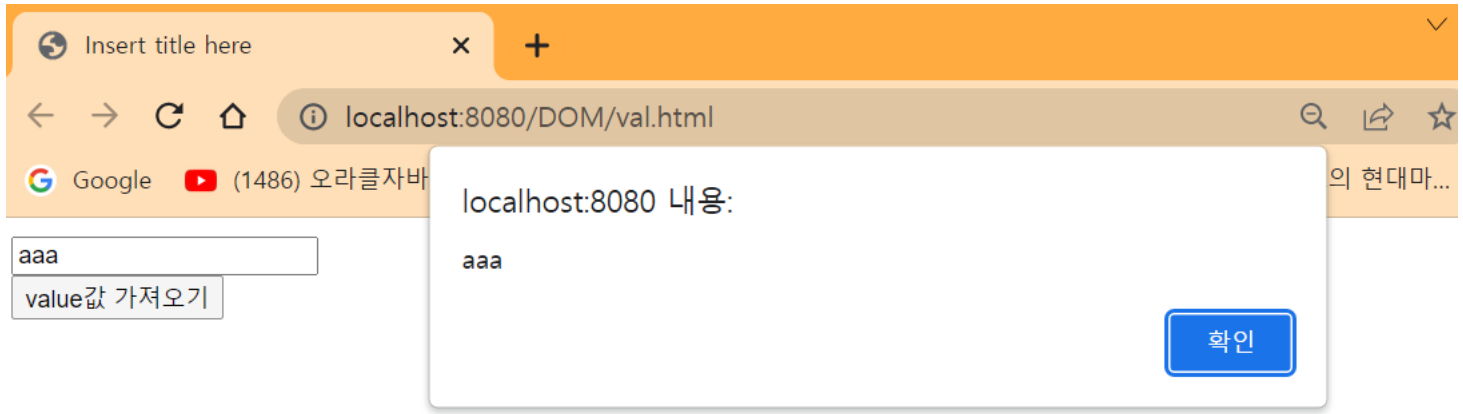
html

```
function setHtml(){
        $("#a1").html("<a href='http://www.apple.co.kr'>apple</a>");
};
```



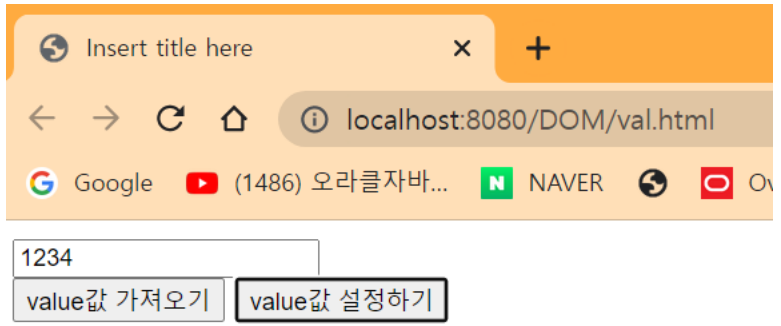
val

```
function getA1(){
    var str = $("#a1").val();
    alert(str);
};
```



val

```
function setA1(){
        $("#a1").val("1234");
};
```

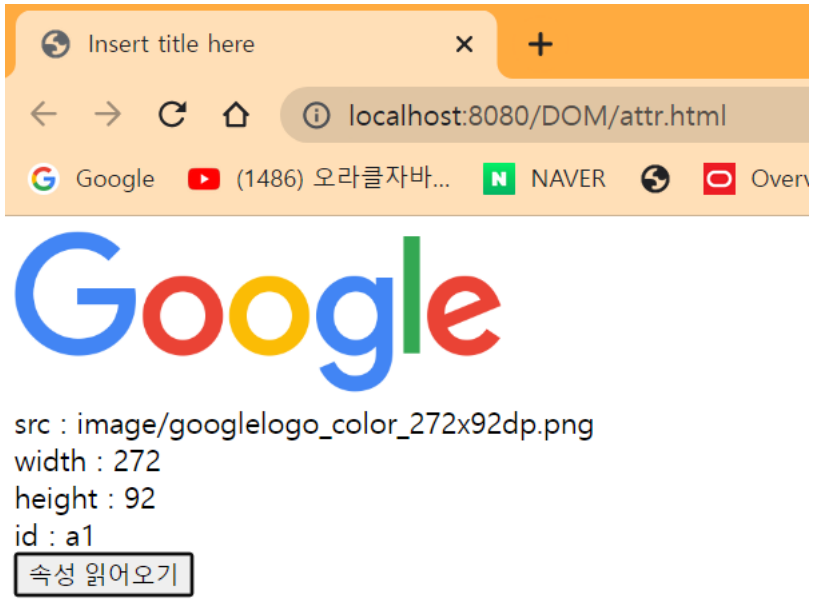


attr

- image 한 개 준비
- 프론젝트 – src – webapp - image 폴더 생성
- 준비한 image를 생성한 image폴더에 넣기

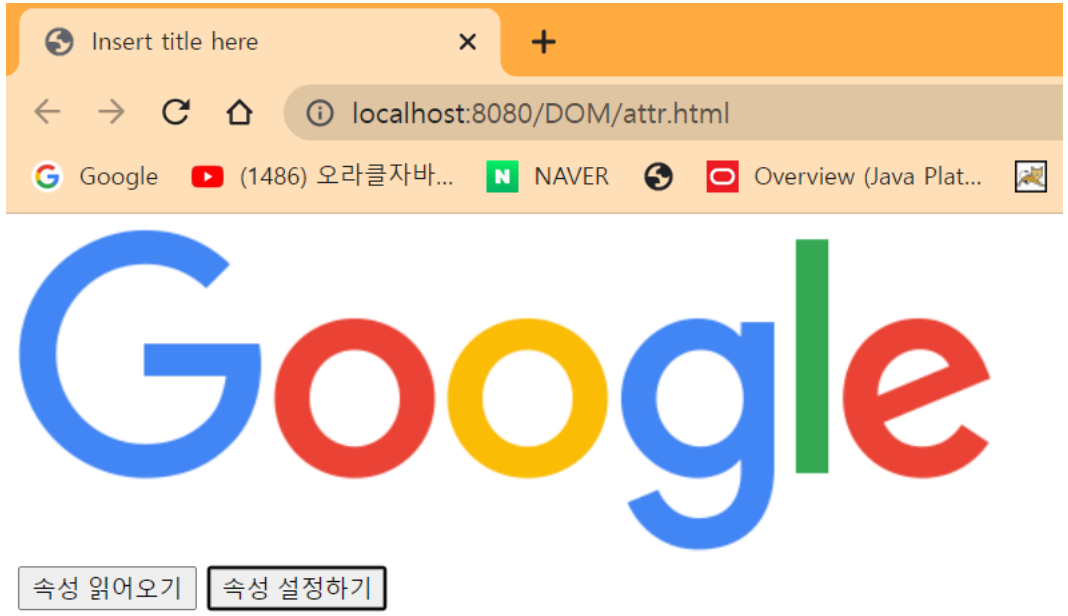
attr

```
function getAttr(){
    var src = $("#a1").attr("src");
    var width = $("#a1").attr("width");
    var height = $("#a1").attr("height");
    var id = $("#a1").attr("id");

    $("#result").html("src : " + src + "<br/>"
            + "width : " + width + "<br/>"
            + "height : " + height + "<br/>"
            + "id : " + id + "<br/>");
};
```



attr

```
function setAttr(){
    $("#a1").attr("width", 544);
    $("#a1").attr("height", 184);
};
```



정리

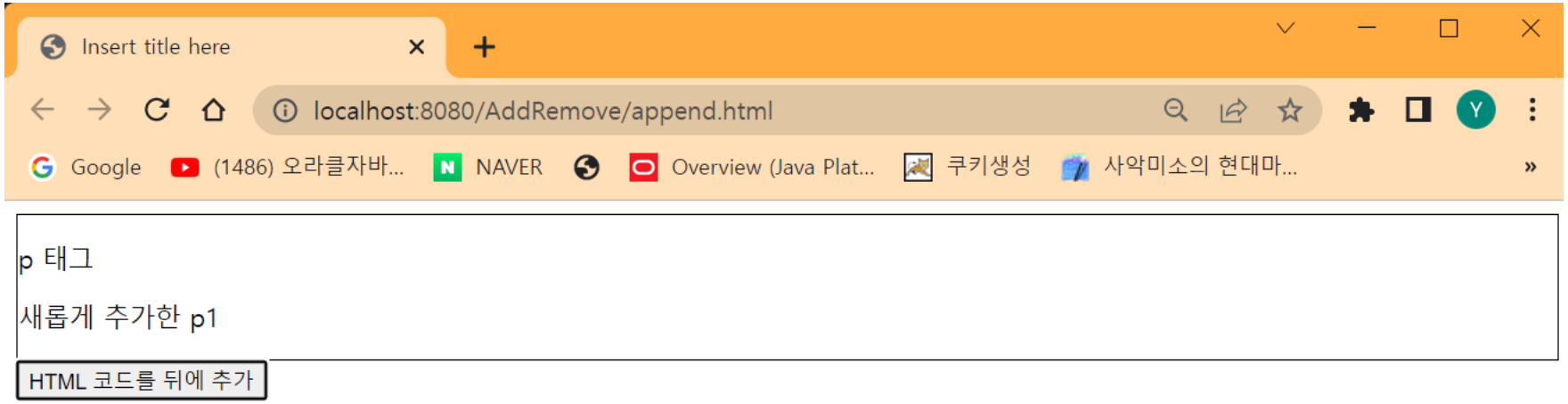
- text 함수를 사용하면 태그 내부의 문자열을 제어할 수 있다.
- html 함수를 사용하면 태그 내부의 html 코드를 제어할 수 있다.
- val 함수를 사용하면 입력 도구들의 value 속성을 제어할 수 있다.
- attr 함수를 사용하면 태그의 속성을 제어할 수 있다.

append, prepend

- append : html 코드나 태그를 태그 내부의 뒤에 추가
- prepend : html 코드나 태그를 태그 내부의 앞에 추가

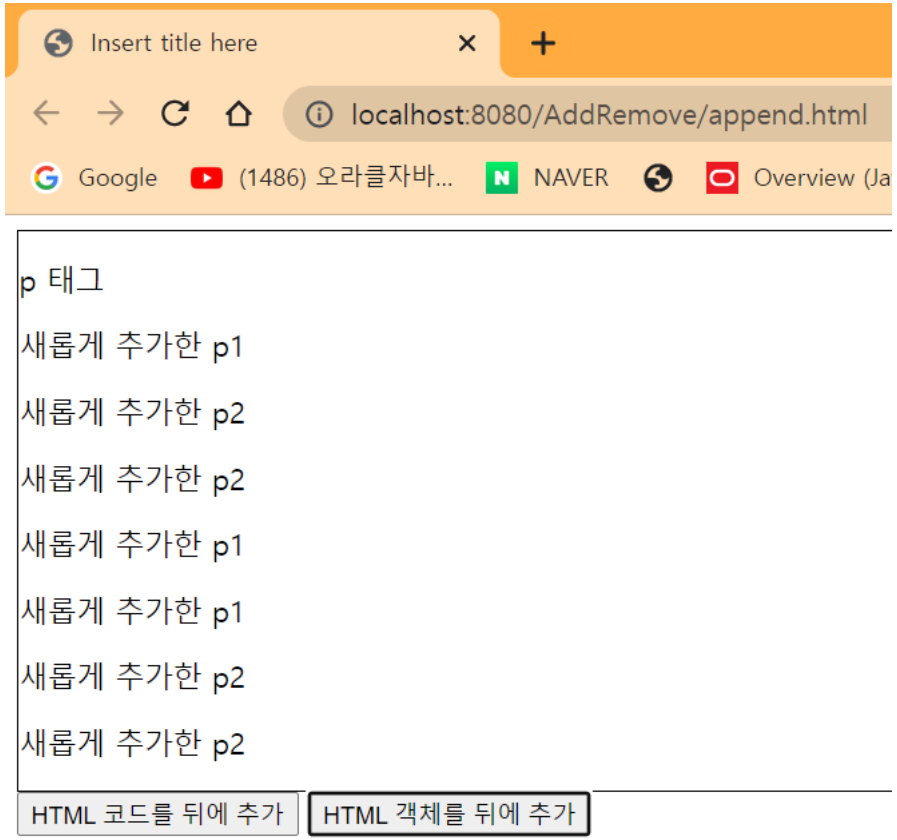
append 태그 추가

```
<script>
    function append1(){
        $("#a1").append("<p>새롭게 추가한 p1</p>");
    };
</script>
<style>
    #a1{
        border : 1px solid black;
    }
</style>
```



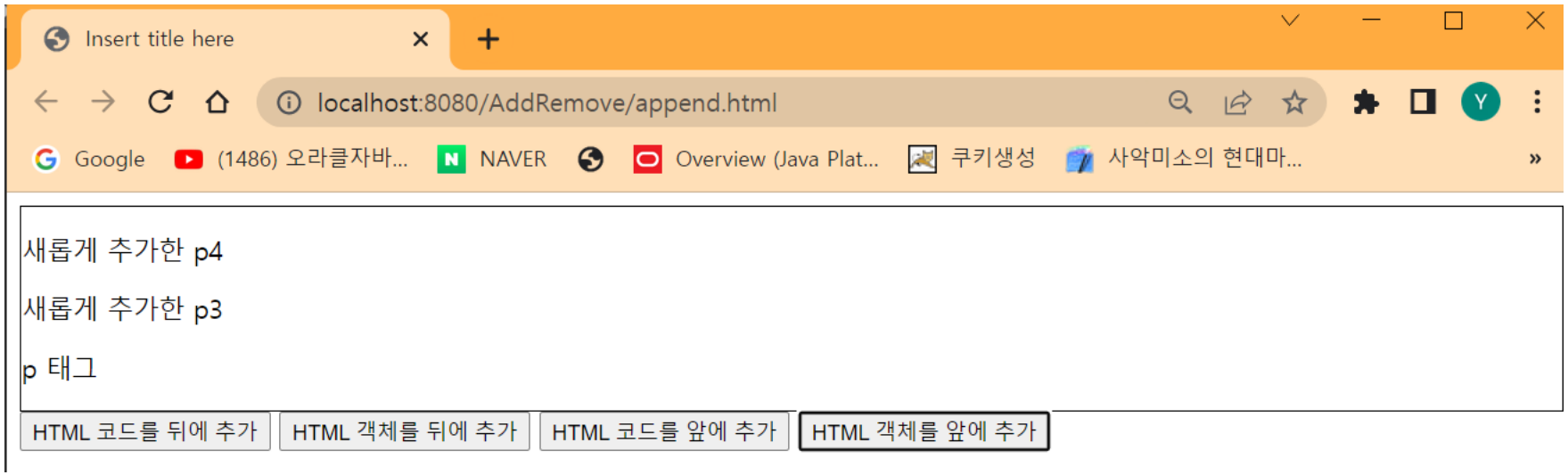
append 객체 추가

```
function append2(){
    var p = $("<p>");
    p.text("새롭게 추가한 p2");

    $("#a1").append(p);
};
<style>
    #a1{
        border : 1px solid black;
    }
</style>
```



repend 태그/객체 추가

```
function prepend1(){
        $("#a1").prepend("<p>새롭게 추가한 p3</p>");
};

function prepend2(){
    var p = $("<p>");
    p.text("새롭게 추가한 p4");

    $("#a1").prepend(p);
};
```

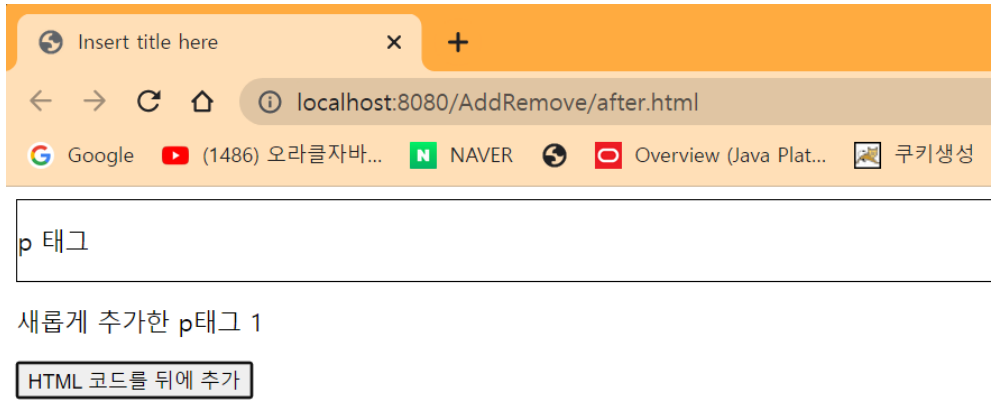


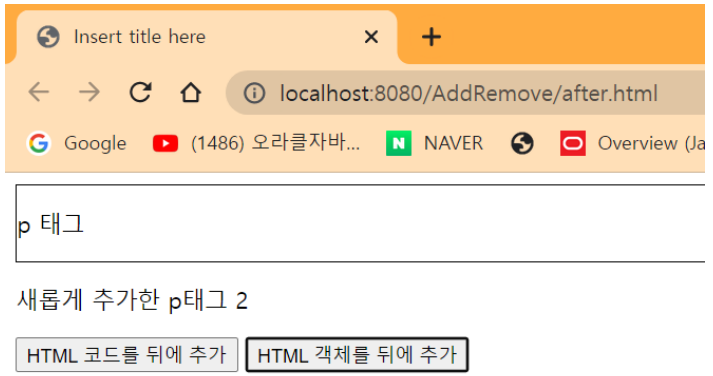
after, before

- after : html 코드나 태그를 태그 다음에 배치
- before : html 코드나 태그를 태그 앞에 배치

after 태그, 객체 추가

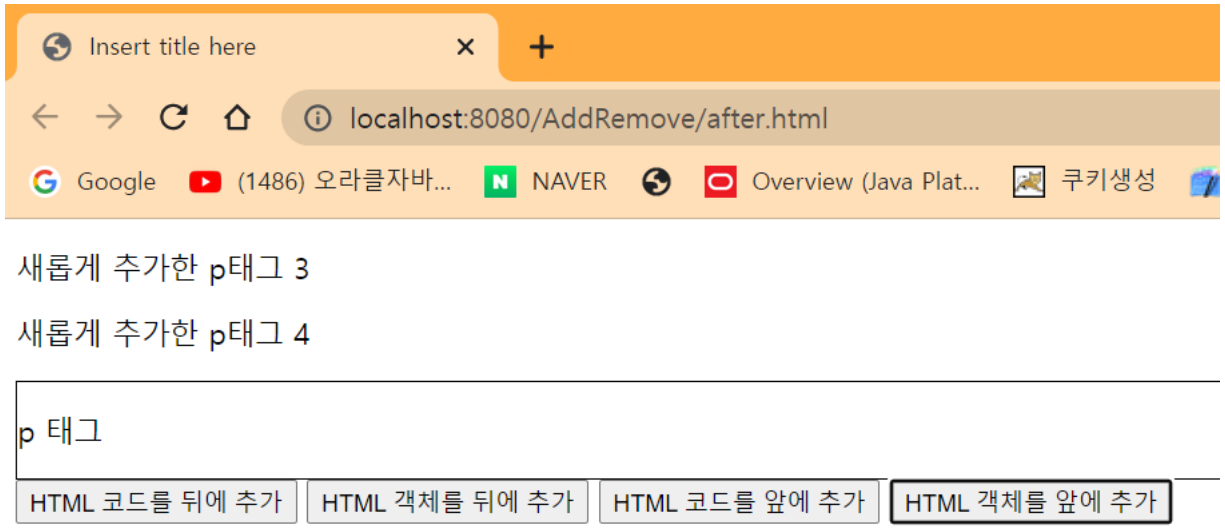
```
function after1(){
        $("#a1").after("<p>새롭게 추가한 p태그 1</p>");
};

function after2(){
    var p = $("<p>");
    p.text("새롭게 추가한 p태그 2");

    $("#a1").after(p);
};
```





before 태그, 객체 추가

```
function before1(){
        $("#a1").before("<p>새롭게 추가한 p태그 3</p>");
};

function before2(){
    var p = $("<p>");
    p.text("새롭게 추가한 p태그 4");

    $("#a1").before(p);
};
```

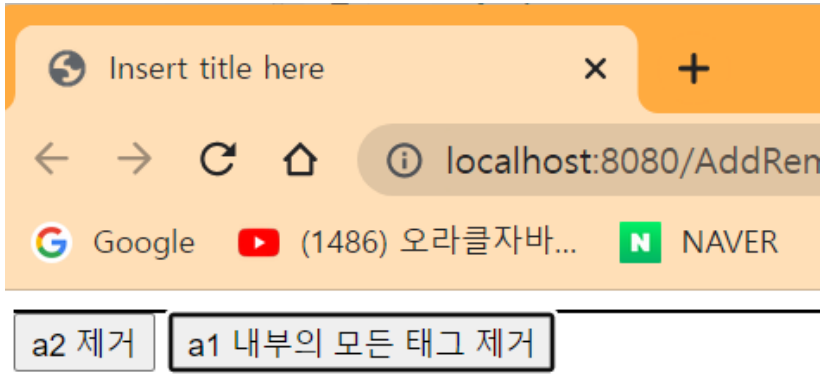


remove, empty

- remove : 태그를 제거
- empty : 태그 내의 모든 태그를 제거

remove, empty

```
function remove_a2(){
        $("#a2").remove();
};

function empty_a1(){
        $("#a1").empty();
};
```



정리

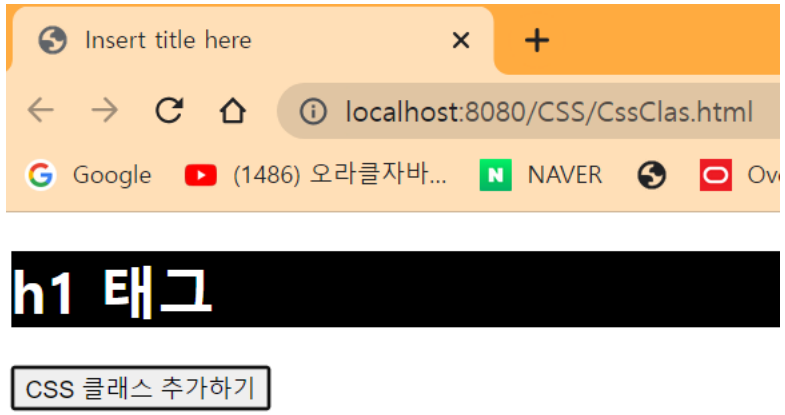
- append : html 코드나 태그를 태그 내부의 뒤에 추가
- prepend : html 코드나 태그를 태그 내부의 앞에 추가
- after : html 코드나 태그를 태그 뒤에 붙힘
- before : html 코드나 태그를 태그 앞에 붙힘
- remove : 태그를 제거
- empty : 태그 내의 모든 태그를 제거

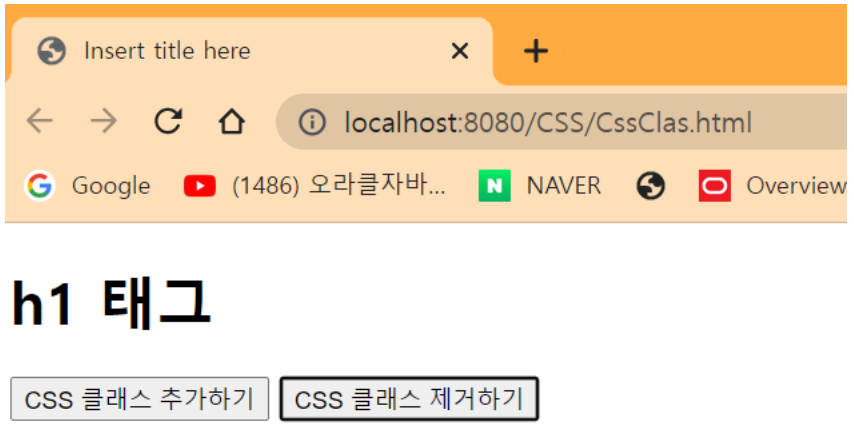
css 제어

- jQuery는 태그의 css를 제어할 수 있는 함수를 제공
- addClass : css class를 설정
- removeClass : css class를 제거
- toggleClass : 지정된 클래스가 없으면 설정하고 있으면 제거
- css : css 속성을 가져오거나 설정

addClass, removeClass

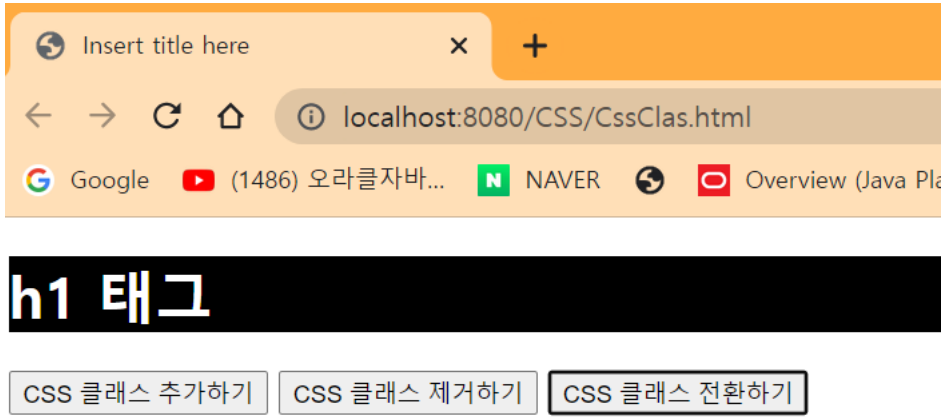
```
<script>
function addClass(){
        $("#a1").addClass("active");
};

function removeClass(){
        $("#a1").removeClass("active");
};

</script>
<style>
    .active{
        background-color : black;
        color : white;
    }
</style>
```





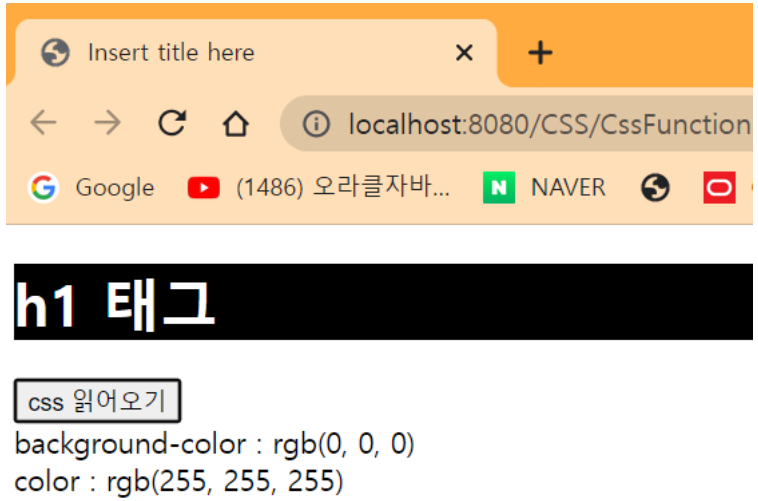
toggleClass

```
function toggleClass(){
        $("#a1").toggleClass("active");
}
```



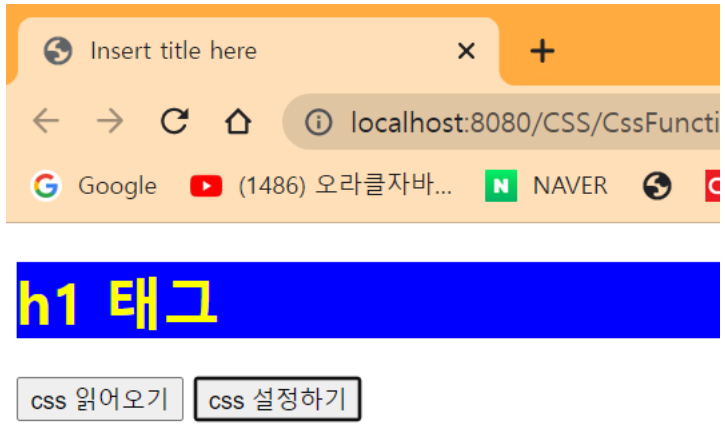
css 값 불러오기

```
<script>
    function getCss(){
        var v1 = $("#a1").css("background-color");
        var v2 = $("#a1").css("color");

        $("#result").append("background-color : " + v1 + "<br/>");
        $("#result").append("color : " + v2 + "<br/>");
    };
</script>
<style>
    #a1 {
        background-color : black;
        color : white;
    }
</style>
```



css 값 설정하기

```
function setCss(){
    $("#a1").css("background-color", "blue");
    $("#a1").css("color", "yellow");
};
```

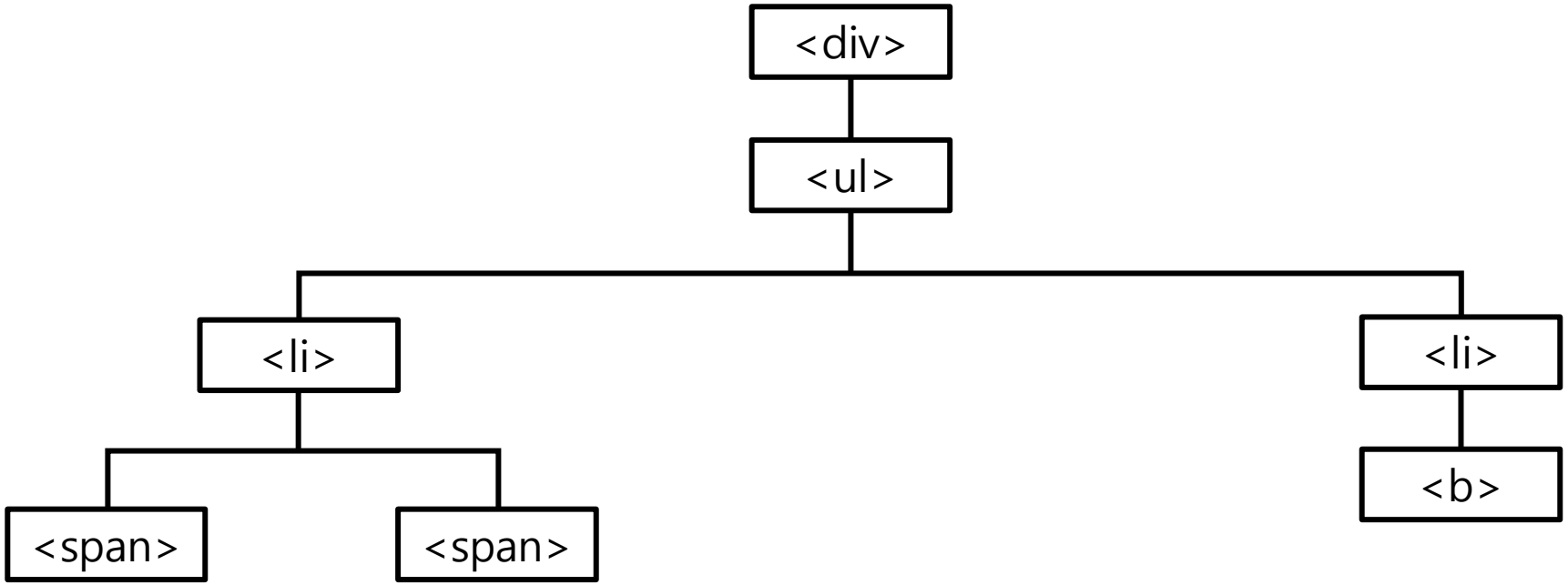


정리

- addClass : css class를 설정
- removeClass : css class를 제거
- toggleClass : 지정된 클래스가 없으면 설정, 있으면 제거
- css : css 속성을 가져오거나 설정

태그 탐색

- jQuery는 선택자를 통해 태그를 선택한 후 선택된 태그를 기준으로 다른 태그들을 탐색 할 수 있다.
- 태그 탐색은 html 문서를 탐색할 경우 사용하지만 xml 문서를 탐색하는 용도로 사용하기도 한다.

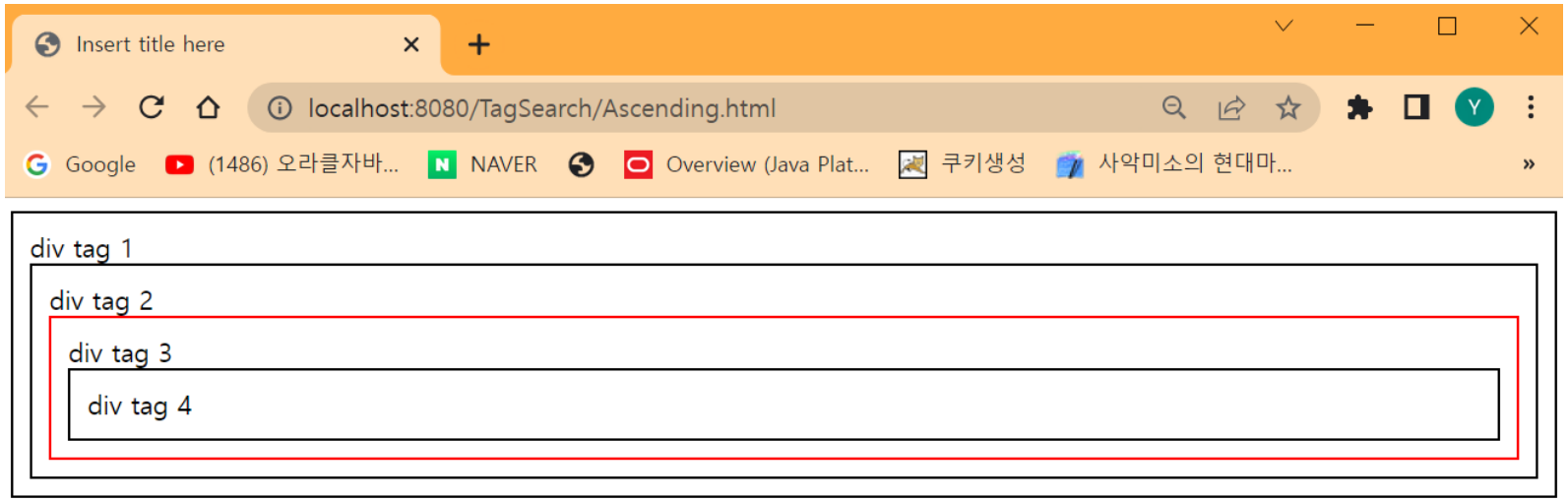


부모 태그 탐색

- parent : 선택된 태그의 부모 태그를 선택
- parents : 선택된 태그의 모든 부모 태그를 선택
- parents(선택자2) : 선택된 태그의 모든 부모 태그 중 선택자2에 해당하는 태그들이 선택
- parentsUntil(선택자2) : 선택된 태그에서 선택자2 태그까지의 모든 부모태그들이 선택

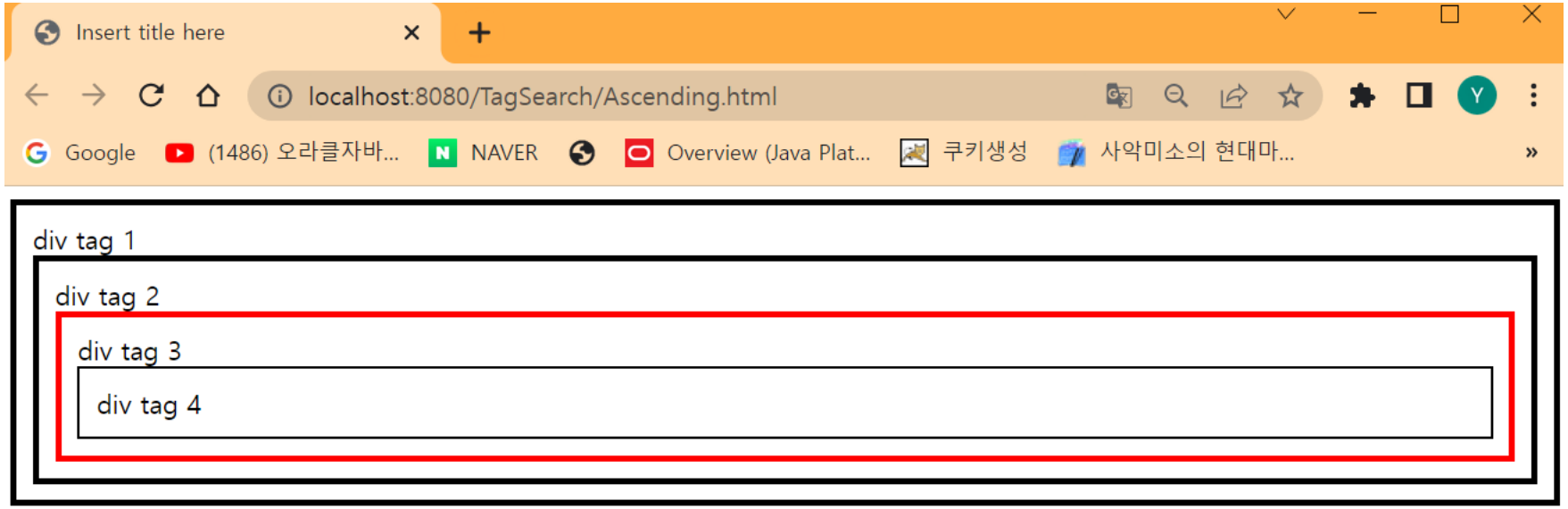
parent

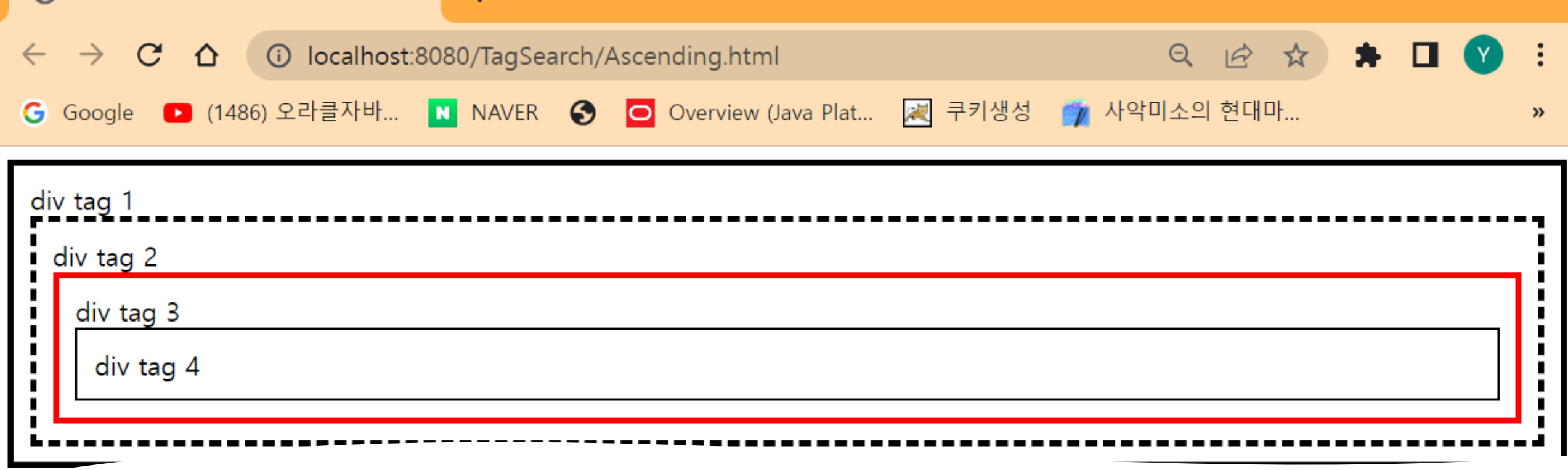
```
<script>
    $(function(){
        $("#a4").parent().css("border-color", "red");
    });
</script>
<style>
    div {
        border : 2px solid black;
        padding : 10px;
    }
</style>
```



parents

```
<script>
$(function(){
    $("#a4").parent().css("border-color", "red");
    $("#a4").parents().css("border-width", "4px");
});
</script>
```



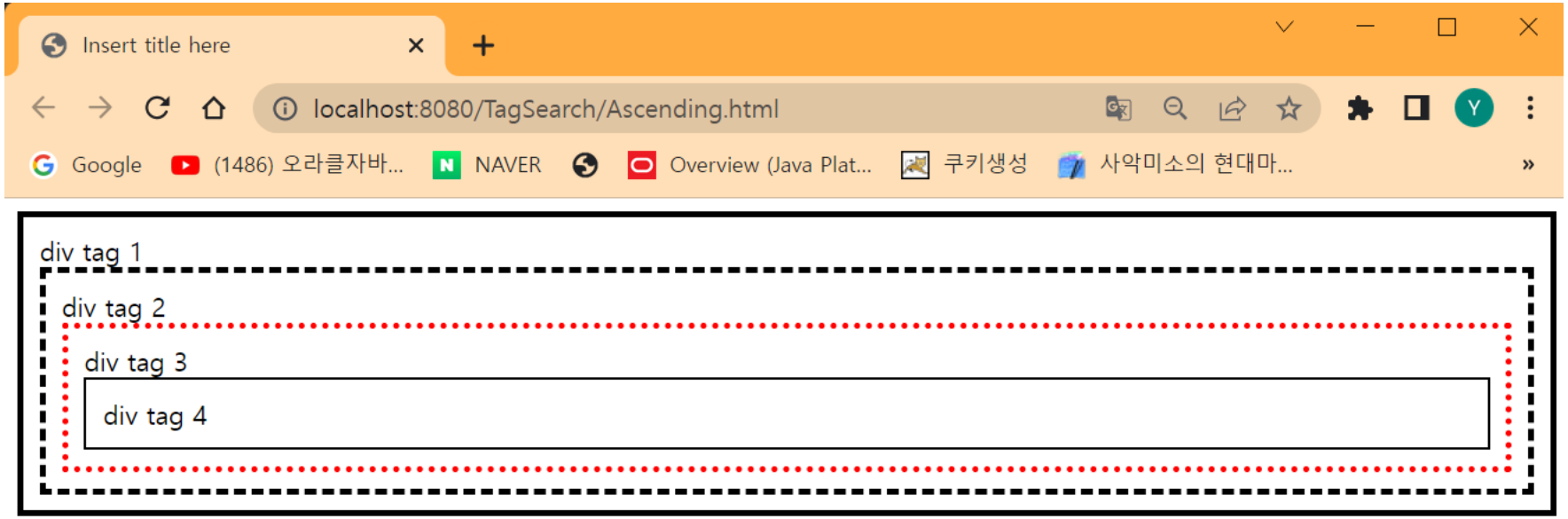


parents(선택자2)

- $(**function(){**
  - $("#a4").parent().css("border-color", "red");
  - $("#a4").parents().css("border-width", "4px");
  - $("#a4").parents(".c1").css("border-style", "dashed");
- });

parentsUntil(선택자2)

```
<script>
    $(function(){
        $("#a4").parent().css("border-color", "red");
        $("#a4").parents().css("border-width", "4px");
        $("#a4").parents(".c1").css("border-style", "dashed");
        $("#a4").parentsUntil(".c1").css("border-style", "dotted");
    });
</script>
```

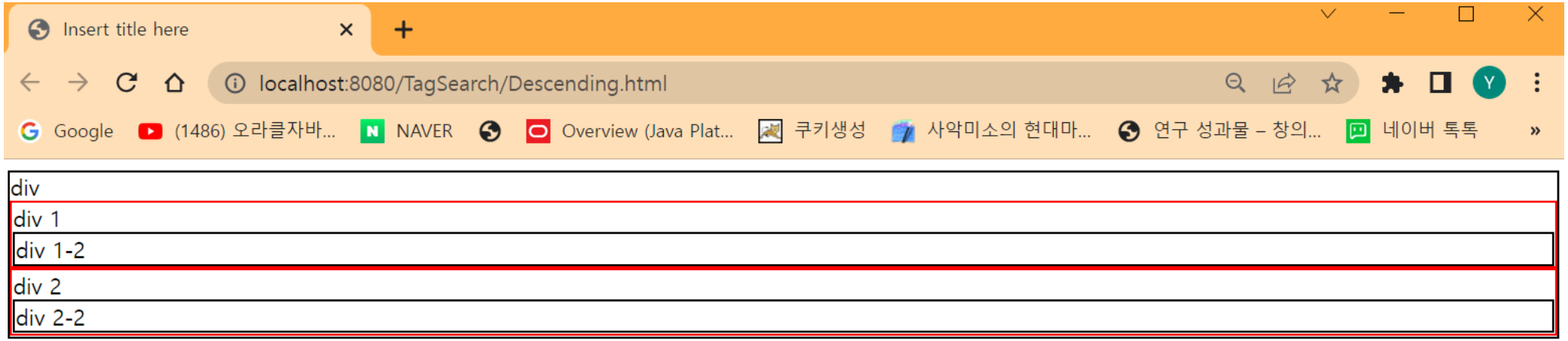


자식 태그 탐색

- children : 선택된 태그의 자식 태그들을 선택
- children(선택자2) : 선택된 태그의 자식 태그들 중 선택자2에 해당하는 태그들이 선택
- find(선택자2) : 선택된 태그의 하위 태그들 중 선택자2에 해당하는 태그들이 선택

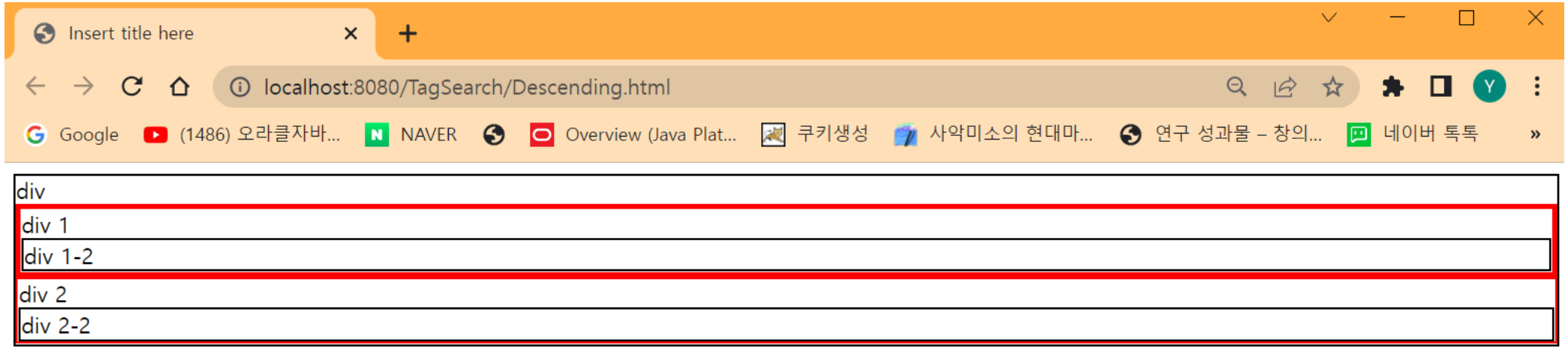
children

```
<script>
    $(function(){
        $("#a1").children().css("border-color","red");
    });
</script>
<style>
    div {
        border : 2px solid black;
        pading : 10px;
    }
</style>
```



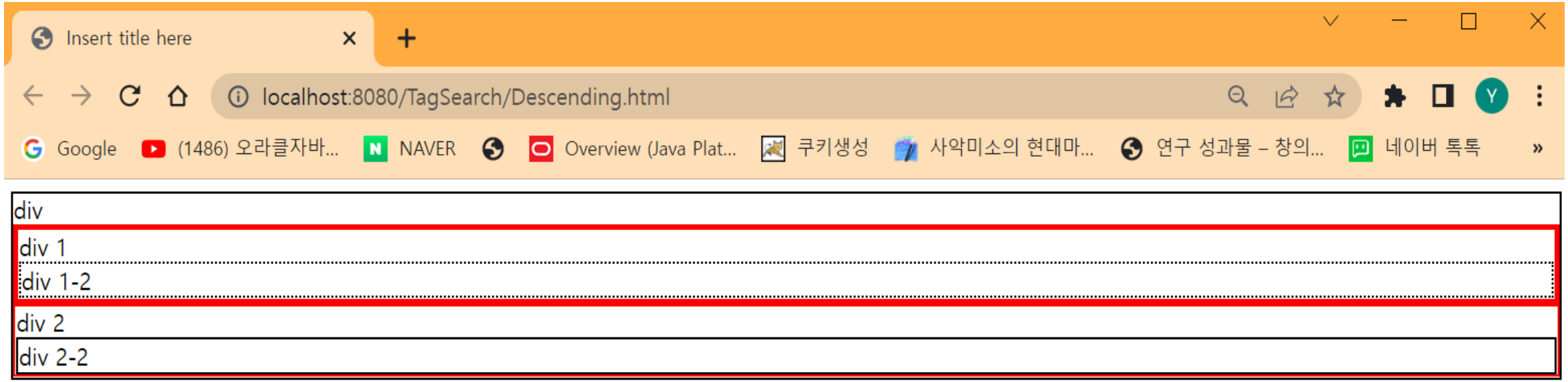
children(선택자2)

```
<script>
    $(function(){
        $("#a1").children().css("border-color","red");
        $("#a1").children(".c1").css("border-width","4px");
    });
</script>
```



find(선택자2)

```
<script>
    $(function(){
        $("#a1").children().css("border-color","red");
        $("#a1").children(".c1").css("border-width","4px");
        $("#a1").find(".c3").css("border-style","dotted");
    });
</script>
```



정리

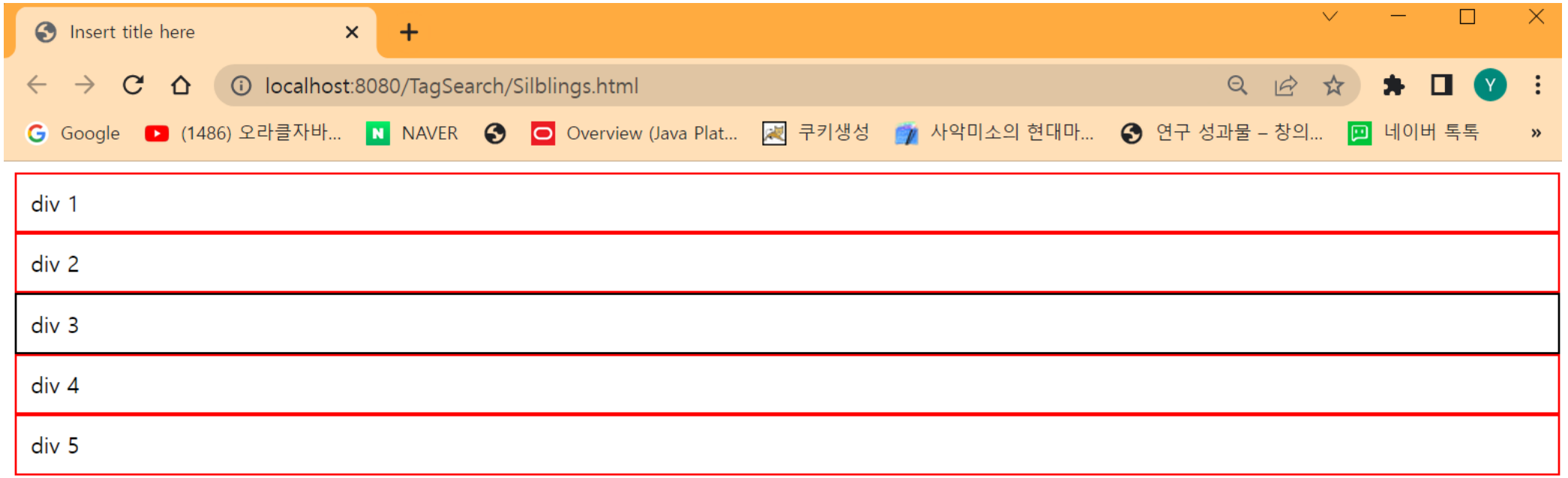
- parent, parents, parentsUntil을 사용하면 부모 태그들을 선택할 수 있다.
- children, find를 사용하면 자식 태그들을 선택할 수 있다.

같은 계층의 태그 선택

- siblings : 선택된 태그와 같은 계층의 모든 태그들이 선택
- siblings(선택자2) : 선택된 태그와 같은 계층의 모든 태그 중 선택자2에 해당하는 태그들이 선택
- next : 선택된 태그 다음 태그가 선택
- nextAll : 선택된 태그의 다음 태그들이 모두 선택
- nextUntil(선택자2) : 선택된 태그 다음 태그들 중 선택자2 까지의 모든 태그들이 선택

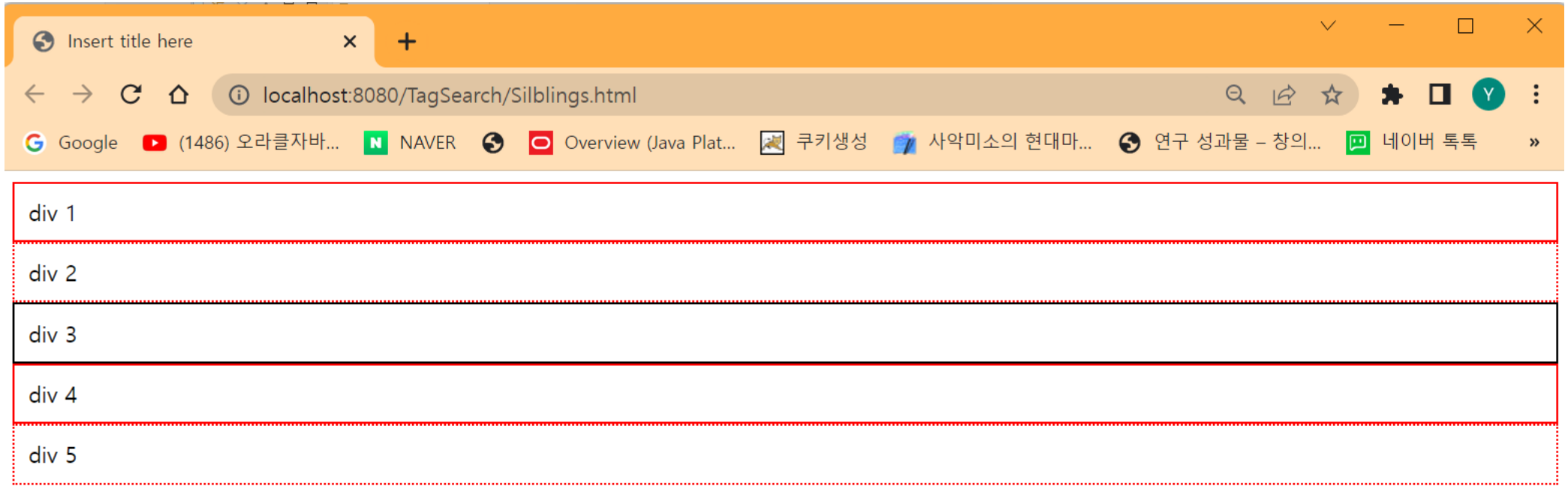
siblings

```
<script>
    $(function(){
        $("#a1").siblings().css("border-color", "red");
    });
</script>
```



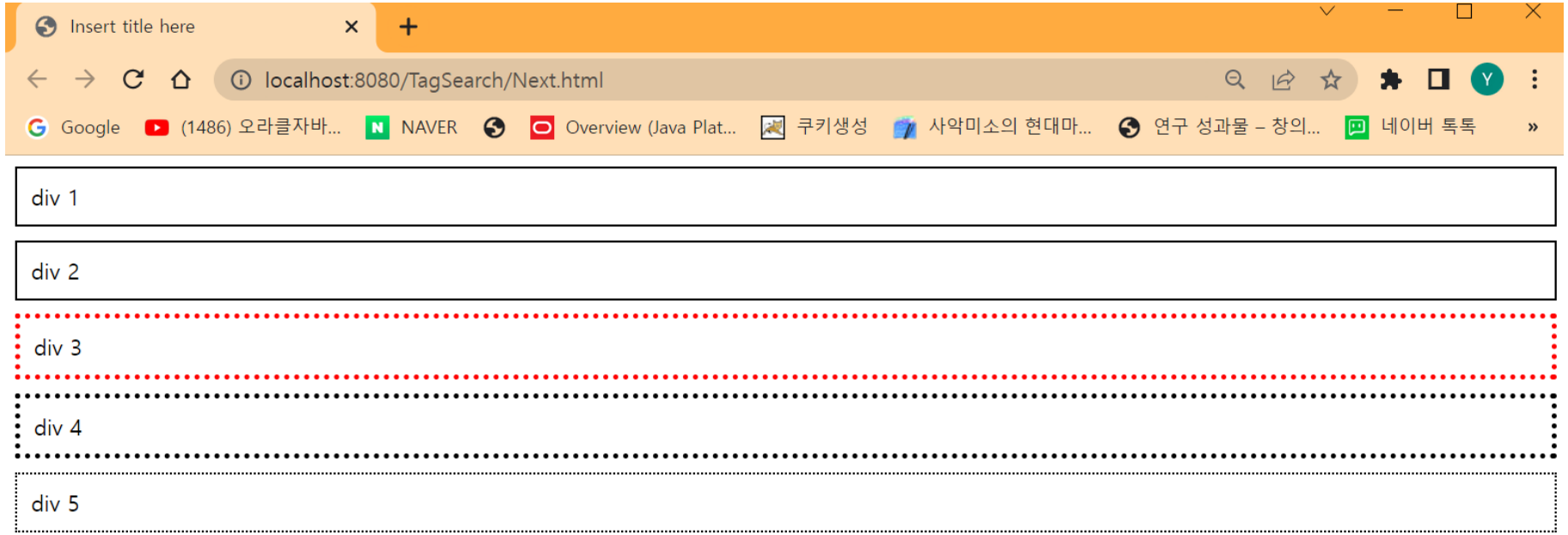
siblings(선택자2)

```
<script>
    $(function(){
        $("#a1").siblings().css("border-color", "red");
        $("#a1").siblings(".c2").css("border-style", "dotted");
    });
</script>
```



next, nextAll, nextUntil(선택자2)

```
<script>
    $(function(){
        $("#a1").next().css("border-color", "red");
        $("#a1").nextAll().css("border-style", "dotted");
        $("#a1").nextUntil("#a2").css("border-width", "4px");
    });
</script>
```

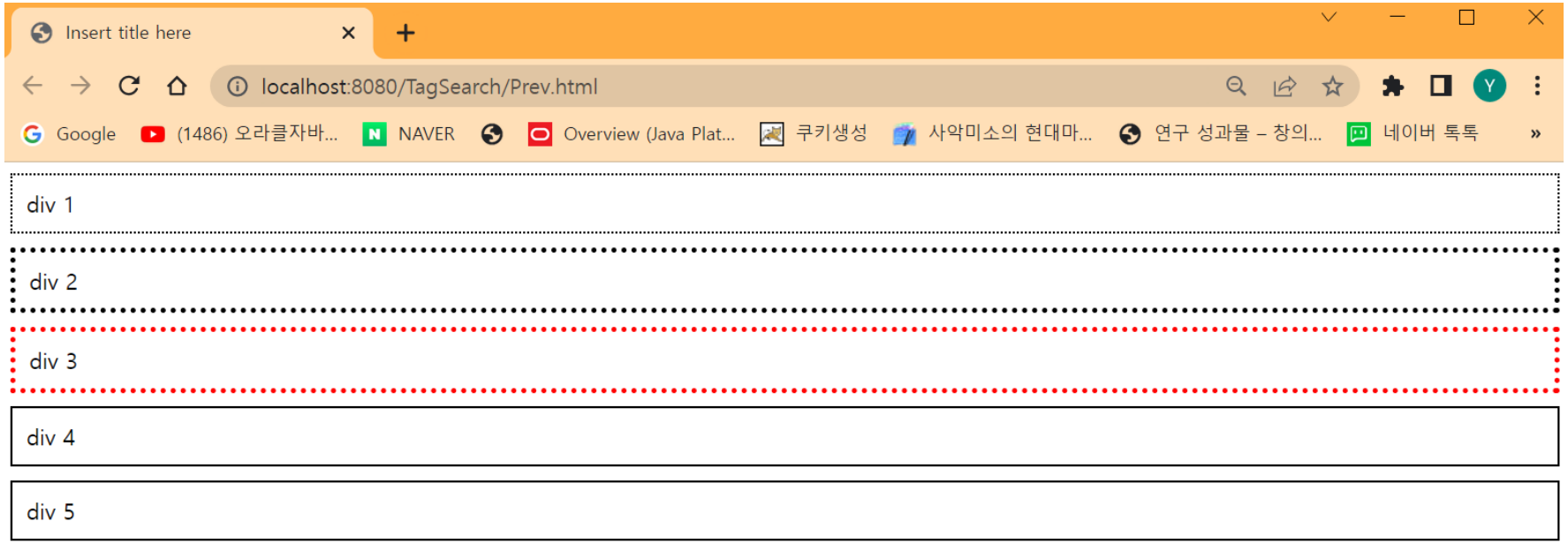


같은 계층의 태그 선택

- prev : 선택된 태그 이전 태그가 선택
- prevAll : 선택된 태그 이전의 모든 태그가 선택
- prevUntil(선택자2) : 선택된 태그의 이전 태그들 중 선택자2 까지의 모든 태그들이 선택

prev, prevAll, prevUntil(선택자2)

```
<script>
    $(function(){
        $("#a1").prev().css("border-color","red");
        $("#a1").prevAll().css("border-style","dotted");
        $("#a1").prevUntil("#a2").css("border-width","4px");
    });
</script>
```



정리

- siblings를 사용하면 같은 계층의 모든 태그들을 선택할 수 있다.
- next, nextAll nextUntil을 사용하면 다음 태그들을 선택할 수 있다.
- prev, prevAll, prevUntil을 사용하면 이전 태그들을 선택할 수 있다.