

jQuery

선택자

jQuery 함수

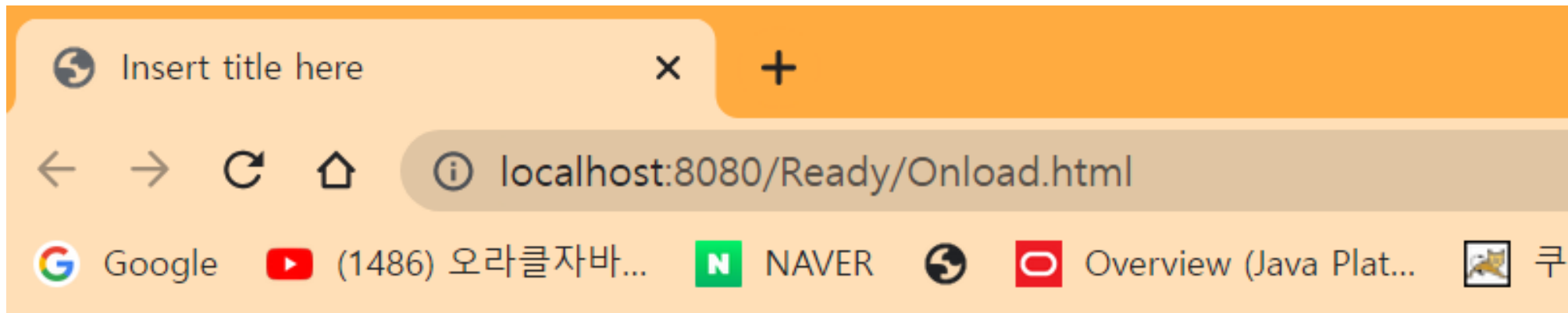
- jQuery에서 가장 중요한 함수
- JavaScript는 생성된 객체 멤버를 추가하는 것이 자유롭다. 이것을 이용해 만들어진 라이브러리가 jQuery 라이브러리
- JavaScript에서 객체를 jQuery 함수에 매개변수로 넣어주면 객체에 jQuery에서 만든 여러 함수를 추가하여 반환
- jQuery 함수는 jQuery(객체) 형태로 작성하지만 jQuery 6글자 대신에 \$(객체)로 작성할 수도 있다.

Ready 함수

- JavaScript의 window.onload와 흡사
- Window.onload는 태그 객체가 모두 생성되면 발생하는 이벤트로 웹 화면이 처음 나타날 때 필요한 작업이 있다면 여기서 구현
- Ready도 동일한 시점에 동작
- 차이점은 window.onload는 여러 개 있을 경우 제일 마지막 것만 동작, Ready는 모든 부분이 동작

window.onload

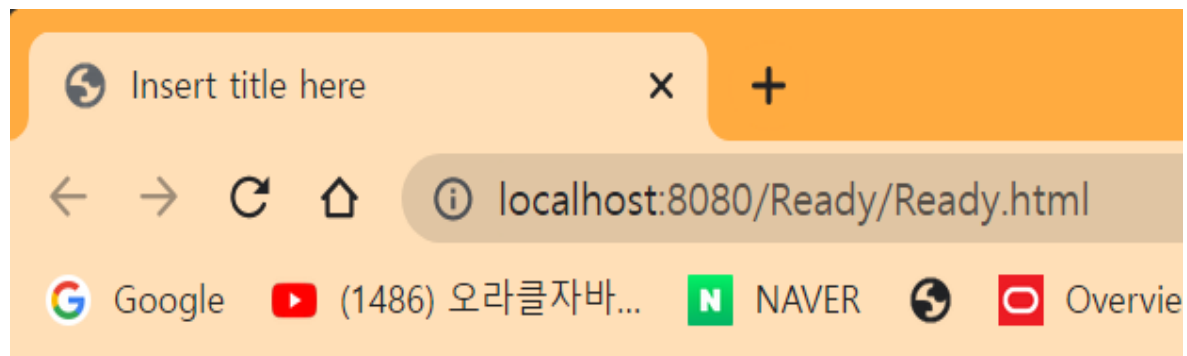
```
<script>
  window.onload = function() {
    var result = document.getElementById("result");
    result.innerHTML = "on load<br/>";
  }
</script>
```



on load

ready

```
<script>
  $(document).ready(function() {
    $("#result").append("ready<br/>");
  });
</script>
```



ready

정리

- jQuery 함수는 JavaScript 객체에 jQuery에서 정의한 멤버를 추가하여 반환하는 함수
- jQuery(객체) 형태로 사용하지만 \$(객체)로 사용할 수 있다
- Ready는 window.onload와 다르게 모든 부분이 실행

jQuery 선택자

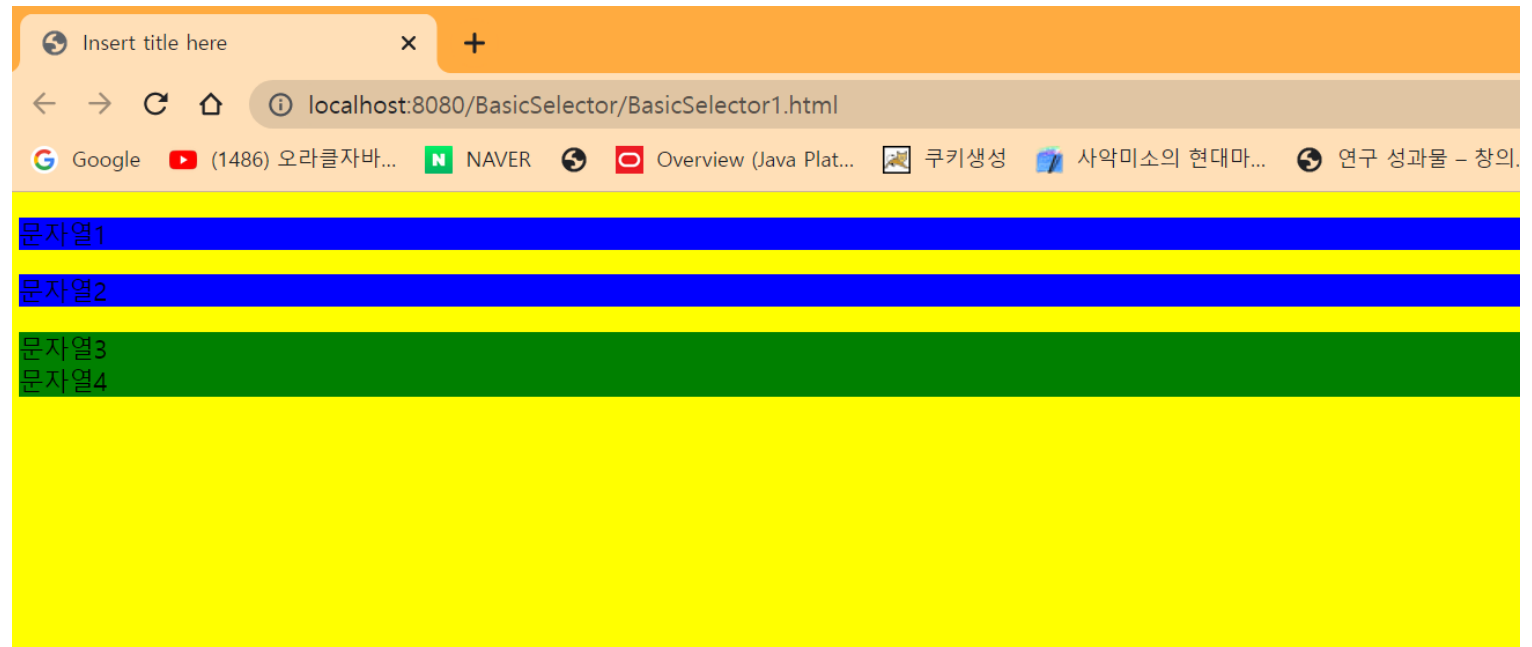
- 기본 선택자
- 기본 선택자 조합
- 순서 필터 선택자
- 인덱스 필터 선택자
- 속성 선택자
- 상태 선택자
- Form 태그 선택자

기본 선택자

- 가장 기본이 되는 선택자, 다른 선택자는 기본 선택자를 조합하여 만듦
- 전체 선택자(*) : HTML 문서 내의 모든 태그를 선택
- 태그이름 선택자(태그이름) : 지정한 태그이름에 해당하는 태그 객체를 가져 온다
- 아이디 선택자(#아이디) : id 속성을 통해 태그 객체를 가져온다
- 클래스 선택자(.클래스) : class 속성을 통해 태그 객체를 가져온다

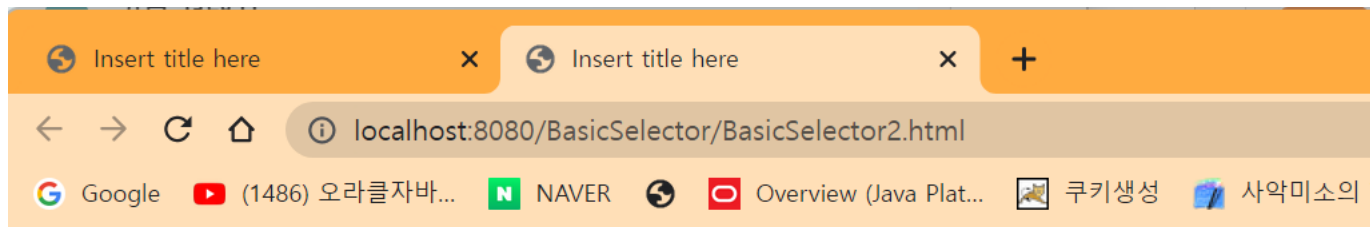
기본 선택자

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    $(function() {
        $("*").css("background-color", "yellow");
        $("p").css("background-color", "blue");
        $("div").css("background-color", "green");
    });
</script>
```



기본 선택자

```
<script>
  $(function() {
    $("#test1").css("color", "blue");
    $("#test2").css("color", "red");
    $(".test3").css("color", "orange");
  });
</script>
```



문자열1

문자열2

문자열3

문자열4

정리

- 선택자는 HTML 문서에 있는 태그 객체를 가져올 수 있는 방법
- 기본 선택자는 전체 선택자, 아이디 선택자, 태그이름 선택자, 클래스 선택자로 구성

기본 선택자 조합

<div class="a1"> div class a1 </div>

<div class="a2"> div class a2 </div>

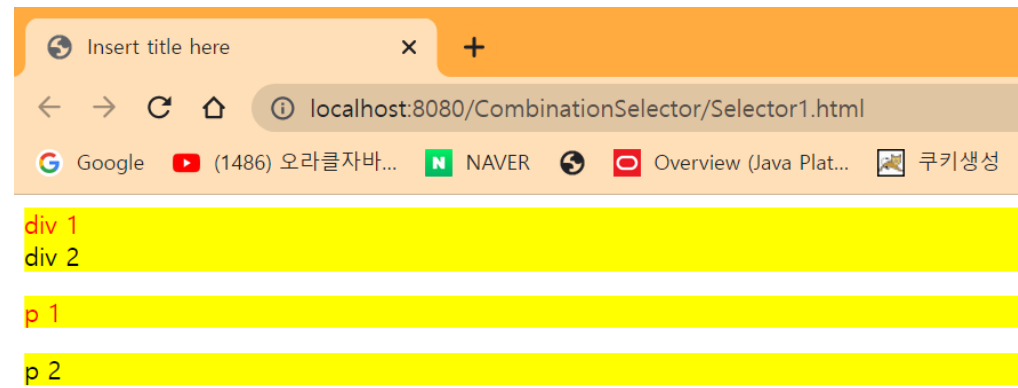
<p class="a3"> p class a3 </p>

<p class="a1"> p class a1 </p>

기본 선택자 조합

- 선택자1, 선택자2 : 동시에 여러 태그를 선택할 때 사용
- 선택자1 > 선택자2 : 태그의 바로 하위의 태그들을 선택할 때 사용
- 선택자1 선택자2 : 태그 내부의 모든 태그들을 선택할 때 사용
- 선택자1.선택자2 : 모든 선택자를 만족하는 태그를 선택
- 선택자1 + 선택자2 : 선택자1 태그 바로 다음에 있는 선택자2 태그를 선택
- 선택자1 ~ 선택자2 : 선택자1 태그 다음에 있는 모든 선택자2 태그를 선택

기본 선택자 조합



- 선택자1, 선택자2 : 동시에 여러 태그를 선택할 때 사용

```
<script>
    $(function() {
        $("div").css("background-color", "yellow");
        $("p").css("background-color", "yellow");

        //$("div, p").css("background-color", "yellow");

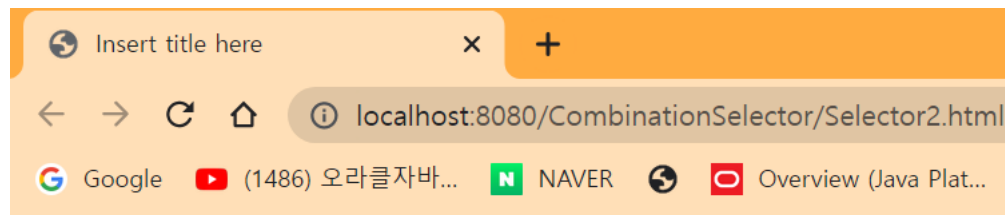
        $("#a1").css("color", "red");
        $("#a2").css("color", "red");

        //$("#a1, #a2").css("color", "red");
    });
</script>
```

기본 선택자 조합

- 선택자1 > 선택자2 : 태그의 바로 하위의 태그들을 선택할 때 사용

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    $(function() {
        $("#a1 > h1").css("background-color", "yellow");
    });
</script>
</head>
```



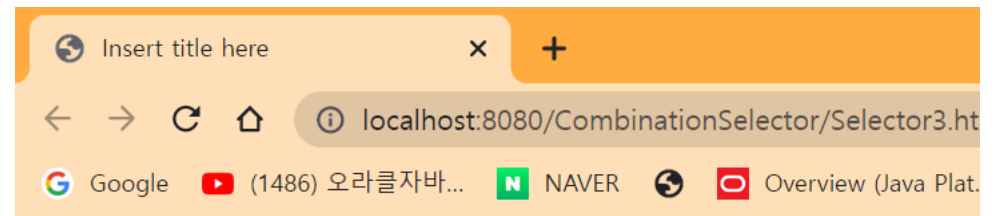
자식 태그 h1

자손 태그 h1

기본 선택자 조합

- 선택자1 선택자2 : 태그 내부의 모든 태그들을 선택할 때 사용

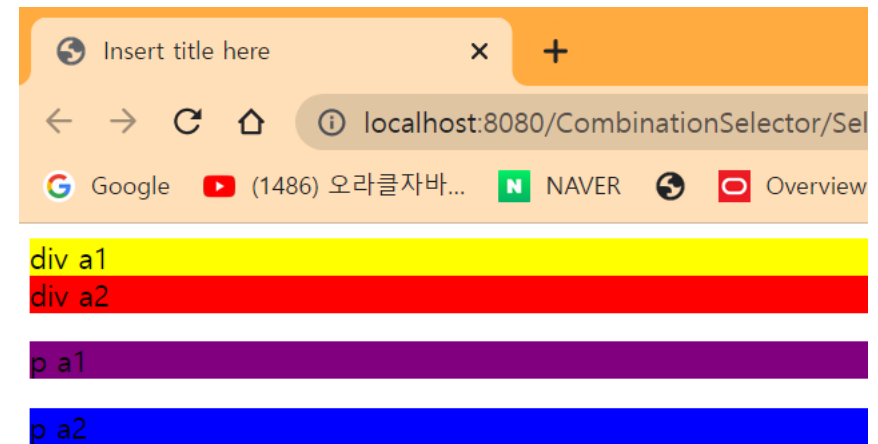
```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
  $(function() {
    $("#a1 h1").css("background-color", "yellow");
  });
</script>
</head>
```



자식 태그 h1

자손 태그 h1

기본 선택자 조합



- 선택자1.선택자2 : 모든 선택자를 만족하는 태그들 선택

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
```

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

```
</script>
```

```
<script>
```

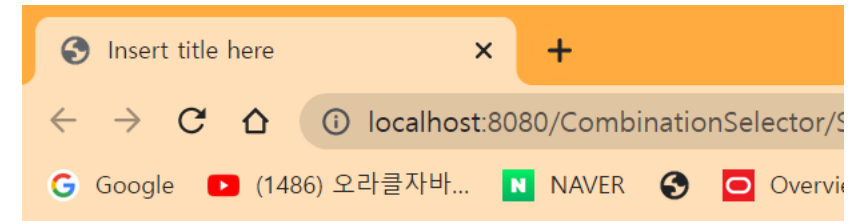
```
    $(function() {
        $("div.a1").css("background-color", "yellow");
        $("div.a2").css("background-color", "red");
        $("p.a1").css("background-color", "purple");
        $("p.a2").css("background-color", "blue");
    });
```

```
</script>
```

기본 선택자 조합

- 선택자1 + 선택자2 : 선택자1 태그 바로 다음에 있는 선택자2 태그를 선택

```
<script>
  $(function() {
    $("#a1 + p").css("background-color", "yellow");
  });
</script>
```



p 태그

p 태그

p 태그

p 태그 a1

p 태그

p 태그

p 태그

p 태그

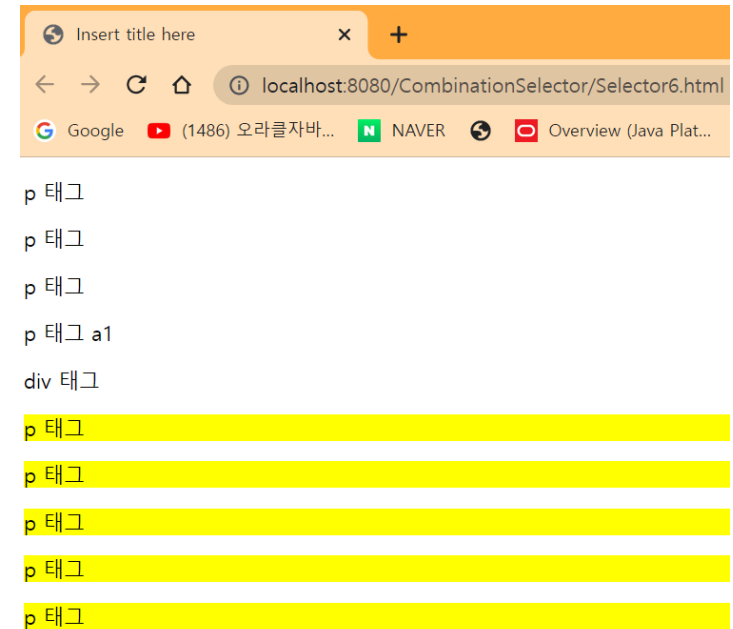
p 태그

p 태그

기본 선택자 조합

- 선택자1 ~ 선택자2 : 선택자1 태그 다음에 있는 모든 선택자2 태그를 선택

```
<script>
    $(function() {
        $("#a1 ~ p").css("background-color", "yellow");
    });
</script>
```



정리

- 기본 선택자를 조합하여 하나의 선택자를 만들 수 있다.

순서필터 선택자

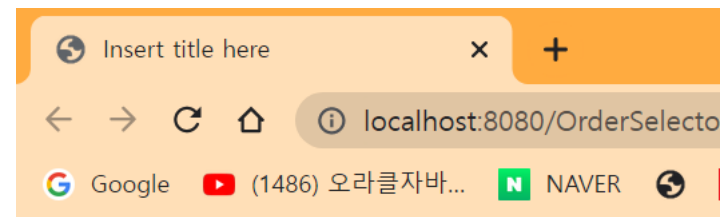
- 선택자를 통해 선택된 태그 중에 태그의 상태에 따라 선택할 수 있는 선택자
- 순서, 태그의현재 상태 등

첫 번째 태그 선택

- :first : 선택자에 해당하는 태그 중 가장 처음 태그
- :first-child : 선택자에 해당하는 태그 중 각 영역별 첫 번째 태그
가 선택. 처음 태그가 선택자에 해당하지 않으면 선택되지 않음
- :first-of-type : 선택자에 해당하는 태그 중 각 영역별 처음 태그
가 선택

:first

```
<script>
    $(function() {
        $("p:first").css("background-color", "yellow");
    });
</script>
```



태그

div 안의 p 태그

div 안의 p 태그

div 안의 p 태그

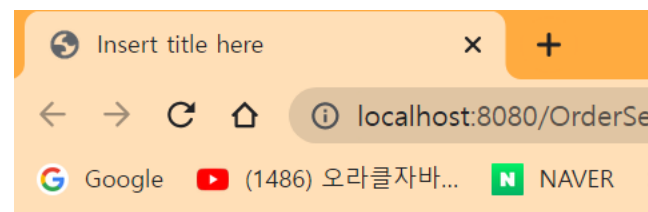
div 안의 div 태그

div 안의 p 태그

div 안의 p 태그

:first-child

```
<script>
  $(function() {
    $("p:first-child").css("color", "orange");
  });
</script>
```



태그

div 안의 p 태그

div 안의 p 태그

div 안의 p 태그

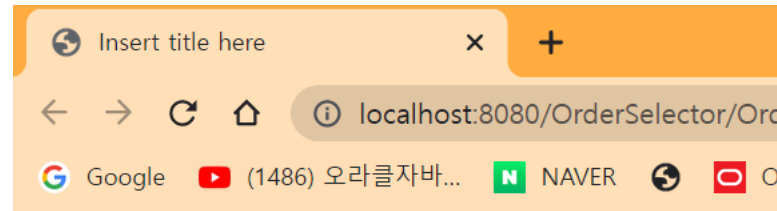
div 안의 div 태그

div 안의 p 태그

div 안의 p 태그

:first-of-type

```
<script>
  $(function() {
    $("p:first-of-type").css("font-size", "30px");
  });
</script>
```



태그

div 안의 p 태그

div 안의 p 태그

div 안의 p 태그

div 안의 div 태그

div 안의 p 태그

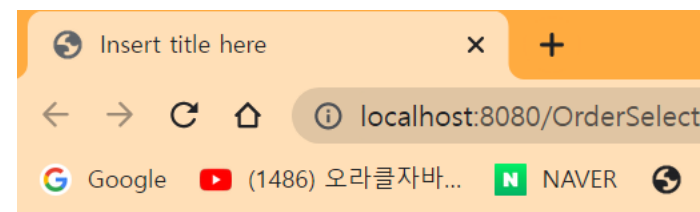
div 안의 p 태그

마지막 태그 선택

- :last : 선택자에 해당하는 태그 중 가장 마지막 태그가 선택
- :last-child : 선택자에 해당하는 태그 중 각 영역별 마지막 태그가 선택. 마지막 태그가 선택자에 해당하지 않으면 선택되지 않음
- :last-of-type : 선택자에 해당하는 태그 중 각 영역별 마지막 태그가 선택

:last

```
<script>
  $(function() {
    $("p:last").css("background-color", "yellow");
  });
</script>
```



태그

div 안의 p 태그

div 안의 p 태그

div 안의 p 태그

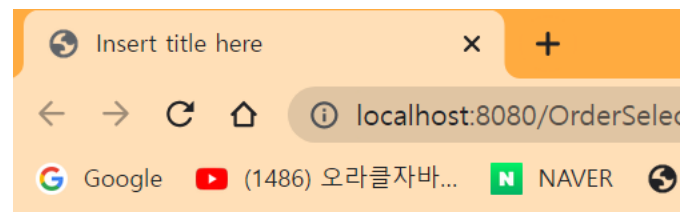
div 안의 p 태그

div 안의 p 태그

div 안의 div 태그

:last-child

```
<script>
  $(function() {
    $("p:last-child").css("color", "orange");
  });
</script>
```



태그

div 안의 p 태그

div 안의 p 태그

div 안의 p 태그

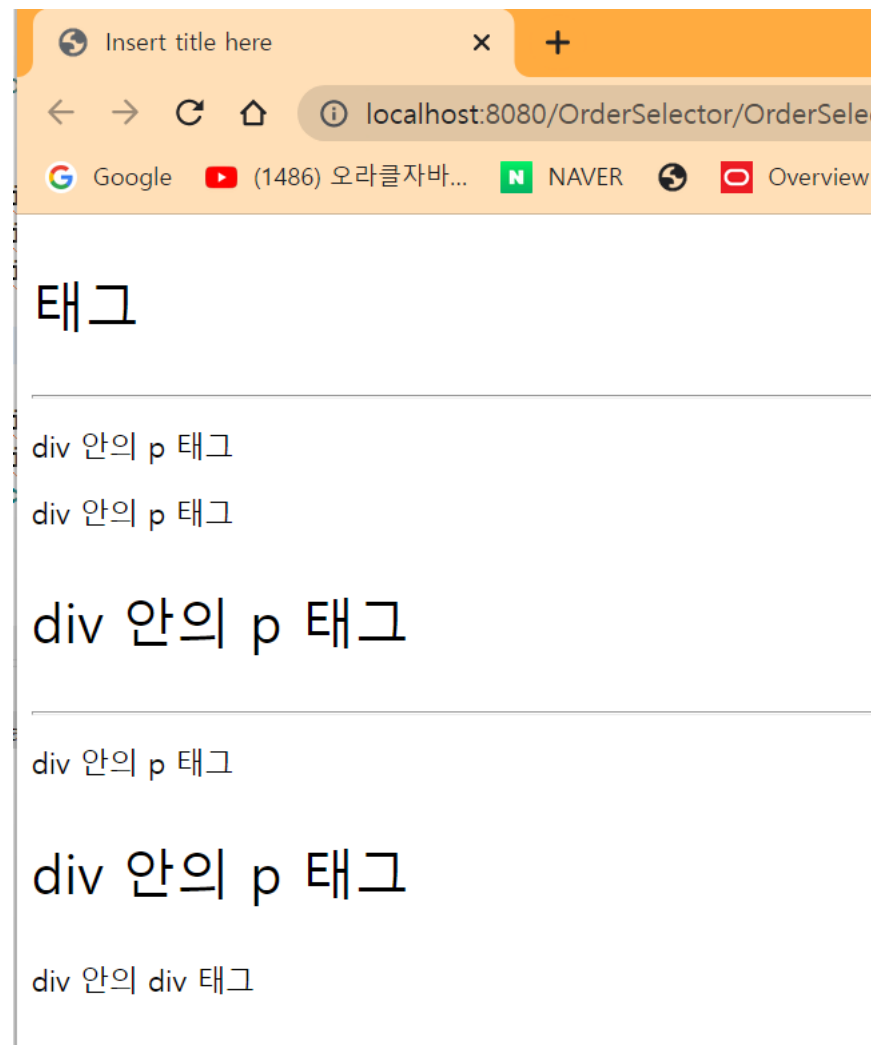
div 안의 p 태그

div 안의 p 태그

div 안의 div 태그

:last-of-type

```
<script>
  $(function() {
    $("p:last-of-type").css("font-size", "30px");
  });
</script>
```

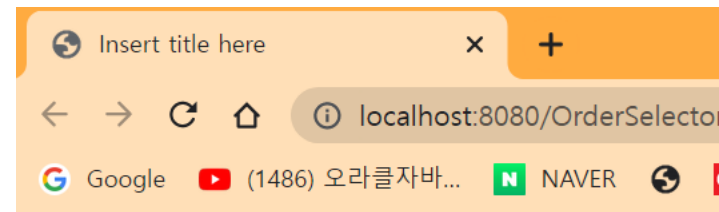


그 외 순서 필터 선택자

- :odd : 인덱스 번호(0부터 시작)가 홀수인 태그 선택
- :even : 인덱스 번호(1부터 시작)가 짝수인 태그 선택
- :only-child : 영역내에서 선택자에 해당하는 태그가 하나일 경우 선택. 다른 태그가 있으면 선택되지 않음
- :only-of-type : 영역내에 선택자에 해당하는 태그가 하나일 경우 선택

:odd

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



p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

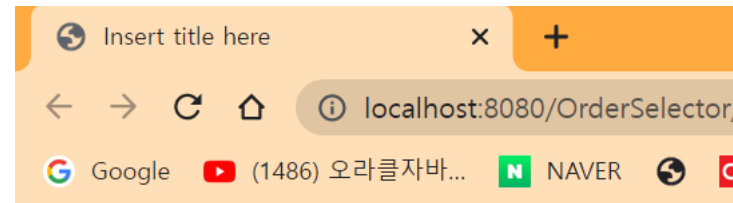
:even

```
$ (function() {  
    $ ("p:even") .css ("background-color", "red");  
});
```



:only-child

```
<script>
  $(function() {
    $("p:only-child").css("background-color", "yellow");
  });
</script>
```



p 태그

p 태그

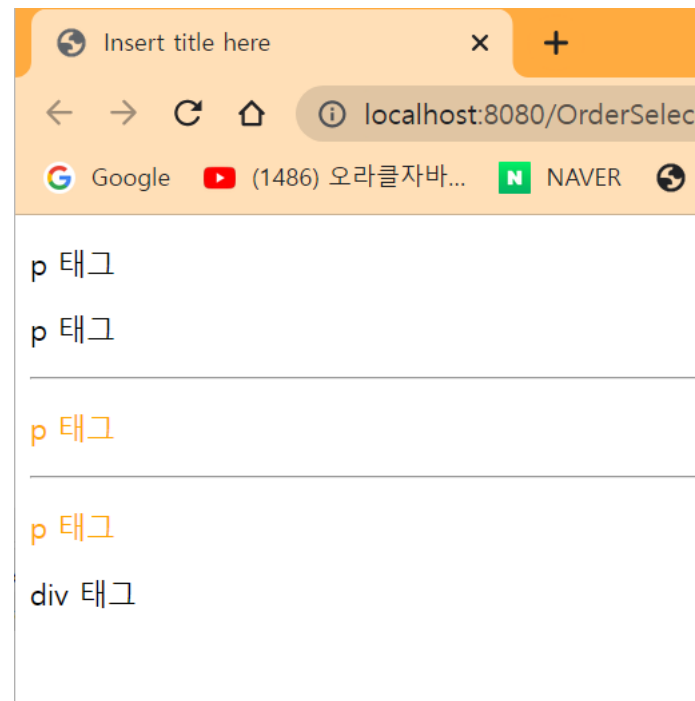
p 태그

p 태그

div 태그

:only-of-type

```
<script>
  $(function() {
    $("p:only-of-type").css("color", "orange");
  });
</script>
```



정리

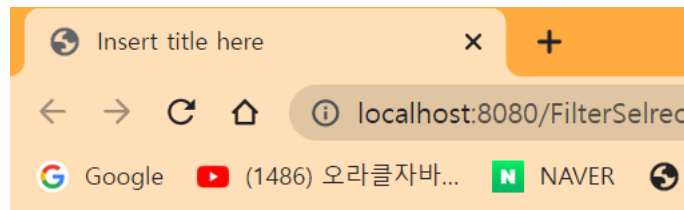
- 선택 필터 선택자를 이용하면 순서를 이용해 태그를 선택할 수 있다.

인덱스 번째 태그 선택

- :ep(인덱스) : 인덱스(0부터 시작)번째 해당하는 태그를 선택
- :nth-child(인덱스) : 각 영역의 인덱스(1부터 시작)번째 해당하는 태그를 선택. 인덱스번째 태그가 선택자에 해당하지 않으면 선택되지 않음
- :nth-of-type(인덱스) : 각 영역의 인덱스(1부터 시작)번째 해당하는 태그를 선택

실습

```
<script>
  $(function() {
    $("p:eq(3)") .css("background-color", "yellow");
    $("p:nth-child(1)") .css("color", "orange");
    $("p:nth-of-type(1)") .css("font-style", "italic");
  });
</script>
```



p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

div 태그

p 태그

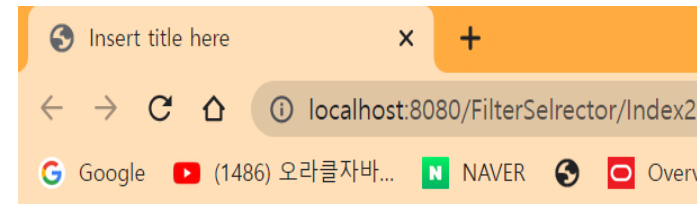
p 태그

인덱스 번째 태그 선택

- :gt(인덱스) : 선택자에 해당하는 태그 중 인덱스(0부터 시작)번째 태그 이후의 태그들이 선택
- :lt(인덱스) : 선택자에 해당하는 태그 중 인덱스(0부터 시작)번째 태그 이전의 태그들이 선택
- :nth-last-child(인덱스) : 뒤에서 인덱스(1부터 시작)번째 태그가 선택. 인덱스번째 태그가 선택자와 다르면 선택되지 않음
- :nth-last-of-type(인덱스) : 뒤에서 인덱스(1부터 시작)번째 태그가 선택

실습

```
<script>
  $(function() {
    $("p:gt(0)").css("background-color", "yellow");
    $("p:lt(7)").css("color", "orange");
    $("p:nth-last-child(1)").css("font-size", "30px");
    $("p:nth-last-of-type(1)").css("font-style", "italic");
  });
</script>
```



p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

p 태그

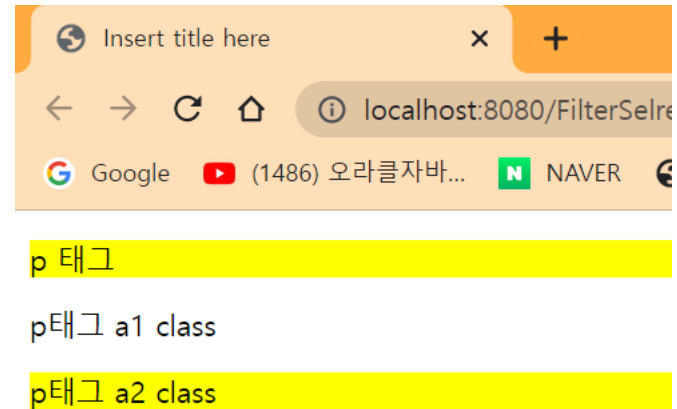
div 태그

:not

- :not : 선택자에 해당하지 않는 태그들이 선택

실습

```
<script>
  $(function() {
    $("p:not(.a1)").css("background-color", "yellow");
  });
</script>
```



정리

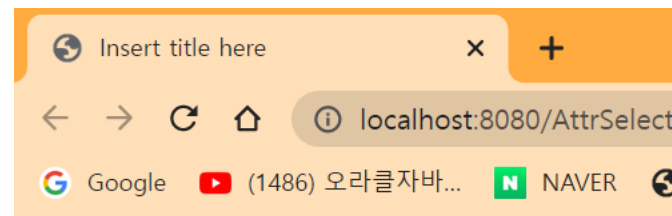
- 인덱스 필터 선택자를 이용하면 순서를 이용해 태그를 선택할 수 있다.

속성 선택자

- 태그에 설정되어 있는 속성을 통해 선택하는 선택자
- [속성] : 속성을 가지고 있는 태그를 선택
- [속성 = 값] : 속성의 값이 지정된 값과 일치하는 태그를 선택
- [속성 != 값] : 속성의 값이 지정된 값과 일치하지 않는 태그를 선택

실습

```
<script>
  $(function() {
    $("p[title]").css("background-color", "yellow");
    $("p[title = 'aaa1']").css("color", "orange");
    $("p[title != 'aaa1']").css("color", "red");
  });
</script>
```



속성이 없는 p태그

title이 aaa1인 p 태그

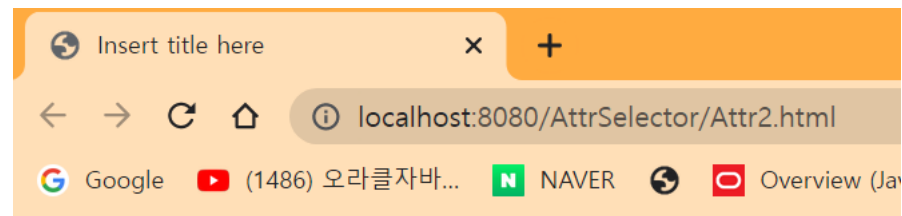
title이 aaa2인 p 태그

속성 선택자

- [속성 \$= 값] : 속성의 값이 지정된 값으로 끝나는 태그를 선택
- [속성 |= 값] : 속성의 값이 지정된 속성의 값과 일치하거나 지정된 속성으로 시작하고 하이픈으로 연결된 태그를 선택
- [속성 ^= 값] : 속성의 값이 지정된 값으로 시작하는 태그를 선택

실습

```
<script>
  $(function() {
    $("p[title $='a3']").css("background-color", "blue");
    $("p[title |= 'aaa3']").css("color", "orange");
    $("p[title ^= 'aaa3']").css("font-size", "30px");
  });
</script>
```



title이 aaa3인 p 태그

title이 aaa3-test인 p 태그

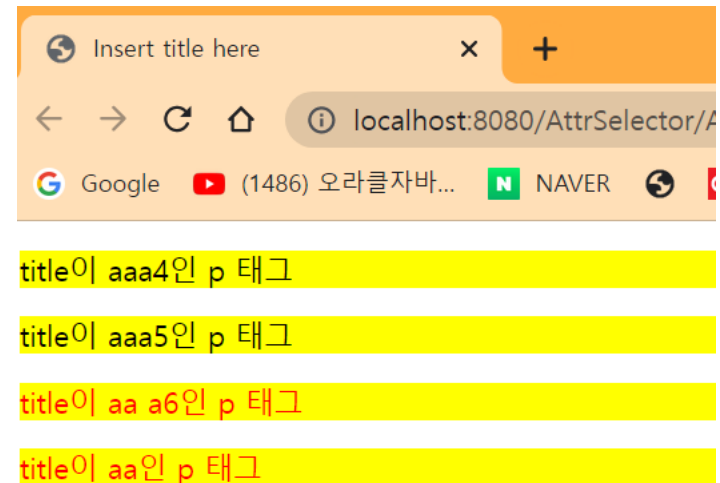
title이 aaa3test인 p 태그

속성 선택자

- [속성 ~= 값] : 속성의 값이 지정된 값을 단어로 가지고 있는 태그를 선택
- [속성 *= 값] : 속성의 값이 지정된 값을 포함하는 태그를 선택

실습

```
<script>
  $(function() {
    $("p[title ~= 'aa']").css("color", "red");
    $("p[title *= 'aa']").css("background-color", "yellow");
  });
</script>
```



정리

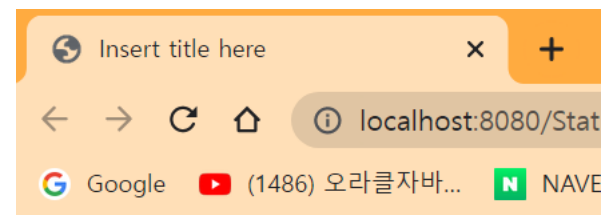
- 속성 선택자를 이용하면 태그에 설정되어 있는 속성을 이용해 태그를 선택할 수 있다.

상태 선택자

- 태그의 상태에 따라 선택하는 선택자
- :header : h1 ~ h6 태그를 선택
- :focus : 현재 포커스가 주어진 태그를 선택
- :contains('문자열') : 지정된 문자열이 포함되어 있는 태그를 선택
- :has(선택자) : 지정된 선택자를 포함하고 태그를 선택

실습

```
<script>
  $(function() {
    $("<strong>:header")</strong> .css("color", "red");
  });
</script>
```



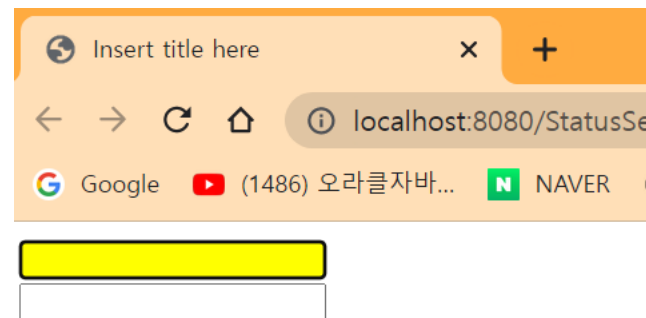
p 태그

h1 태그

h3 태그

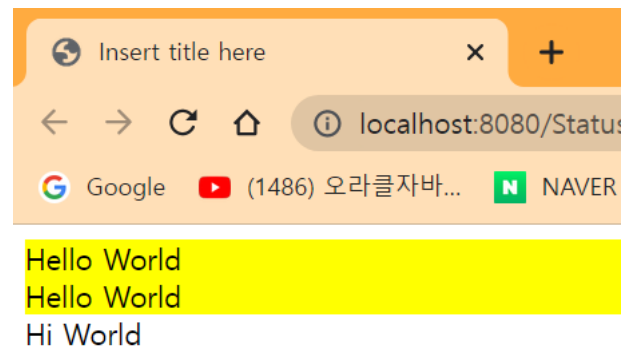
실습

```
<script>
  $(function() {
    $(":focus").css("background-color", "yellow");
  });
</script>
```



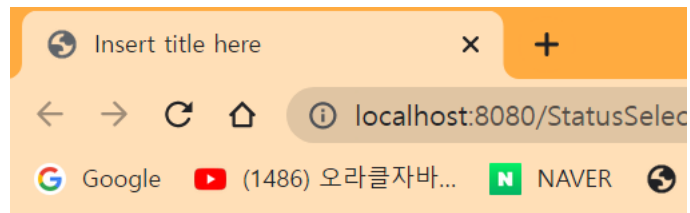
실습

```
<script>
  $(function() {
    $("div:contains('Hello')").css("background-color", "yellow");
  });
</script>
```



실습

```
<script>
  $(function() {
    $("div:has(p)").css("background-color", "yellow");
  });
</script>
```



p 태그를 가지고 있는 div 태그

div 태그를 가지고 있는 div 태그

정리

- 상태 선택자를 이용하면 태그의 상태에 따라 태그를 선택