

Chapter 3

CSS의 기초와 선택자



CSS의 기초



■ Style

- HTML 문서에 삽입된 텍스트, 이미지 등의 콘텐츠에 대해 각종 효과(예를 들어 색상, 모양 등)을 지정하는 것

■ CSS (Cascade Style Sheet)

- HTML 문서 내의 콘텐츠에 대해 적용할 효과를 미리 정의하고, HTML 문서의 일부분 또는 문서 전체에 적용하는 방법을 제공
- 기존 HTML의 기능을 보완
 - HTML만으로 할 수 없는 부분을 수행 가능





▣ CSS의 장점

- **HTML 기능의 확장**

- HTML만으로 수행할 수 없는 기능을 수행 가능
- [예]
 - : 글자의 크기를 픽셀로 지정하여 더욱 세밀한 문자나 더욱 큰 문자 표현이 가능
 - : 문자나 단어의 간격이나 문장의 간격을 지정 가능

- **통일된 문서 양식을 제공**

- 동일한 효과를 여러 부분에 일괄 지정 가능
 - : HTML에서는 동일한 효과라도 각각 지정해야 함
- 적용된 효과의 수정이 용이

- **모듈화된 문서 작성 가능**

- 디자인 부분과 내용 부분의 분리

- **문서 로딩 시간의 단축**

- 브라우저가 읽을 문자의 수가 적으므로 빠르 로딩 가능





▣ CSS의 기존 구조

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

    선택자1 {
        속성1: 값1;
        .....
        속성n : 값n;
    }
    .....

    선택자n {
        .....
    }

</style>
```

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

    body {
        text : black;
        textalign : center;
    }
    .....
</style>
```

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

    body {
        text : black;  textalign : center;
    }
    .....
</style>
```





■ 선택자(selector), 속성(attribute), 값(value)

- HTML의 이 부분(**selector**)에 대해 이런 효과(**attribute**)을 이렇게(**value**) 지정한다.

```
선택자 { 속성 : 값 }
```

- 선택자 : CSS 효과를 지정할 대상 (HTML 문서의 특정 부분)
- 속성 : 지정하고자 하는 효과의 종류
- 값 : 지정하고자 하는 효과의 내용

- 하나의 선택자는 하나 이상의 속성과 값을 가질 수 있음

- 각 속성과 값의 쌍은 세미콜론(;)으로 구분해야 함





■ HTML의 속성과 값 vs. CSS의 속성과 값

HTML의 속성과 값

- 속성과 값은 = 기호로 구분
- 값은 인용부호로 지정
- 속성과 값의 쌍은 빈 공간으로 구분
 - [예] color = "red"

HTML의 속성과 값

- 속성과 값은 콜론(:) 기호로 구분
- 값은 인용부호로 지정하지 않음
- 속성과 값의 쌍은 세미콜론(;)으로 구분
 - [예] color : red;
- 값이 두 개 이상인 경우 세미콜론으로 지정
 - [예] font-family : "HY견명조";





■ 인라인(Inline) 스타일 지정방식

- `style` 속성을 사용해 태그 내에 직접 지정하는 방식
- 특정 부분에만 CSS를 적용하고자 하는 경우 사용

■ 임베디드(Etbedded) 스타일 지정방식

- `<STYLE>` 태그를 사용해 지정하는 방식
- 현재 수행중인 문서 전체에 적용될 스타일을 지정하는 방식

■ 링크(link) 스타일 지정방식

- 문서 외부에 CSS 파일을 별도로 생성
 - 사용하고자 하는 문서에 링크를 연결하여 지정하는 방식
- 여러 문서에 동일한 스타일을 지정하고자 할 경우 사용



세 가지 방식은 혼용되어 사용됨





▣ 인라인(Inline) 스타일 지정방식

- **style 속성을 사용해 적용하고자 하는 태그 내에 직접 지정하는 방식**

```
<태그 style = "속성:값; 속성:값; ... ">
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>인라인 스타일 지정</title>
</head>

<body style="background:yellow">
    <h1 style="font-size:25pt">이 문자의 크기는 25pt입니다.</h1>
    <h2 style="color:blue">이 문자의 색깔은 blue입니다.</h2>
    <p style="margin-left:100px">이 문자의 왼쪽 여백은 100픽셀입니다.</p>
</body>

</html>
```





▣ 임베디드(Embedded) 스타일 지정방식

- <style> 태그 내에 선택자를 사용해 효과를 지정하는 방식
 - 현재 문서에 존재하는 동일한 요소들에 동일한 효과를 지정하는데 사용
- 일반적으로 <head>태그 내에 위치 함

```
<head>

    <style type="text/css">

        선택자 { 속성:값; 속성:값, ... }

        .....
        .....

    </style>

</head>
```





▣ 임베디드(Embedded) 스타일 지정방식의 예

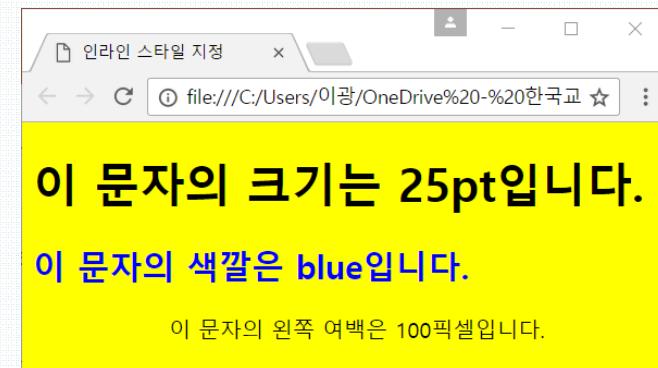
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>임베디드 스타일 지정</title>

    <style type="text/css">
        body {background:yellow }
        h1 { font-size:25pt; }
        h2 { color:blue; }
        p { margin-left:100px; }
    </style>

</head>

<body>
    <h1>이 문자의 크기는 25pt입니다.</h1>
    <h2>이 문자의 색깔은 blue입니다.</h2>
    <p>이 문자의 왼쪽 여백은 100픽셀입니다.</p>
</body>

</html>
```





▣ 링크(link) 스타일 지정방식

- CSS코드 만으로 구성된 새로운 문서를 만들고 문서에 연결하는 방식
 - CSS 파일의 확장자는 .css임
- CSS 파일을 문서에 연결하는 방법
 - <LINK> 태그를 사용

```
<LINK rel="stylesheet" type="text/css" href="CSS파일이름">
```





▣ 링크(link) 스타일 지정방식의 예

```
body {background:yellow }  
h1 { font-size:25pt; }  
h2 { color:blue; }  
p { margin-left:100px; }
```

mystyle.css

별도로 작성된 css 파일에서는 <style> 태그를 지정하지 않음

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>링크 스타일 지정</title>  
    <link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>  
  
<body>  
    <h1>이 문자의 크기는 25pt입니다.</h1>  
    <h2>이 문자의 색깔은 blue입니다.</h2>  
    <p>이 문자의 왼쪽 여백은 100픽셀입니다.</p>  
</body>  
  
</html>
```





■ 스타일 속성의 상속과 우선순위

• 스타일 속성의 상속

- 링크 스타일 -> 임베디드 스타일 -> 인라인 스타일

- : 링크 스타일에 적용되는 효과 임베디드나 인라인 스타일로 지정되지 않아도 문서에 적용됨
- : 링크 스타일에 적용되지 않았지만 임베디드에 적용된 스타일은 문서에 적용됨

• 스타일의 우선순위

- 인라인 스타일 < 임베디드 스타일 < 링크 스타일

- : 같은 속성이 서로 다른 값으로 인라인, 임베디드, 링크에 모두 적용된 경우, 인라인에 지정된 값으로 적용됨
- : 적용 대상에 가까운 스타일이 적용됨

• !important 속성

- !important 속성이 적용되 경우 우선순위를 무시하고 해당 속성이 적용됨

- : 같은 속성이 서로 다른 값으로 인라인, 임베디드, 링크에 모두 적용되고 인라인 속성에 !important 속성이 링크 스타일에 지정된 경우, 우선순위를 무시하고 링크 스타일에 지정된 값이 적용됨



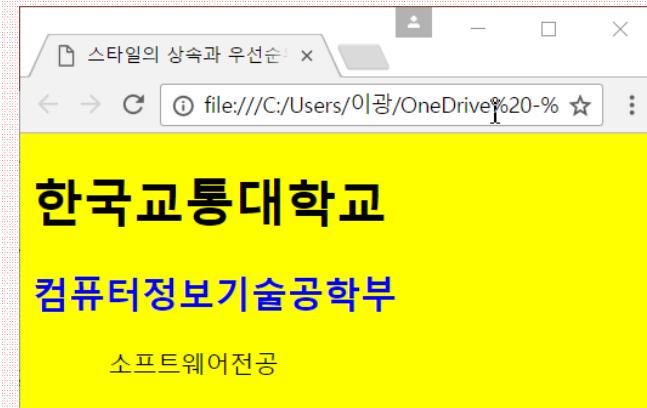


▣ 스타일 속성의 상속과 우선순위의 예

```
body {background:yellow !important; }
h1 { font-size:25pt; }
h2 { color:blue; }
p { margin-left:100px; }
```

mystyle2.css

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>스타일의 상속과 우선순위</title>
    <link rel="stylesheet" type="text/css" href="mystyle2.css">
    <style type="text/css">
        body {background:green }
        h1 { font-size:25pt; }
        h2 { color:blue; }
    </style>
</head>
<body>
    <h1 style="font-size:20pt">한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p style="margin-left:50px">소프트웨어전공</p>
</body>
</html>
```





CSS의 선택자



■ 선택자(selector)

- CSS를 사용해 효과를 적용하고자 하는 대상을 선택하는 역할 수행

■ 선택자의 종류

• 단순선택자

- 하나의 요소를 사용해 선택자를 구성
 - : [예] 전체 선택자, 요소 선택자, 클래스 선택자 등

• 조합 선택자

- 하나 이상의 요소의 조합으로 선택자를 구성
 - : [예] 그룹 조합 선택자, 자식 조합 선택자 등

• 가상 클래스 선택자

- 가상 클래스에 사용되는 선택자
 - : [예] :link, vlink, alink 등



단순 선택자



■ 단순 선택자의 종류

| 종류 | 표현 | 설명 |
|------------|----------|--------------------------|
| 전체 선택자 | * | 문서의 모든 내용에 스타일을 지정 |
| 태그(형식) 선택자 | 요소 | 지정한 태그에 스타일을 지정 |
| 클래스 선택자 | (요소).선택자 | 클래스를 정의하여 스타일을 지정 |
| 아이디 선택자 | (요소)#선택자 | 아이디를 지정하여 스타일을 지정 |
| 속성 선택자 | 요소[속성] | 요소가 가지는 속성 값에 따라 스타일을 지정 |





▣ 전체 선택자

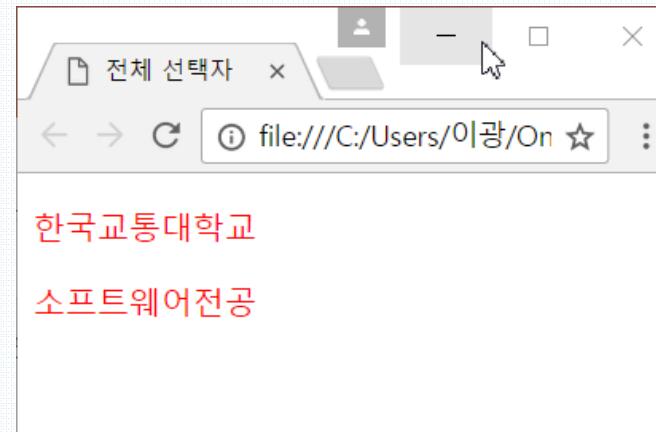
- 현재 문서의 모든 요소를 선택하기 위해 사용되는 선택자

```
* { 속성1=값1; ... 속성n=값n }
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>전체 선택자</title>

    <style type="text/css">
        * { color:red; font-size:12pt; }
    </style>

</head>
<body>
    <p>한국교통대학교</p>
    <p>소프트웨어전공</p>
</body>
</html>
```





▣ 태그(형식) 선택자

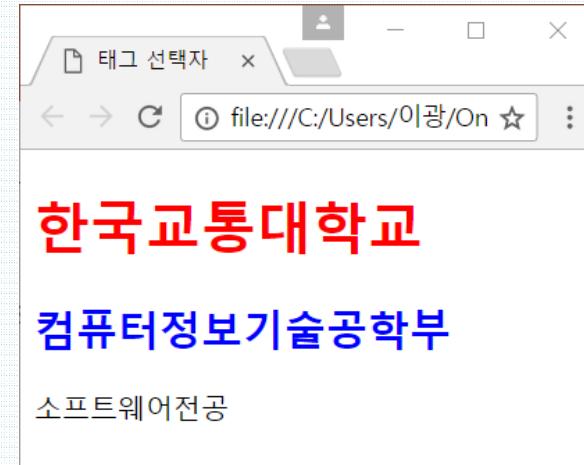
- CSS를 적용할 대상으로 HTML 요소(태그)를 사용하는 경우

```
태그 { 속성1=값1; ... 속성n=값n }
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>

    <style type="text/css">
        h1 { color:red; }
        h2 { color:blue; }
    </style>

</head>
<body>
    <h1>한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p>소프트웨어전공</p>
</body>
</html>
```





▣ 클래스(class) 선택자 / 아이디(id) 선택자

- 태그 이름을 사용하지 않고 임의의 이름을 선택자로 사용
- 하나의 스타일을 지정하고 여러 요소에 지정하기 위해 사용

▣ 태그 선택자와의 차이점

- **태그 선택자**
 - 지정된 태그에만 효과를 지정할 수 있음
- **클래스/ 아이디 선택자**
 - 여러 종류의 태그에 동일한 효과를 지정 가능





▣ 클래스 선택자와 아이디 선택자의 차이

- **클래스 선택자**

- 하나의 효과를 여러 태그에 중복하여 지정 가능
 - : 하나의 클래스 선택자를 여러 요소에 작용 가능

- **아이디 선택자**

- 가능 면에서는 클래스 선택자와 유사
- 하나의 문서에 한번만 사용 가능
 - : HTML 문서에서는 아이디 선택자를 여러 번 사용 가능
 - : 서버 프로그램에서 아이디 선택자를 중복 사용할 경우 오류나 경고를 출력함





▣ 클래스 선택자의 생성과 적용

- **클래스 선택자의 생성**

- 클래스 선택자의 이름은 임의로 사용 가능 (예약어는 사용 불가)
- 클래스의 이름은 도트문자(.)으로 시작해야 함

```
<style>  
    .클래스 이름 {속성1=값1; ... 속성n=값n }  
</style>
```

- **클래스 선택자의 적용**

- 적용할 태그 내에서 class 속성을 사용해 적용
 - : : 클래스의 이름을 class 속성의 값을 지정함

```
<태그 class = 클래스 이름 >
```



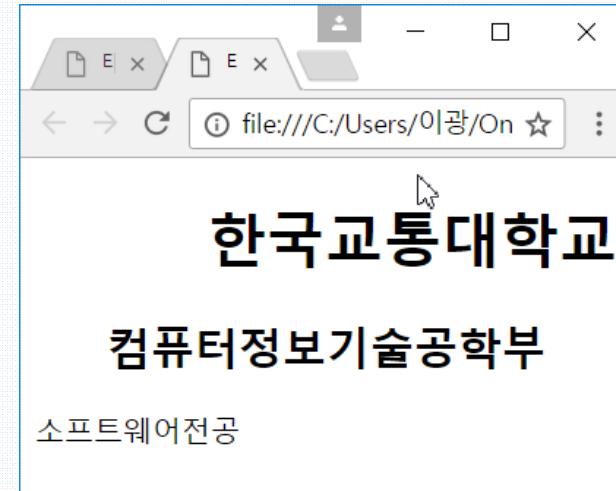


▣ 클래스 선택자의 생성과 적용 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>

    <style type="text/css">
        .right { text-align:right; }
        .center { text-align:center; }
    </style>

</head>
<body>
    <h1 class=right>한국교통대학교</h1>
    <h2 class=center>컴퓨터정보기술공학부</h2>
    <p>소프트웨어전공</p>
</body>
</html>
```





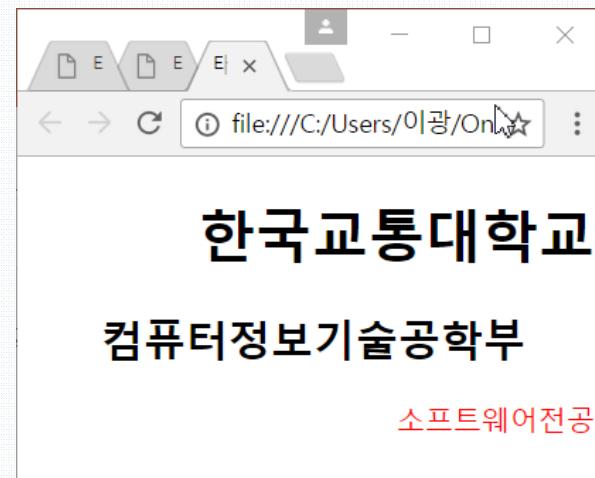
▣ 클래스 선택자의 생성과 적용 예

- 클래스 선택자는 하나의 태그 내에서 중복해 사용 가능

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>

    <style type="text/css">
        .right { text-align:right; }
        .center { text-align:center; }
        .color { color:red }
    </style>

</head>
<body>
    <h1 class=right>한국교통대학교</h1>
    <h2 class=center>컴퓨터정보기술공학부</h2>
    <p class="right color">소프트웨어전공</p>
</body>
</html>
```





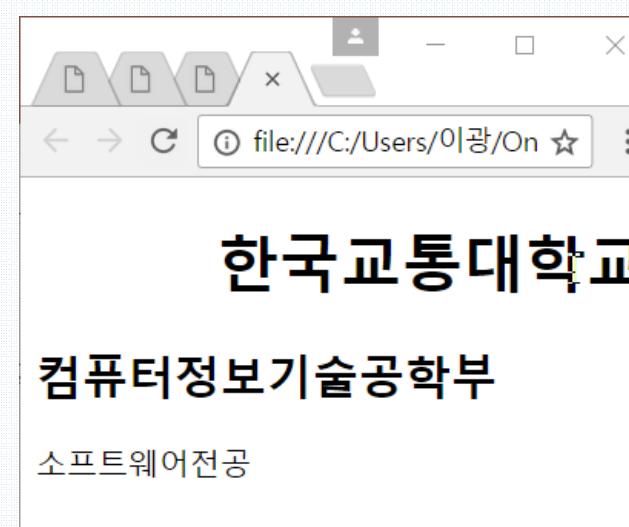
▣ 클래스 선택자는 태그 선택자와 함께 사용될 수 있음

- 클래스 선택자가 태그 선택자와 함께 사용될 경우, 클래스 선택자는 해당 태그에만 적용될 수 있음

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>

    <style type="text/css">
        h1.right { text-align:right; }
        div.center { text-align:center; }
    </style>

</head>
<body>
    <h1 class=right>한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p class=center>소프트웨어전공</p>
</body>
</html>
```





▣ 아이디 선택자의 생성과 적용

- **아이디 선택자의 생성**

- 아이디 선택자의 이름은 임의로 사용 가능 (예약어는 사용 불가)
- 아이디의 이름은 #으로 시작해야 함

```
<style>  
    #아이디 이름 {속성1=값1; ... 속성n=값n }  
</style>
```

- **아이디 선택자의 적용**

- 적용할 태그 내에서 id 속성을 사용해 적용
: 아이디의 이름을 id 속성의 값을 지정함

```
<태그 id = 아이디 이름 >
```



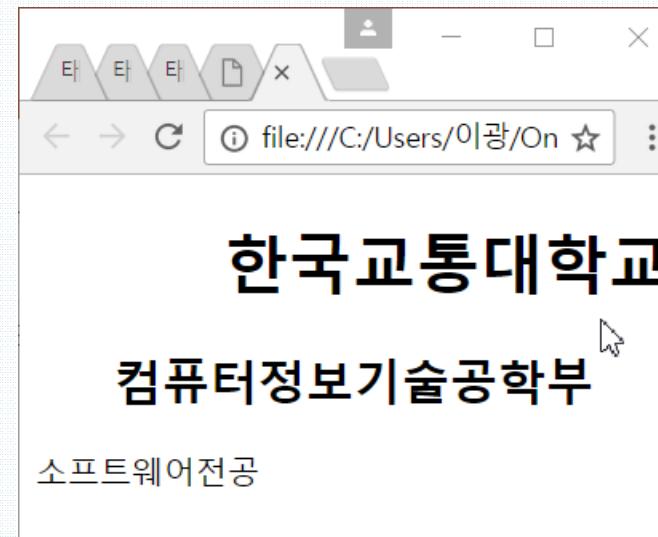


▣ 아이디 선택자의 생성과 적용 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>아이디 선택자</title>

    <style type="text/css">
        #right { text-align:right; }
        #center { text-align:center; }
    </style>

</head>
<body>
    <h1 id=right>한국교통대학교</h1>
    <h2 id=center>컴퓨터정보기술공학부</h2>
    <p>소프트웨어전공</p>
</body>
</html>
```





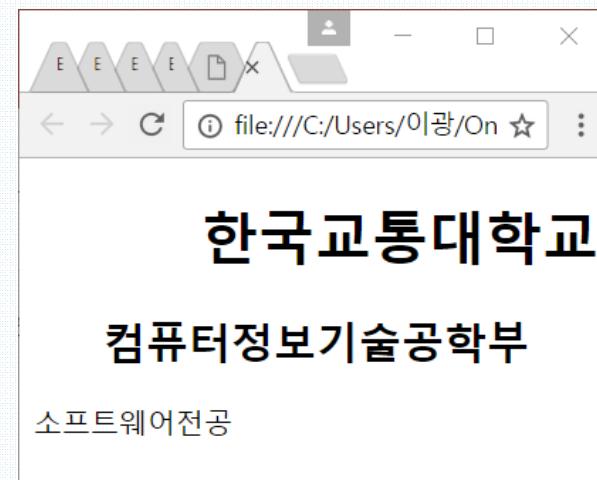
▣ 아이디 선택자의 생성과 적용 예

- 아이디 선택자는 하나의 태그 내에서 중복해 사용할 수 없음

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>

    <style type="text/css">
        #right { text-align:right; }
        #center { text-align:center; }
        #color { color:red; }
    </style>

</head>
<body>
    <h1 id=right>한국교통대학교</h1>
    <h2 id=center>컴퓨터정보기술공학부</h2>
    <p id="right color">소프트웨어전공</p>
</body>
</html>
```





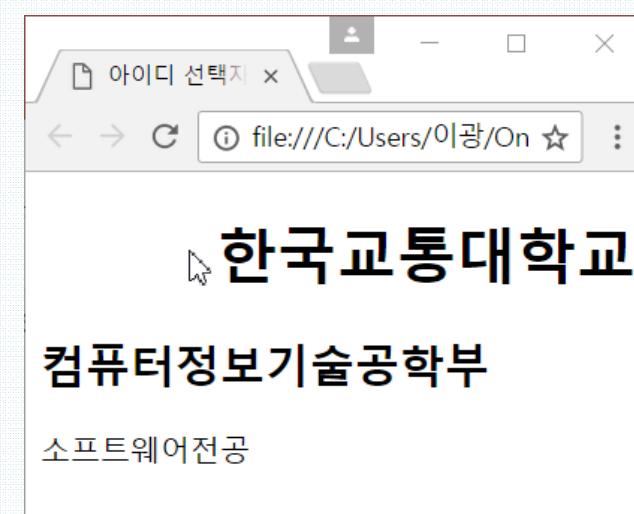
▣ 아이디 선택자는 태그 선택자와 함께 사용될 수 있음

- 아이디 선택자가 태그 선택자와 함께 사용될 경우, 아이디 선택자는 해당 태그에만 적용될 수 있음

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>아이디 선택자</title>

    <style type="text/css">
        h1#right { text-align:right; }
        div#center { text-align:center; }
    </style>

</head>
<body>
    <h1 id=right>한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p id=center>소프트웨어전공</p>
</body>
</html>
```





▣ 속성(attribute) 선택자

- 특정 요소가 가지는 속성과 값을 사용해 선택자를 지정

| 선택자 | 설명 |
|--------------|--|
| 요소[속성] | 속성 값에 관계없이 해당 속성을 사용하는 요소 |
| 요소[속성 = 값] | 속성 값이 정확히 일치하는 요소 |
| 요소[속성 ~= 값] | 속성이 지정된 값을 가지는 요소 (값은 공백문자로 구분되는 단어임) |
| 요소[속성 = 값] | 속성 값이 정확히 값과 일치하는 요소 또는 속성 값이 지정된 값으로 시작하면서 바로 뒤에 “_” 기호가 나타난 요소 |
| 요소[속성 ^= 값] | 속성 값의 시작 부분의 문자와 일치하는 요소 |
| 요소[속성 \$= 값] | 속성 값의 끝 부분의 문자와 일치하는 요소 |
| 요소[속성 *= 값] | 속성 값 전체 중 어떤 일부분이라도 일치하는 요소 |





▣ 속성(attribute) 선택자의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>속성 선택자</title>
    <style>
        a { line-height: 25pt; text-decoration: none; padding-left: 70px; background-repeat: no-repeat; }
        p[title="bookmark"] { font-weight: bold; font-size: 12pt; text-decoration: underline; }
        a[title=전자신문] { font-weight: bold; background-image: url(images/site_etnews.jpg); }
        a[title~=신문] { color: red; background-image: url(images/site_haninews.jpg); }
        a[title^=경향] { font-weight: bold; font-style: italic; background-image: url(images/site_kyungnews.jpg); }
        a[title$=뉴스] { color: blue; background-image: url(images/site_ohmynews.jpg); }
        a[title|=구글] { color: green; background-image: url(images/site_google.jpg); }
        a[title*=투] { font-weight: bold; background-image: url(images/site_youtube.jpg); }
    </style>
</head>
<body>
    <h3>CSS3 속성 선택자 연습하기 </h3>
    <p title="bookmark">즐겨찾기</p>
    <a title="전자신문" href="http://www.etnews.com">전자신문</a><br>
    <a title="한겨레 신문" href="http://www.hani.co.kr/">한겨레 신문</a><br>
    <a title="경향신문" href="http://www.khan.co.kr">경향신문</a><br>
    <a title="오마이뉴스" href="http://www.ohmynews.com">오마이뉴스</a><br>
    <a title="구글-검색" href="https://www.google.co.kr/">구글 검색</a><br>
    <a title="유튜브" href="http://www.youtube.com/?gl=KR&hl=ko">유튜브</a><br>
</body>
</html>
```



조합 선택자



▣ 조합 선택자의 종류

| 조합선택자 | 표현 | 설명 |
|-----------|-------------|-------------------------------|
| 하위 조합 선택자 | 선택자1 선택자2 | 선택자2가 선택자1에 반드시 포함되어 있을 경우 선택 |
| 자식 조합자 | 선택자1 > 선택자2 | 부모 선택자1의 직계 자식 선택자2를 선택 |
| 인접 형제 조합자 | 선택자1 + 선택자2 | 선택자1 바로 다음에 위치한 선택자2를 선택 |
| 일반 형제 조합자 | 선택자1 ~ 선택자2 | 선택자1 뒤에 인접하여 나타나는 모든 선택자2를 선택 |
| 그룹 조합자 | 선택자1, 선택자2 | 선택자1과 선택자2 모두 선택 |





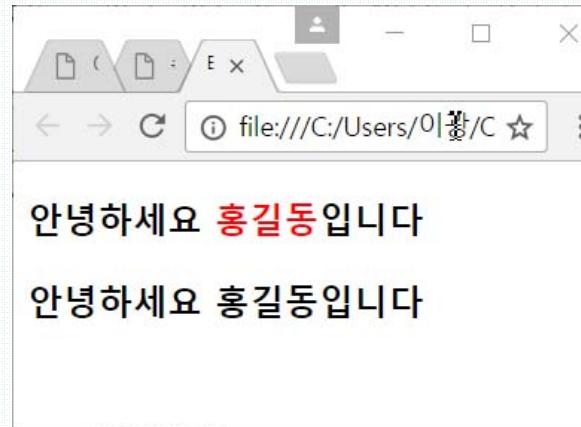
▣ 하위 조합 선택자

- 자손(descendent) 선택자 라고도 함

선택자1(부모) 선택자2(자식) { 속성1=값1; ... 속성n=값n }

- 선택자2(자식)이 선택자1(부모)에 포함되어 있을 경우, 선택자2에 CSS를 적용

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>
    <style type="text/css">
        h3 b { color:red; }
    </style>
</head>
<body>
    <h3>안녕하세요 <b>홍길동</b>입니다</h3>
    <h3>안녕하세요 홍길동입니다</h3>
</body>
</html>
```





▣ 그룹(group) 조합 선택자

- 동시에 지정하고자 하는 모든 요소를 콤마(,)로 구분하여 표현
 - 콤마로 구분된 모든 요소에 동일한 효과를 지정
 - 다중(multiple) 선택자라고도 함

선택자1, 선택자2 { 속성1=값1; ... 속성n=값n }

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>
    <style type="text/css">
        h1, h2, p { color:red; }
    </style>
</head>
<body>
    <h1>한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p>소프트웨어전공</p>
</body>
</html>
```





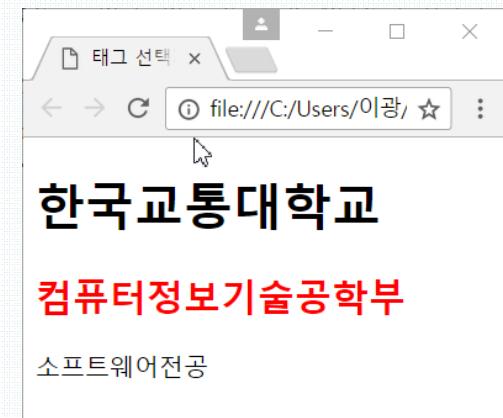
▣ 인접 형제 조합 선택자

- 특정 선택자를 기준으로 다음 선택자를 선택하는 선택자

```
선택자1 + 선택자2 { 속성1=값1; ... 속성n=값n }
```

- 선택자1 바로 다음에 위치한 선택자2를 선택

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>태그 선택자</title>
    <style type="text/css">
        h1 + h2 { color:red; }
    </style>
</head>
<body>
    <h1>한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p>소프트웨어전공</p>
</body>
</html>
```



인접한 요소가 h2가 아닐 경우 어떤 요소도 선택되지 않음





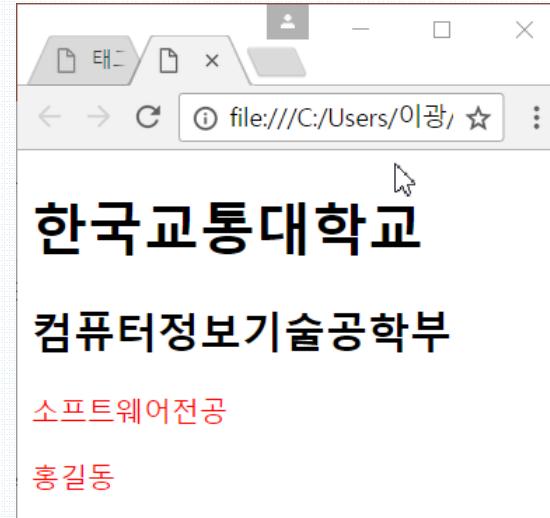
▣ 일반 형제 조합 선택자

- 특정 선택자를 기준으로 다음에 나오는 모든 선택자를 선택하는 선택자

```
선택자1 ~ 선택자2 { 속성1=값1; ... 속성n=값n }
```

- 선택자1 바로 다음에 위치한 모든 선택자2를 선택

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>일반 인접 선택자</title>
    <style type="text/css">
        h1 ~ p { color:red; }
    </style>
</head>
<body>
    <h1>한국교통대학교</h1>
    <h2>컴퓨터정보기술공학부</h2>
    <p>소프트웨어전공</p>
    <p>홍길동</p>
</body>
</html>
```





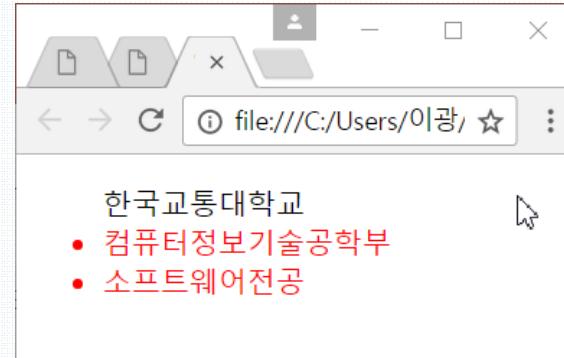
▣ 자식 조합 선택자

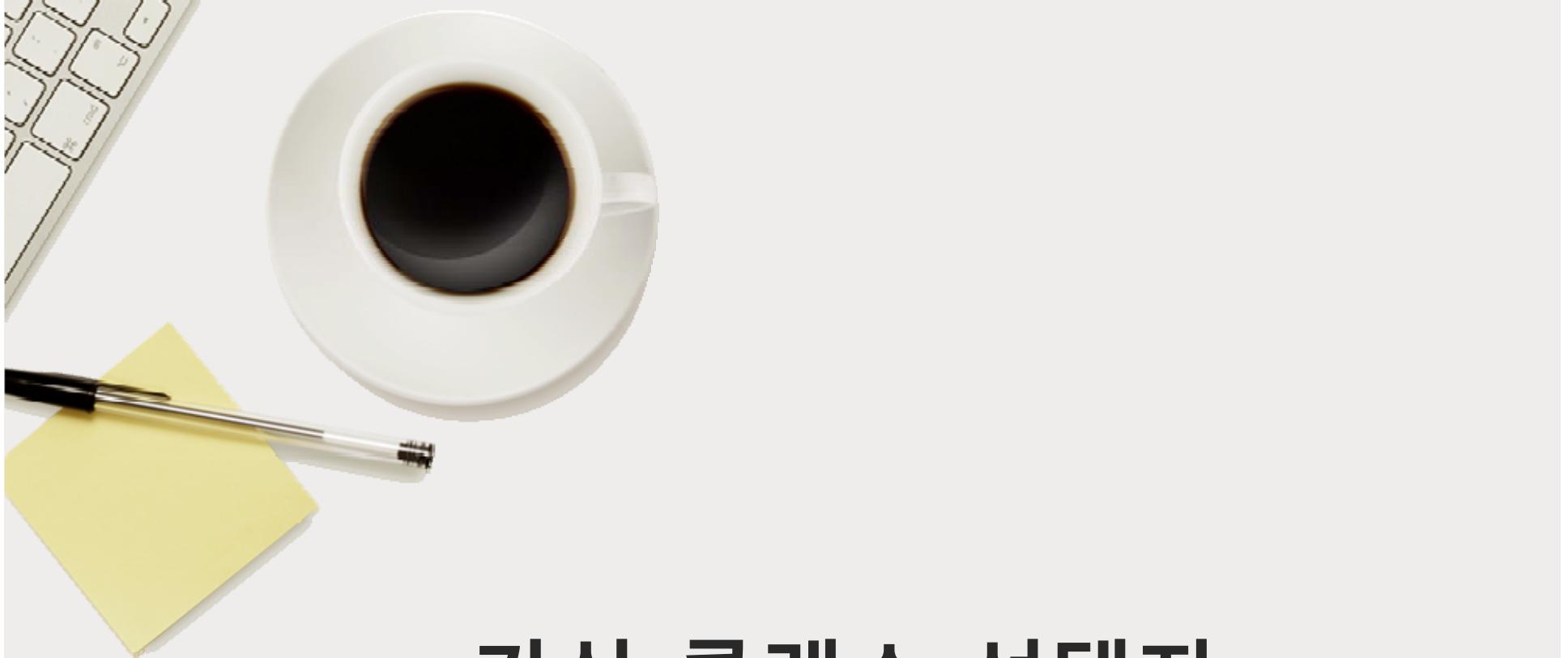
- 특정 요소의 직계 자식 요소를 선택하는 선택자

선택자1(부모) > 선택자2(자식) { 속성1=값1; ... 속성n=값n }

- 선택자1(부모)의 직계자식인 선택자2(자식)을 선택

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>일반 인접 선택자</title>
    <style type="text/css">
        ul > li { color:red; }
    </style>
</head>
<body>
    <ul>한국교통대학교
        <li>컴퓨터정보기술공학부
        <li>소프트웨어전공
    </ul>
</body>
</html>
```





가상 클래스 선택자



▣ 가상 클래스(pseudo class)

- 웹 문서의 소스(HTML)에는 실제로 존재하지 않지만 필요에 의해 임의로 가상의 선택자(selector)를 지정해서 특별한 상태를 설정하는 것
 - 클래스 선택자는 아니지만 엘리먼트들의 상태에 따라서 마치 클래스 선택자처럼 여러 엘리먼트를 선택할 수 있음
 - 한 개의 콜론(:) 기호를 사용
 - : [예] :link, :visited 등

▣ 가상 요소(pseudo element)

- 가상 클래스(pseudo class)처럼 선택자(selector)에 추가되지만 특별한 상태를 기술하는 대신, 문서의 특정 부분의 스타일을 지정
 - 두 개의 콜론(::) 기호를 사용
 - : [예] ::first-line, ::first-letter 등





■ 동적 가상 클래스

- 하이퍼링크 관련해 상태에 따른 효과를 지정하기 위한 가상 클래스
- 동적 가상 클래스 선택자

| 선택자 | 설명 |
|---------|---|
| :link | 아직 방문한 적이 없는 링크에 적용 |
| :vlink | 방문한 적이 있는 링크에 적용 |
| :active | 사용자에 의하여 요소가 활성화되는 동안 적용 |
| :hover | 하이퍼링크나 지정된 요소 위에 마우스를 올려 놓을 경우 적용 |
| :focus | 특정 요소에 초점이 있는 동안(예를 들면, 텍스트 입력 양식에 입력할 때, 텍스트 필드 안에 마우스 커서를 놓기 위해 Tab키를 눌렀을 때, 또는 기타 입력 형태를 수용할 때 등) 적용 |



동적(Dynamic) 가상 클래스



▣ 동적 가상 클래스의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>동적 가상 클래스</title>
    <style>
        p { font-size: 14px }
        a:link { color: green; }
        a:visited { color: blue; }
        a:hover { color: red; font-size: 18px }
        a:active { background: yellow }
        a:focus { background: yellow; color: black; font-weight: bold }
        a:focus:hover { background: blue; color: white }
    </style>
</head>
<body>
    <h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
    <p>즐겨찾기 1</p>
    <a href="http://www.etnews.com">전자신문</a> | <a href="http://www.hani.co.kr/">한겨레 신문</a> |
    <a href="http://www.khan.co.kr">경향신문</a> | <a href="http://www.ohmynews.com">오마이뉴스</a> |
    <a href="http://www.youtube.com/?gl=KR&hl=ko">유튜브</a>
</body>
</html>
```





▣ 목적지 가상 클래스

- 문서 내 이동 하이퍼링크 생성 시 앵커(name 속성으로 지정된 이동위치)에 효과를 지정할 때 사용되는 가상 클래스
- 목적지 가상 클래스 선택자

| 선택자 | 설명 |
|---------|--------------------------------|
| :target | 앵커(name 속성으로 지정된 이동위치)에 효과를 지정 |



목적지(target) 가상 클래스



▣ 목적지 가상 클래스

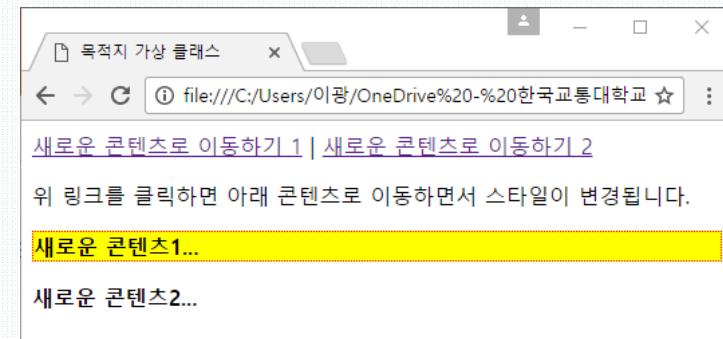
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>HTML5 스타일시트 연습</title>
    <style>
        p:target { background: yellow; border: 1px dotted red }
    </style>
</head>
<body>
    <a href="#sample1">새로운 콘텐츠로 이동하기 1</a> |
    <a href="#sample2">새로운 콘텐츠로 이동하기 2</a>

    <p>위 링크를 클릭하면 아래 콘텐츠로 이동하면서 스타일이 변경됩니다.</p>

```

```
<p id="sample1"><b>새로운 콘텐츠1...</b></p>
<p id="sample2"><b>새로운 콘텐츠2...</b></p>
```

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





■ 언어 가상 클래스

- 현재 문서에 나타나 있는 내용 중 여러 언어가 사용될 경우 언어마다 효과를 지정하기 위해 사용되거나, 언어에 따른 문자의 출력 방향을 지정하기 위한 가상 클래스
- 언어 가상 클래스의 종류

| 선택자 | 설명 |
|--------|--|
| :lang | 지정된 요소 중 지정된 언어로 시작되는 요소를 선택 |
| :focus | 언어의 출력 방향에 일치하는 요소를 선택 <ul style="list-style-type: none">- dir(ltr) : 왼쪽에서 오른쪽으로의 방향성을 가지는 요소 선택- dir(rtl) : 오른쪽에서 왼쪽으로의 방향성을 가지는 요소 선택 |

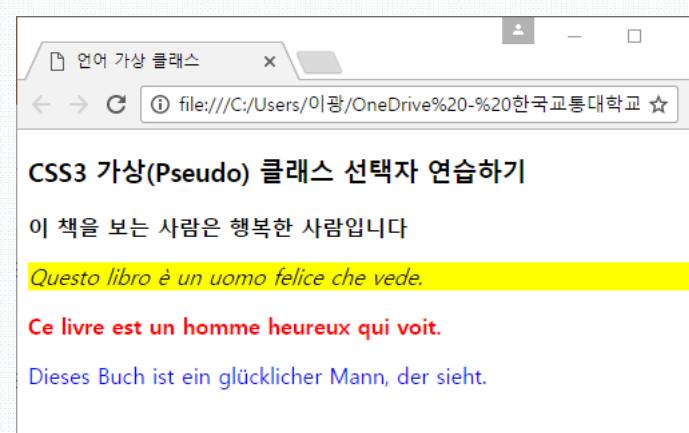




□ 언어 가상 클래스의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>언어 가상 클래스</title>
    <style>
        p:lang(ko) { font-weight:bold; }
        p:lang(it) { background:yellow; font-style: italic; }
        p:lang(fr) { color: #f00; font-weight:bold; }
        p:lang(de) { color: blue; }
    </style>
</head>

<body>
    <h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
    <p lang="ko">이 책을 보는 사람은 행복한 사람입니다</p>
    <p lang="it">Questo libro è un uomo felice che vede.</p>
    <p lang="fr">Ce livre est un homme heureux qui voit.</p>
    <p lang="de">Dieses Buch ist ein glücklicher Mann, der sieht.</p>
</body>
</html>
```





■ 사용자 인터페이스 가상 클래스

- 요소의 사용 가능 불가능 여부와 체크박스나 라디오버튼의 checked와 unchecked에 따른 효과를 지정하기 위한 가상 클래스
- 사용자 인터페이스 가상 클래스의 종류

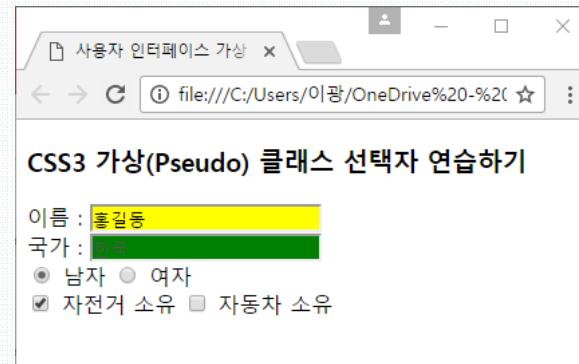
| 선택자 | 설명 |
|-----------|---|
| :enabled | 지정된 요소 중에서 활성화된 상태인 요소를 선택 |
| :disabled | 지정된 요소 중에서 활성화가 해제된 상태인 요소를 선택 |
| :checked | radio button이나 checkbox button에서 체크된 요소를 선택 |





■ 사용자 인터페이스 가상 클래스의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>사용자 인터페이스 가상 클래스</title>
    <style>
        input[type="text"]:enabled { background: yellow }
        input[type="text"]:disabled { background: green }
        input:checked { background: red }
    </style>
</head>
<body>
    <h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
    <form action="">
        이름 : <input type="text" value="홍길동" /><br>
        국가 : <input type="text" value="한국" disabled /><br>
        <input type="radio" value="male" name="gender" checked/> 남자
        <input type="radio" value="female" name="gender" /> 여자<br>
        <input type="checkbox" value="Bike" checked/> 자전거 소유
        <input type="checkbox" value="Car" /> 자동차 소유
    </form>
</body>
</html>
```



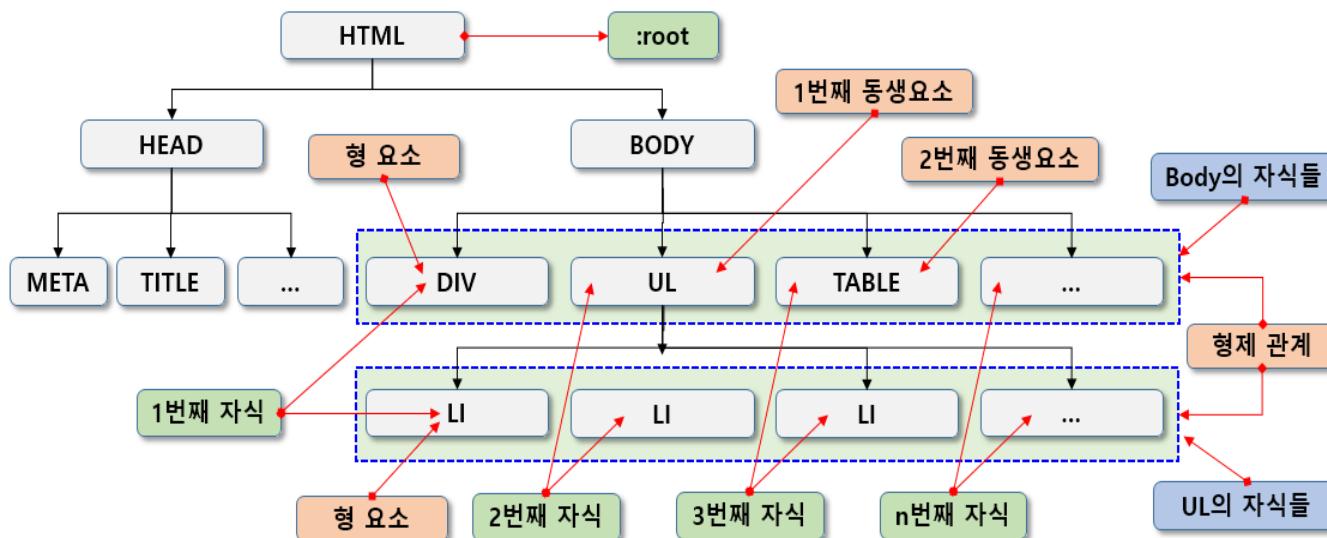
현재 :checked 선택자는 정상적으로 지원되지 않음





▣ 트리 구조 가상 클래스

- DOM(Document Object Model)을 형성하는 HTML 문서에 대해 각각의 노드에 접근해 선택하기 위한 가상 클래스
- 트리 구조의 예





- 트리 구조 가상 클래스 선택자의 종류

| 선택자 | 설명 |
|-------------------------|---------------------------------------|
| :root | 웹 문서 안의 최상위 요소(루트)를 선택 (문서 전체 선택) |
| :nth-child(an+b) | 동일한 부모를 가진 요소에 대해 처음부터 n번째의 요소를 선택 |
| :nth-last-child(an+b) | 동일한 부모를 가진 요소에 대해 마지막부터 n번째의 요소를 선택 |
| :first-child | 동일한 부모를 가진 요소 중 첫번째 요소를 선택 |
| :last-child | 동일한 부모를 가진 요소 중 마지막 요소를 선택 |
| :nth-of-type(an+b) | 동일한 부모를 가지는 요소 중 처음부터 n번째의 지정한 요소 선택 |
| :nth-of-last-type(an+b) | 동일한 부모를 가지는 요소 중 마지막부터 n번째의 지정한 요소 선택 |
| :first-of-type | 동일한 부모를 가지는 요소 중 첫번째 지정한 요소 선택 |
| :last-of-type | 동일한 부모를 가지는 요소 중 마지막 지정한 요소 선택 |
| | |
| | |
| | |





■ root 가상 선택자

- 현재 문서에서 최상위 요소인 <HTML> 태그를 선택
 - 문서 전체 선택을 위해 사용

■ nth-child(an+b) 가상 선택자

- 동일한 부모 요소 안에서 n번째에 있는 자식 요소를 선택
 - a, b는 반드시 정수여야 하며, 모두 0일 경우 어떤 요소도 선택하지 않음
 - [예]
 - : li:nth-child(2n+1) – 홀수 번째 들을 선택
 - : li:nth-child(2n) – 짝수 번째 들을 선택

■ nth-last-child(an+b)

- 동일한 부모 요소 중 끝에서부터 n번째에 있는 자식 요소를 선택
 - [예]
 - : li:nth-child(2) – 마지막에서 두 번째 를 선택

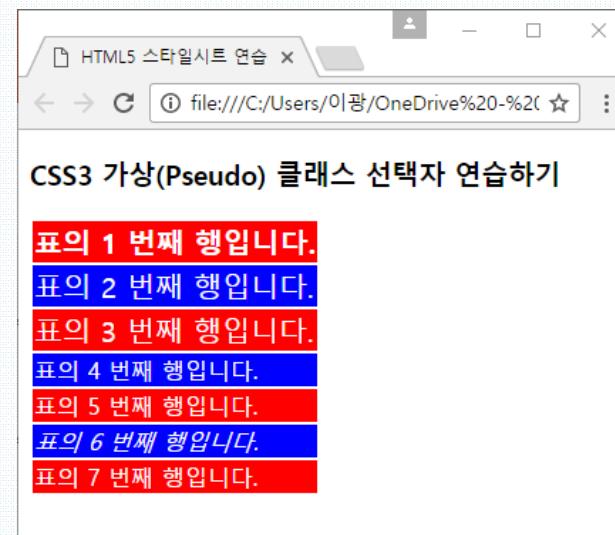




■ nth-child(an+b), nth-last-child(an+b)의 예

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>HTML5 스타일시트 연습</title>
<style>
tr:nth-child(1) { font-weight: bold; }
tr:nth-last-child(2) { font-style: italic; }
tr:nth-child(2n+1) { background: red; color:white; }
tr:nth-child(even) { background: blue; color:white; }
tr:nth-child(-n+3) { font-size: 20px; }
</style>

</head>
<body>
<h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
<table>
<tr><td>표의 1 번째 행입니다.</td></tr>
<tr><td>표의 2 번째 행입니다.</td></tr>
<tr><td>표의 3 번째 행입니다.</td></tr>
<tr><td>표의 4 번째 행입니다.</td></tr>
<tr><td>표의 5 번째 행입니다.</td></tr>
<tr><td>표의 6 번째 행입니다.</td></tr>
<tr><td>표의 7 번째 행입니다.</td></tr>
</table>
</body>
</html>
```





▣ :first-child, :last-child 가상 클래스 선택자

- **:first-child 가상 클래스 선택자**
 - 동일한 부모의 요소 중 첫번째 요소를 선택
 - :nth-child(1)과 동일

- **:first-child 가상 클래스 선택자**
 - 동일한 부모의 요소 중 마지막 요소를 선택
 - :nth-last-child(1)과 동일





▣ :first-child, :last-child 가상 클래스 선택자의 예

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>:first-child, :last-child 가상 클래스 선택자</title>
  <style>
    p { font-size: 14px }
    p:first-child mark { font-size: 14px; color: red; font-weight: bold }
    div { width: 450px; border: 1px solid black; padding: 3px }
    div > p:first-child { text-indent: 20px; font-size: 18px }
  </style>
</head>
<body>
  <h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
  <div>
    <p>가상 선택자는 <mark>가상의 형용사</mark> 같은 것으로, 시적으로 정의하지 않아도 가상으로 설정한 것처럼 동작한다.</p>
    <p>:first-child 가상 클래스는 첫 번째 자식인 요소에 일치하는 선택자이다.</p>
  </div>
</body>
</html>
```

CSS3 가상(Pseudo) 클래스 선택자 연습하기

가상 선택자는 **가상의 형용사** 같은 것으로, 명시적으로 정의하지 않아도 가상으로 설정한 것처럼 동작한다.

:first-child 가상 클래스는 첫 번째 자식인 요소에 일치하는 선택자이다.





- **:nth-of-type(an+b), :nth-of-last-type(an+b) 가상 클래스 선택자**
- **:first-of-type, :last-of-type 가상 클래스 선택자**

- **:nth-of-type(an+b) 가상 클래스 선택자**
 - 동일한 부모를 가지는 요소 중 처음부터 n번째의 **지정한** 요소 선택
- **:nth-of-last-type(an+b) 가상 클래스 선택자**
 - 동일한 부모를 가지는 요소 중 마지막부터 n번째의 **지정한** 요소 선택
- **:first-of-type 가상 클래스 선택자**
 - 동일한 부모를 가지는 요소 중 첫번째 **지정한** 요소 선택
- **:last-of-type 가상 클래스 선택자**
 - 동일한 부모를 가지는 요소 중 마지막 **지정한** 요소 선택
- **child와의 차이점**
 - child : 요소의 종류를 가리지 않고 무조건 지정된 순서만으로 요소를 선택
 - type : 요소의 종류와 지정된 순서를 모두 고려하여 선택



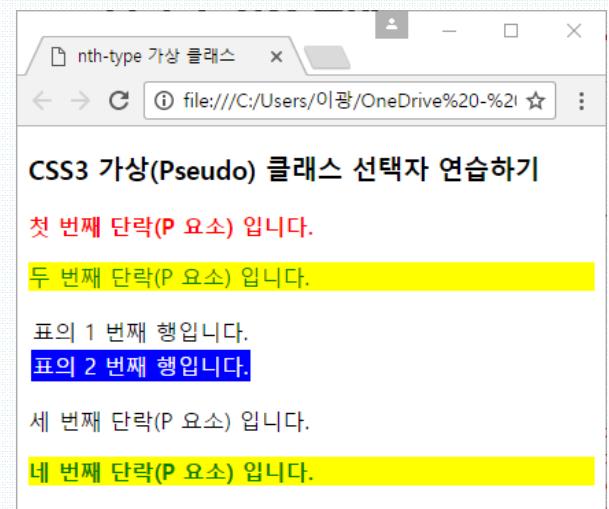


- :nth-of-type(an+b), :nth-of-last-type(an+b) 가상 클래스 선택자의 예
- :first-of-type, :last-of-type 가상 클래스 선택자의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>nth-type 가상 클래스</title>

    <style>
        tr:nth-child(even) { background: blue; color: white }
        p:first-of-type { color: red; font-weight: bold }
        p:last-of-type { color: blue; font-weight: bold }
        p:nth-of-type(even) { background: yellow; color: green; }
    </style>

</head>
<body>
    <h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
    <p>첫 번째 단락(P 요소) 입니다.</p>
    <p>두 번째 단락(P 요소) 입니다.</p>
    <table> <tr><td>표의 1 번째 행입니다.</td></tr> <tr><td>표의 2 번째 행입니다.</td></tr></table>
    <p>세 번째 단락(P 요소) 입니다.</p>
    <p>네 번째 단락(P 요소) 입니다.</p>
</body>
</html>
```





▣ :only-child, :only-of-type, :empty 가상 클래스 선택자

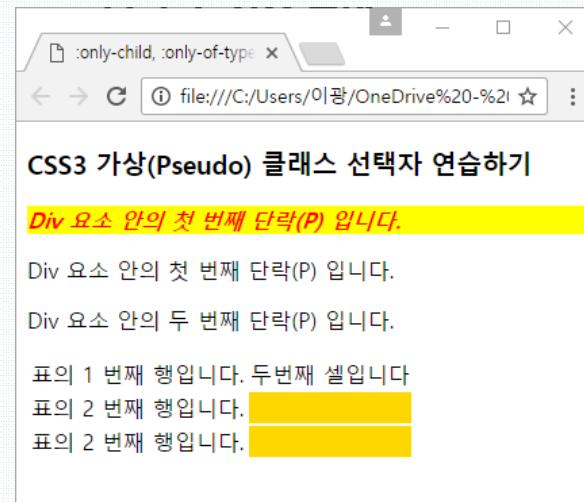
- **:only-child**
 - 부모 요소의 자식요소 중 지정된 요소가 유일한 자식요소일 경우 선택
 - [예] p:only-child
 - : 부모 요소의 자식 요소 중 <p>요소가 유일한 자식 요소일 경우 선택
- **:only-of-type**
 - 부모 요소의 자식요소 중 지정된 요소가 하나밖에 없을 경우 선택
 - [예] p:only-of-type
 - : 부모 요소의 자식 요소 중 <p>요소가 하나밖에 없을 경우, 해당 <p> 요소를 선택
- **:empty**
 - 지정된 요소가 자식이 없는 요소일 경우 선택





▣ :only-child, :only-of-type, :empty 가상 클래스 선택자의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>:only-child, :only-of-type, :empty 가상 클래스 선택자</title>
    <style>
        p:only-of-type { background: yellow }
        p:only-child { color: red; font-weight: bold; font-style: italic }
        td:empty { background: gold; }
    </style>
</head>
<body>
    <h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3>
    <div class="box600">
        <p>Div 요소 안의 첫 번째 단락(P) 입니다.</p>
    </div>
    <div class="box600">
        <p>Div 요소 안의 첫 번째 단락(P) 입니다.</p>
        <p>Div 요소 안의 두 번째 단락(P) 입니다.</p>
    </div>
    <table>
        <tr><td>표의 1 번째 행입니다.</td><td>두 번째 셀입니다</td></tr>
        <tr><td>표의 2 번째 행입니다.</td><td></td></tr>
        <tr><td>표의 2 번째 행입니다.</td><td></td></tr>
    </table>
</body>
</html>
```





■ ::first-line, ::first-letter 가상 요소 선택자

- **::first-line**
 - 어떤 문단에서 첫 번째 문자열을 선택
- **::first-letter**
 - 어떤 문단에서 첫 번째 문자를 선택

■ ::before, ::after 가상 요소 선택자

- **::before**
 - 문서 내에서 지정된 부분 앞에 특정 문자를 삽입
- **::after**
 - 문서 내에서 지정된 부분 뒤에 특정 문자를 삽입
- **::before와 ::after를 사용해 삽입할 문자의 내용은 content 속성을 사용**





▣ ::first-line, ::first-letter, ::before, ::after 가상 요소 선택자의 예

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>가상 요소 선택자</title>
    <style>
        p { font-size: 14px }
        div > p.line::first-line { color: blue }
        div > p.line::first-letter { color: green; font-size: 200% }
        div > p.note::before { content: "Read this - "; border: solid green }
        div > p.note::after { content: "- Remember this"; background-color: yellow; color: red }
    </style>
</head>
<body>
    <h3>CSS3 가상(Pseudo) 요소 선택자 연습하기 </h3>
    <div>
        <p class="line">Love many things, for therein lies the true strength, and whosoever loves much performs much, and can accomplish much, and what is done in love done well.</p>
    </div>
    <div>
        <p class="note">::before 가상 요소는 내용의 앞(before)에 ::after는 내용의 뒤(after)에 생성된 내용을 삽입한다.</p>
    </div>
</body>
</html>
```

