

HTML

표준화 문서를 기반으로 하는 지침서



# 속이 권은 HTML5 & CSS3

김명진 지음

04강

CSS3 Part-1 기본 및 선택자

#### 학습 목표

하나의 웹 문서를 작성하기 위해서는 HTML 문서와 디자인을 위한 CSS, 그리고 동작을 위한 자바스크립트가 서로 조화를 이루도록 해야 한다. HTML5로 업그레이드가 이루어지는 동안, CSS도 CSS2에서 CSS3로 업그레이드가 되면서 다양한 속성들이 추가되었다. 이번 장에 서는 CSS의 기본 사용 방법 및 선택자들에 대하여 살펴보도록 하겠다.

#### Section

- 1 CSS3 사용하기
- 2 선택자 이해하기
- 3 가상 선택자
- 4 조합자
- 5 선택자의 우선 순위

# O1 CSS3 ^[-8-5]

#### **!!!** 스타일시트를 사용해야 하는 이유

#### ♥️ 웹 문서의 내용과 디자인 분리

✓ 웹 문서의 내용을 수정해야 하는 경우, 디자인에 전혀 영향을 주지 않고 내용만 수정할 수 있거나, 내용에는 영향을 주지 않고 디자인을 수정하여 레이아웃을 변경하여 다른 느낌의 문서를 만들 수 있다.

#### 🥏 다양한 기능의 확장

✓ HTML 문서는 단순한 기능만을 제공하기 때문에 CSS를 사용해야 다양한 기능을 추가하거나 변경시킬 수 있다.

#### ▼일된 문서 양식 제공

✓ CSS를 사용하면 한 번의 속성을 정의함으로써 동시에 해당 요소에 속성들을 적용할 수 있다.

### ✓ 로딩 시간 단축

✓ 통일된 문서 양식 CSS를 사용함으로써 웹 브라우저가 읽어야 하는 문서의 내용도 작아지기 때문에 웹 문서의 로딩 시간이 단축된다.

# O1 CSS3 AT-8-OTTI

#### 🟪 스타일시트의 레벨(Level) 관리

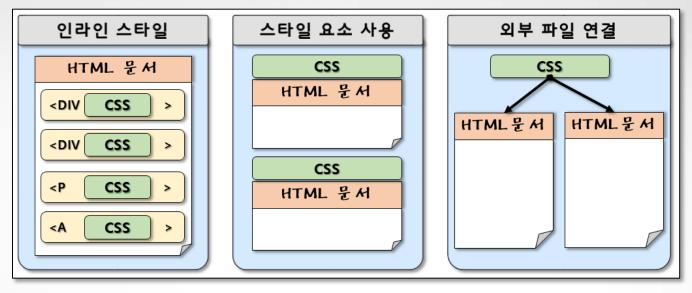
- CSS는 레벨(Level) 관리를 채택하여 기술 발전에 맞추어 기능들을 보완 및 추가하는 등으로 표준화를 진행하고 있다.
- CSS의 각 레벨은 이전 레벨의 기능들을(features) 수정 및 보완하고 새로운 기능들을 추가하는데 기반으로 하고 있다.
- 기본적으로 높은 레벨 수준의 기능들은(feature set) 낮은 레벨 수준의 상위 집합이 된다. 따라서 각 레벨 수준의 호환을 위해서 높은 수준의 주어진 기능에 허용되는 동작(behavior)들은 낮은 수준에서도 동작이 된다고 볼 수 있다.

#### 뿗 스타일시트의 모듈

- CSS Level 3부터는 기능에 따라 모듈로 분리하여 정의하고 있다.
- 웹의 기능이 발전함에 따라 HTML도 진화를 거듭하여 동적 애플리케이션 및 웹을 표현하는 디바이스들이 다양해짐에 따라서 웹의 표현도 더 다양해질 필요성이 확대되었기 때문.
- CSS2.1과는 비교할 수 없을 만큼 방대한 내용을 담고 있지만, 모듈들은 대부분 작성중에 있으며, 통합되거나 삭제될 수 있고, 필요에 따라서 새로운 모듈이 추가될 수도 있다.
- http://www.w3.org/Style/CSS/current-work



#### **!!! 스타일시트를 사용하는 방법**



#### **!!** 스타일의 형식

사용 방법

p { color: red; }

#### **!!** 인라인 스타일 - 스타일 속성으로 태그 안에 직접 적용

#### 사용 예제 🕟

〈p style="color: red"〉직접 태그 안에서 CSS를 적용하였습니다.〈/p〉

#### **!!** 스타일 요소 사용 – 동일한 파일 안에서 지정

#### CSS\_04-01\_StyleInFile.html

```
<!DOCTYPE html>
   <head>
   <title>HTML 스타일시트 연습</title>
4
5
6
   <style>
     color: #FFFFFF;
     background-color: #0000FF
   #p1 { color: gold }
  </style>
11
   </head>
12
   <body>
13
     <h3>HTML5 뉴스</h3>
     HTML5는 차세대 웹 표준으로 W3C에서 웹 표준화를 위해서 만들어진 새로운 ...
14
      여자에도 계속해서 개발 중에 있다.  2004년 6월 WHATWG에서 웹 애플리케이션 1.0이라는 이름으로 시작하였다.
15 l
16
17
   </body>
18
   </html>
```

### **## 외부 파일 연결 – 외부로부터 스타일을 읽어들여 지정**

#### **CSS\_04-01.css**

```
1  p {
    color: #FFFFF;
    background-color: #0000FF;
4  }
5  #p1 {
    color: gold;
7  }
```

#### CSS\_04-01\_StyleInFile.html

```
1<!DOCTYPE html>2<head>3<title>HTML 스타일시트 연습4link rel="stylesheet" href="CSS_04-01.css">5</head>6<body>7<h3>HTML5 뉴스</h3>8> HTML5는 차세대 웹 표준으로 W3C에서 웹 표준화를 위해서 만들어진 새로운 ...9현재에도 계속해서 개발 중에 있다.10< 2004년 6월 WHATWG에서 웹 애플리케이션 1.0이라는 이름으로 시작하였다.</p>11</hd>12</hd>
```

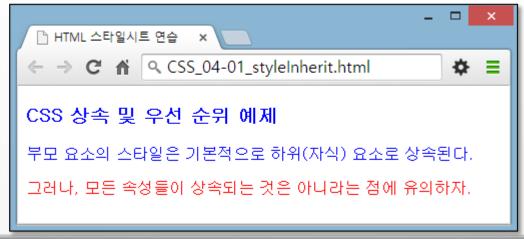


#### **!!! 스타일시트의 상속 및 우선순위**

인라인 스타일 방법 > 스타일 요소 방법(동일 파일) > 외부 파일 연결 방법

#### CSS\_04-01\_StyleInherit.html

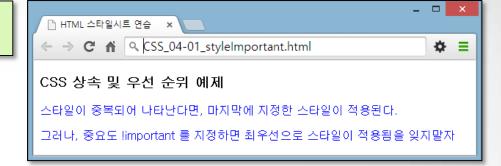
```
1 <!DOCTYPE html>
2 <head>
3 <title>HTML 스타일시트 연습</title>
4 <style>
5 body { color: blue }
6 </style>
7 </head>
8 <body>
9 <h3>CSS 상속 예제</h3>
10 부모 요소의 스타일은 기본적으로 하위(자식) 요소로 상속된다.
11 style="color: red">그러나, 모든 속성들이 상속되는 것은 아니라는 점에 유의하자.
12 </body>
13 </html>
```



#### **!!** 스타일시트의 중복 및 우선 순위

#### 사용 예제 🕟

p { color: red }
p { color: blue }



#### 사용 예제 🕟

p { color: red !important }
p { color: blue }

HTML 스타일시트 연습 ×
 CSS\_04-01\_styleImportant.html
 CSS 상속 및 우선 순위 예제
 스타일이 중복되어 나타난다면, 마지막에 지정한 스타일이 적용된다.
 그러나, 중요도 Iimportant 를 지정하면 최우선으로 스타일이 적용됨을 잊지말자

# O2 스크택자 이해하기

표준화 문서1	Selectors Level 3 - http://www.w3.org/TR/css3-selectors/
표준화 단계	Recommandation, (2011-09-29)
표준화 문서2	Selectors Level 4 - http://dev.w3.org/csswg/selectors4/
표준화 단계	Editor's Draft, (2014-05-28)

#### 罪 CSS2와의 차이점

- 선택자, 선택자 그룹, 단순 선택자의 기본적인 정의가 변경. CSS2에서는 "단순 선택자(Simple selector)"로 불리던 것이 선택자의 순서(sequence) 구성 요소로 사용되기 때문에 "단순 선택자의 순서(Simple Selector Sequence)"로 불리운다.
- 선택적인 네임스페이스(Namespace) 구성 요소가 전체 선택자(Universal Selector), 형식 선택자(Type Selector), 그리고 속성 선택자(Attribute Selector)에서 사용이 가능하게 되었다.
- 속성 선택자에 새로운 문자열을 포함하는 단순 선택자가 추가되었다.
- 새로운 가상 클래스(Pseudo-classes) 및 가상 요소(Pseudo-elements), 조합자(Combinator)가 추가되었다. 이로 인하여 가상 요소를 위한 "::"규칙이 새롭게 도입이 되면서 문법이 다시 작성 되었다.
- 프로파일(Profile)을 명세서에 포함시킴으로써 각 명세서에 의해 지원되는 선택자 집합을 정의하였다. 이로 인해 선택자가 CSS3의 독립적인 모듈로 존재함으로써 다른 명세서들도 선택자 명세서를 참조할 수 있게 되었다.

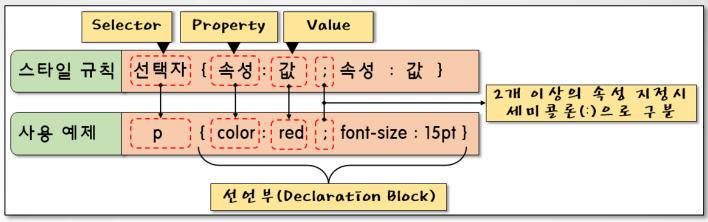
### ≝ 선택자 구문(Syntax)

- 선택자는 조합자로 구분되는 하나 이상의 단순 선택자의 순서가 연결된(chain) 것.
  - ✓ 하나의 가상 요소(Pseudo-element)는 단순 선택자의 마지막 순서에 추가될 수 있다.
  - ✓ 단순 선택자의 순서는 조합자로 구분되지 않는 단순 선택자의 연결이다.
  - ✓ 항상 형식 선택자 또는 전체(\*) 선택자로 시작되어야 하고 순서(sequence)에서는 허용되지 않는다.

단순 선택자	형식(Type) 선택자, 전체(*) 선택자, 속성(Attribute) 선택자, 클래스(Class) 선택자,
(Simple Selector)	아이디(ID) 선택자 및 가상 클래스(Pseudo-class)
조합자 (Combinator)	공백(whitespace), 보다 큰 기호(>), 더하기 기호(+) 및 틸드 기호(~) 공백은 조합자와 그 주위의 단순 선택자 사이에서 나타나고, [space], [tab], [line feed], [carriage return], [form feed]만이 될 수 있다.

# 02 스택자 이해하기

#### 🚜 스타일 규칙



- ➤ **type** selectors 형식 선택자
- > .class selectors 클래스 선택자
- ➤ #id selectors 아이디 선택자
- ➤ [attribute] selectors 속성 선택자
- ➤ :pseudo-class and ::pseudo-element 가상 클래스와 가상 요소
- ➤ Combinators 조합자

#### 🔡 선택자 그룹

# 개별 선택자 h1 { font-family: sans-serif } h2 { font-family: sans-serif } h3 { font-family: sans-serif }

# 02 스택자 이해하기(단순산택자)

#### 

Level	선택자 이름		설명
CSS2	* <b>전체</b> 선택자		전체에 스타일을 적용한다.
CSS1	요소 <b>형식</b> 선택자		지정한 요소에 스타일을 적용한다.
CSS1	요소 <b>.클래스이름 클래스</b> 선택자		지정한 클래스에 스타일을 적용한다.
CSS1	요소#아이디	아이디 선택자	지정한 아이디에 스타일을 적용한다.
CSS2	요소 <b>[속성]</b>	<b>속성</b> 선택자	지정한 속성에 해당하는 요소에 스타일을 적용한다.

#### # 전체(Universal) 선택자 – 모든 요소에 스타일 적용

- 웹 문서에 있는 모든 요소들에 스타일을 적용시키는 것으로, 모든 하위(자손) 요소에 스타일을 적용시키고자 할 때 주로 사용
- 전체 선택자를 사용하려면 별표(\*) 기호를 사용한다.

사용 방법	[네임스페이스] <*> { property: value; property: value; }
사용 예제	* { margin: 0; padding: 0; color: gray }

### # 전체(Universal) 선택자 – 모든 요소에 스타일 적용

전체 선택자 사용	전체 선택자 생략
*[hreflang =en]	[hreflang =en]
*warning	.warning
*#myid	#myid

※ div:first-child의 경우에는 혼선을 줄이기 위하여 div \*:first-child와 같이 별표 기호를 생략하지 않는 것이 더 좋을 수 있다.

#### ■ CSS3부터는 전체 선택자에 네임스페이스 요소가 선택적으로 사용 가능

ns *	네임스페이스 ns안에 있는 모든 요소들
* *	모든 요소들
<b> </b> *	네임스페이스 없는 모든 요소들
*	기본적인 네임스페이스가 지정되지 않은 경우, 이것은 * * 와 동등하다. 그렇지 않다면 ns가 기본 네임스페이스인 ns *와 동등하다.



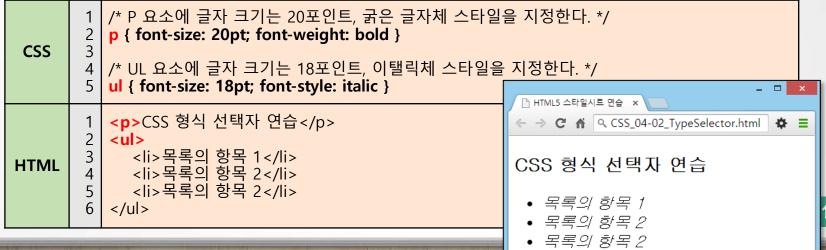
#### ■ 형식(Type) 선택자 – 특정 요소에 스타일 적용

- 형식 선택자는 스타일을 지정할 요소를 의미하는 것으로, HTML 요소의 형식과 관련이 있기 때문에 형식 선택자라 부른다.
  - ✓ 형식 선택자는 CSS 공식 명칭의 구문을 사용하여 작성되는 문서 언어 요소 형식의 명칭이다. 즉, HTML 요소에는 DIV, P, UL, TABLE 등의 다양한 요소들이 존재하는데, 이런 요소 이름에 따라 형식이 결정되기 때문이다.

```
사용 방법 [네임스페이스]|<요소> { property: value; property: value; ... }

사용 예제 table { color: red; }
ul { color: red; }
```

#### CSS\_04-02\_TypeSelector.html



# 02 산택자 이해하기(단순산택자)

#### # 클래스(Class) 선택자 - 특정 부분에 스타일 적용

- 같은 요소이지만, 각각 다른 스타일을 적용하려면 각 요소들을 구별할 수 있는 클래스 선택자를 이용한다
  - ✓ 요소 안에서 다중의 값을 가질 수 있는데, 기본적으로 모든 요소에 적용되겠지만, 특정 클래스를 가진 모든 요소를 선택하기 위해서는 요소 다음에 마침표(₃)를 넣어서 선택자를 사용하도록 한다.

 사용 방법
 요소.클래스이름 { 속성: 값; 속성: 값; ... }

 사용 예제
 .fancy { color: red; }

#### CSS\_04-02\_ClassSelector.html

 CSS
 1 p.select1 { font-size: 15pt; font-weight: bold } /\* 클래스이름이 select1인 p 요소 \*/
.select2 { font-size: 20pt; font-style: italic } /\* 클래스이름이 select2인 모든 요소 \*/

 HTML
 1 CSS 클래스 선택자 연습

 > c c class="select2">CSS 클래스 선택자 연습

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D x

 - D

CSS 클래스 선택자 연습

CSS 클래스 선택자 연습

http://www.google.com

# 02 산택자 이해하기(단순산택자)

#### **♯ 아이디(ID) 선택자 − 특정 부분에 스타일 적용**

- 문서 언어와 관계 없이 ID를 활용하여 요소를 고유하게 식별할 수 있는 것이 특징이다. HTML에서의 모든 ID 속성은 "id"로 지정한다.
  - ✓ 아이디 선택자는 요소 및 클래스 선택자와는 달리, 요소의 정체성을 나타내기 때문에 하나만 있어야 한다.

사용 방법	요소#아이디이름 { 속성: 값;	}	
사용 예제	① h1#fancy {color: red }	② #fancy { color: red }	③ *#z98y { color: red }

#### CSS\_04-02\_ldSelector.html

CSS	1 2	p#select1 { font-size: 15pt; font-weight: bold } /* 아이디이름이 select1인 p요소 선택 */ #select2 { font-size: 20pt; font-style: italic } /* 아이디이름이 select2인 모든 요소 선택 */	
HTML	1 2	<h2 id="title">CSS 아이디(ID) 선택자</h2>	_
	3	<pre> CSS 아이디 선택자 1   CSS 아이디 선택자 2</pre>	
			← → C ↑ Q CSS_04-02_IdSelector.html 🌣 🗏

CSS 아이디(ID) 선택자

CSS 아이디 선택자 2

CSS 아이디 선택자 1

### 3 속성(Attribute) 선택자 – 속성 값에 따른 스타일 적용

- 속성 선택자는 특정한 속성의 이름을 지정할 수 있는 것으로, 대괄호 '[' 와 ']'로 묶여 있으며, 대괄호 안에 찾고자 하는 속성 값을 지정한다.
  - ✓ 모든 요소에서 사용할 수 있는 거의 모든 속성들이 지원되기 때문에, 속성 값에 따른 스타일을 다르게 적용할 수 있다는 것이 특징이라 할 수 있다.

사용 방법 요소[네임스페이스|속성 연산자 값 ] { 속성: 값; }



#### CSS2에서의 속성 선택자의 연산자 종류

연산자	설명
요소[네임스페이스 속성]	속성 값과 관계없이 해당 속성을 사용하는 요소를 나타낸다.
요소[네임스페이스 속성 = 값]	속성 값이 정확히 <b>일치하는</b> 요소를 나타낸다.
요소[네임스페이스 속성 ~= 값]	속성 값을 <mark>공백으로 구분한</mark> 요소를 나타낸다.
요소[네임스페이스 속성  = 값]	속성 값이 <b>정확히 값과 일치</b> 하거나 <b>값으로 시작하고 바로 뒤에 "-" 기호가</b> 나타난 요소를 나타낸다.

① h1[title] { ... }
② span[class="example"] { ... }
③ span[hello="President"][goodbye="Dictator"] { ... }
④ a[rel~="copyright"] { ... }
⑤ a[hreflang|="en"] { ... }

### **≝** 속성(Attribute) 선택자 – 속성 값에 따른 스타일 적용

CSS3의 레벨3 및 레벨4에서 추가된 선택자의 연산자 종류

Level	연산자	설명
	요소[네임스페이스 속성 ^= 값]	속성 값의 <b>시작 부분의 문자와 일치하는</b> 요소를 나타낸다.
CSS3	요소[네임스페이스 속성 \$= 값]	속성 값의 끝 부분의 문자와 일치하는 요소를 나타낸다.
	요소[네임스페이스 속성 *= 값]	속성 값 <b>전체 중 어떤 일부분이라도 일치하는</b> 요소를 나타낸다.
CSS4	요소[네임스페이스 속성 <b>=</b> 값 <b>i</b> ]	속성 값이 ASCII 범위의 순열에 정확히 일치하는 요소를 나타낸다.

```
1 p[title^="hello"] { ... }
2 p[title$="bye"] { ... }
3 a[href*="toube"] { ... }
```

#### @namespace foo url(http://www.example.com);

사용 예제

```
[foo|attr=val] { color: blue } /* 첫 번째 규칙 */
[*|attr] { color: yellow } /* 두 번째 규칙 */
[|attr] { color: red } /* 세 번째 규칙 */
[attr] { color: green } /* 네 번째 규칙 */
```



### **≝ 속성(Attribute) 선택자 – 속성 값에 따른 스타일 적용**

#### CSS\_04-02\_AttributeSelector.html

```
a { line-height: 25pt; text-decoration: none; padding-left: 70px; background-repeat: no-repeat; }
           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;
 CSS
                        background-image: url(images/site_kyungnews.jpg) }
         6 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); }
           <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>
HTML
           <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="유투브" href="http://www.youtube.com/?gl=KR&hl=ko">유튜브</a><br>
```

#### CSS3 속성 선택자 연습하기

#### <u>즐겨찾기</u>

**♥**전자신문 전자신문

한겨래 한겨레신문

경향신문 *경향신문* 

OhmyNews 오마이뉴스

▼ Google 구글 검색

■ YouTube 유튜브

#### 🔡 가상 선택자

- 가상의 형용사 같은 것으로, 명시적으로 정의하지 않아도 가상으로 설정한 것처럼 동작하도록 하는 것이다. 즉, 웹 문서상에는 존재하지 않지만 임의로 가상 선택자를 지정하여 사용하는 것을 의미한다.
  - ✓ 가상 클래스(Pseudo-class)와 가상 요소(Pseudo-element)로 구분하여 사용

#### ➡ 가상 클래스(Pseudo-classes)

- 문서 트리를 벗어나거나 다른 단순 선택자를 활용하여 표현할 수 없는 정보를 토대로 한 선택이 가능하도록 한 것이 바로 가상 클래스의 개념으로, 슈도 클래스라고도 부른다.
  - ✓ 실제로 존재하지는 않지만, 요소들의 이름 이외의 속성이나 내용에 따라 분류(classify) 함으로써 임의로 가리키는 기능을 가상의 선택자로 지정하여 사용.



#### 즐링크 가상 클래스 - :link와 :visited

- ✓ :link 가상 클래스는 아직 방문한 적이 없는 링크에 적용
  - 기본적으로 파란색 글자와 밑줄로 표시되기 때문에, 밑줄을 없애거나 색상을 바꾸려고 할 때는 바로 이 :link 선택자를 이용해야 한다.
- ✓ :visited 가상 클래스는 방문한 적이 있는 링크에 적용된다.

#### 사용 예제 🕟

a:link { color: green }
a.outside:visited { color: blue }

<a class="outside" href="http://mySite.com">mySite 방문</a>

a 요소에서는 기본적으로 링크를 녹색으로 표시하지만, 클래스 이름 "outside"를 사용하는 링크에서는 방문했던 연결이면 파란색으로 표시된다.



동적 가상 클래스(Dynamic Pseudo-classes)

# 3

#### ^(용자 조치 가상 클래스 - :hover, :active, :focus

- ✓ :hover 가상 클래스는 사용자가 포인팅 장치로 특정 요소를 지정하는 동안 적용이 되지만, 반드시 활성되는 것은 아니다.
- ✓ :active 가상 클래스는 사용자에 의하여 요소가 활성화되는 동안(예를 들면, 사용자가 마우스를 눌렀다 놓는 동안) 적용된다.
- ✓ :focus 가상 클래스는 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) 적용된다.

a:hover { color: red; font-size: 18px }
a:active { background: yellow }
a:focus { background: yellow; color: red }
a:focus:hover { background: blue; color: white }

:hover 선택자는 반드시 :link 선택자와 :visited 선택자 다음에 위치시켜야 한다. 그렇지 않는다면 :hover 선택자의 color 속성(property)을 감추기 때문이다. 마찬가지로, :active가 :hover 다음에 위치해야 a 요소를 활성화 하고 그 위를 오버(hover)할 때 활성화된(active) 배경 색상이(yellow) 적용된다. 따라서 반드시 다음과 같은 순서로 적용시켜야 한다.

- ✓ A:link { ... }
- ✓ A:visited { ... }
- ✓ A:hover { ... }
- ✓ A:active { ... }



# 동적 가상 클래스(Dynamic Pseudo-classes)

#### CSS\_04-03\_PseudoClass\_Link.html

CSS	1 2 3 4 5 6 7	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 }
HTML	1 2 3 4 5 6 7	<pre><h3>CSS3 가상(Pseudo) 클래스 선택자 연습하기 </h3> 즐겨찾기 1 <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.khan.co.kr">경향신문</a>   <a href="http://www.ohmynews.com">오마이뉴스</a>   <a href="http://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a></pre>

전자신문 | 한겨레 신문 | 경향신문 | 오마이뉴스 | 유튜브

전자신문|한겨레신문|경향세문|오마이뉴스|유튜브

전자신문 | 한겨레 신문 | <mark>경향신문</mark> | 오마이뉴스 | 유튜브

<mark>전자신문</mark> | 한겨레 신문 | 경향신문 | 오마이뉴스 | 유튜브

전지시문 | 한겨레 신문 | 경향신문 | 오마이뉴스 | 유튜브

### :: 가상 클래스(Pseudo-classes)



#### 목적지(Target) 가상 클래스 - CSS3

# :target 가상 클래스

- ✓ 일부 URI는 리소스 내의 특정 위치를 참조하기 위하여 URI 뒤에 앵커(Anchor) 식별자 기호 #를 덧붙인다. 이러한 앵커가 있는 URI는 목적지 요소라고 하는 문서 내의 특정 요소로 연결되는 것이 특징이다.
- ✓ 일반적인 웹 문서에서 동일한 사이트의 웹 문서 또는 다른 사이트의 웹 문서로 이동할 때는 링크를 이용하지만, 동일한 문서 안에서 다른 위치로 이동하고자 할 때는 이와 같은 앵커를 이용한다.

http://example.com/top.html#section\_2 특정 HTML 문서에서 section\_2라는 이름의 앵커를 가리키는 URI

※ 목적지 요소는 :target 가상 클래스로 표현할 수 있지만, 문서 URI에 앵커가 없다면 그 문서에는 목적지 요소가 없는 것이다.

p:target { background: yellow; border: 1px dotted red }

이동되는 앵커 목적지의 p 요소를 빨간색의 점선으로 사각형을 그리고 배경을 노란색으로 변경한다.

#sample1:target { background: blue; color: white }

이동되는 앵커 이름이 #sample1인 곳으로 이동되는 목적지의 배경색을 파란색으로, 글자 색상을 흰색으로 변경한다.

\*:target { color: red }

\*:target::before { content: url(target.png) }

목적지 요소를 빨간색으로 변경하고 이미지가 있을 경우에 그 앞에 이미지를 놓는다. ::before 선택자에 대해 서는 학습하게 될 가상 요소를 참고하도록 한다.

### : 가상 클래스(Pseudo-classes)



### | 목적지(Target) 가상 클래스 - CSS3



#### :target 가상 클래스



#### CSS\_04-03\_ PseudoClass\_Target.html

CSS	1 2	<pre>p:target { background: yellow; border: 1px dotted red } #sample1:target { background: blue; color: white }</pre>
HTML	1 2 3 4 5	<a href="#sample1">새로운 콘텐츠로 이동하기 1</a>   <a href="#sample2">새로운 콘텐츠로 이동하기 2</a> 위 링크를 클릭하면 아래 콘텐츠로 이동하면서 스타일이 변경됩니다. <b>새로운 콘텐츠1새로운 콘텐츠2</b>

#### 새로운 콘텐츠로 이동하기 1 | 새로운 콘텐츠로 이동하기 2

위 링크를 클릭하면 아래 콘텐츠로 이동하면서 스타일이 변경됩니다.

새로운 콘텐츠1...

새로운 콘텐츠2...

#### 새로운 콘텐츠록(위동하기 1 | 새로운 콘텐츠로 이동하기 2

위 링크를 클릭하면 아래 콘텐츠로 이동하면서 스타일이 변경됩니다.

#### 새로운 콘텐츠1...

새로운 콘텐츠2...

#### 새로운 콘텐츠로 이동하기 1 | 새로운 콘텐事론 이동하기 2

위 링크를 클릭하면 아래 콘텐츠로 이동하면서 스타일이 변경됩니다.

새로운 콘텐츠1...

새로운 콘텐츠2...

### ⋾ 計 か け か き 引 か ( Pseudo-classes)



#### 언어적인 기상 클래스 - CSS3/CSS4



#### :lang(language) 가상 클래스

- ✓ 가상 클래스 :lang 선택자는 지정된 요소 중 language로 지정된 언어로 시작되는 요소를 선택한다.
  - HTML에서 언어는 lang 속성과 meta 요소 또는 프로토콜 정보의(HTTP 헤더와 같은) 조합에 의해서 결정된다.

#### CSS\_04-03\_PseudoClass\_Lang.html

CSS	1 2 3	p:lang(ko) { font-weight:bold; } /* 한국어 */ p:lang(it) { background:yellow; font-style: italic } /* 이태리어 */ p:lang(fr) { color: #f00; font-weight:bold; } /* 프랑스어 */ p:lang(de) { color: blue; } /* 독일어 */	
HTML	3	<pre>이 책을 보는 사람은 행복한 사람입니다 Questo libro è un uomo felice che vede. Ce livre est un homme heureux qui voit. Dieses Buch ist ein glücklicher Mann, der sieht.</pre>	

#### 이 책을 보는 사람은 행복한 사람입니다

Questo libro è un uomo felice che vede.

Ce livre est un homme heureux qui voit.

Dieses Buch ist ein glücklicher Mann, der sieht.

### ⋾ 計 か け か き 引 か ( Pseudo-classes)



사용자 인터페이스(UI) 상태 가상 클래스 - CSS3



#### :enabled, :disabled 가상 클래스

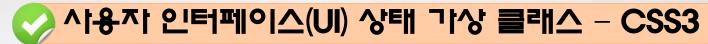
- ✓ :enabled 가상 클래스는 요소 중에서 활성화된 상태의 인터페이스 요소(이에 해당하는 활성화 해제 상태가 있다)를 나타낸다.
- ✓ :disabled 가상 클래스는 요소 중에서 활성화가 해제된 상태의 인터페이스 요소(이에 해당하는 활성화 상태가 있다)를 나타낸다.
  - 사용자 인터페이스에 영향을 미칠 수 있는 CSS 속성(property)인 display 및 visibility 속은 특정 요소의 활성화/해제 상태에 영향을 미치지 않는다.

```
input[type="text"]:enabled { background: yellow }
input[type="text"]:disabled { background: green }
```

이 예제에서, 선택자는 기본적으로 enabled 되어 있는 input 데이터의 배경색을 노란색으로 지정하고 disabled 시킨 input 데이터의 배경색은 녹색으로 지정한다.



#### ⋾ 計 か け か き 引 ム (Pseudo-classes)



## 을:checked 가상 클래스

- ✓ 사용자는 라디오 버튼이나 체크박스 요소를 클릭해서 상태를 반전(토글) 시킬 수 있다.
  - 사용자가 선택하면 일부 메뉴 항목이 "체크 표시" 된다. 해당 요소를 "on"시키면 :checked 가상 클래스가 적용된다.

input:checked { background: red; }

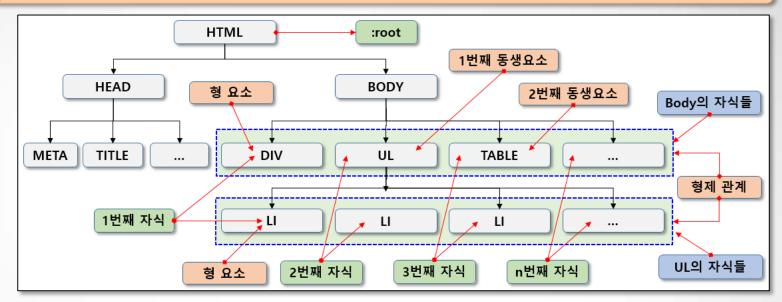
이 예제에서, 라디오버튼과 체크박스에서 체크가 되어 있는 input 데이터에는 배경색을 빨간색으로 지정한다.

#### CSS\_04-03\_PseudoClass\_State.html

</form>

	CSS				: <mark>홍길동</mark> : 한국
•		1 2 3	<form action=""> 이름 : <input type="text" value="홍길동"/> 국가 : <input disabled="" type="text" value="한국"/></form>		남자 ○ 여자 나전거 소유 ■ 자동차 소유
	HTML	4 5	<input checked="" name="gender" type="radio" value="male"/> 남자 <input name="gender" type="radio" value="female"/> 여자 	4	
		6 7	<input checked="" type="checkbox" value="Bike"/> 자전거 소유 <input type="checkbox" value="Car"/> 자동차 소유		29

#### ⋾ 計 か け か き 引 ム (Pseudo-classes)



## 🍧:root 가상 클래스

- ✓ 웹 문서 안의 최상위 요소(루트)를 선택하여 스타일을 적용
  - HTML 문서의 경우에는 HTML 요소가 루트 요소가 되기 때문에 이 선택자를 이용하면 HTML 전체에 동일한 스타일을 적용시킬 수 있다.

### :: 가상 클래스(Pseudo-classes)



#### 트리 구조적인 기상 클래스 - CSS2/CSS3/CSS4



#### :nth-child(an+b) 가상 클래스

**(1/2)**]

- ✓ 동일한 부모 요소 안에서 n번째에 있는 자식 요소를 선택
  - 문서 트리에서 양의 정수 또는 n의 제로 값에 대하여 앞에 an+b-1 형제가 있고 부모 요소를 갖춘 요소를 나타낸다. a와 b의 값은 반드시 정수(양수, 음수, 또는 제로)여야 하고 첫 번째 자식의 인덱스는 1이 된다.
  - an+b의 값을 odd(홀수), even(짝수)와 같은 방법으로도 지정할 수 있다. (odd는 2n+1과 동일한 의미, even은 2n과 동일한 의미)

2n+1	a의 값이 2이고 b의 값이 1이기 때문에 자식 요소의 그룹을 2개로 나누고, 각 그룹에서 1번째 인덱스는 홀수 번째가 된다. 따라서 2n+1은 odd와 같게 되는 것이다.		
tr:nth-child(2n+1)		HTML 표의 홀수 번째 행들을 선택한다. :nth-child(odd)와 동일하다	
tr:nth-child(2n+0)		HTML 표의 짝수 번째 행들을 선택한다. :nth-child(even) 및 tr:nth-child(2n)와 동일	
tr:nth-child(10n-1)		요소의 9, 19, 29번째, 자식 요소를 선택한다. :nth-child(10n+9)와 동일하다	
tr:nth-c	child(-n+6)	첫 번째 ~ 6번째 요소까지 선택한다	

1 tr:nth-child(1) { font-weight: bold }

- 3 tr:nth-child(even) { background: blue; color: white }
- 2 tr:nth-child(2n+1) { background: red; color: white } | 4 tr:nth-child(-n+3) { font-size: 16px }
- ① HTML 표에서 첫 번째 행의 글꼴 굵기를 굵게 지정한다.
- ② HTML 표에서 (2n+1)번째인 홀수 번째 행의 배경색을 빨간색으로, 글자 색상을 흰색으로 지정한다.
- ③ HTML 표에서 짝수(even)번째인 행의 배경색을 파란색으로, 글자 색상을 흰색으로 지정한다.
- ④ HTML 표에서 첫 번째 행에서 3번째 행까지의 글꼴의 크기를 16픽셀로 지정한다.

### ⋾ 計 か け か き 引 か ( Pseudo-classes)



#### 트리 구조적인 가상 클래스 - CSS2/CSS3/CSS4



#### :nth-child(an+b) 가상 클래스

(2/2)

✓ 동일한 부모 요소 안에서 n번째에 있는 자식 요소를 선택

2 <b>n</b> +1	2n+1은 odd와 같게 되는 것이다. ✓ (2 x 0) + 1 = 1 (1번째 요소) ✓ (2 x 1) + 1 = 3 (3번째 요소) ✓ (2 x 2) + 1 = 5 (5번째 요소)  2n+0은 even과 같게 되는 것이다. ✓ (2 x 0) + 0 = 0 (선택 안함) ✓ (2 x 1) + 0 = 2 (2번째 요소) ✓ (2 x 2) + 0 = 4 (4번째 요소)	
3 <b>n</b> +3	3n+3은 (3 x n) + 3으로 계산되기 때문에 3, 6, 9번째 요소 선택  ✓ (3 x 0) + 3 = 3 (3번째 요소)  ✓ (3 x 1) + 3 = 6 (6번째 요소)  ✓ (3 x 2) + 3 = 9 (9번째 요소)	
10 <b>n</b> -1	10n-1은 (10 x n) - 1로 계산되기 때문에 9, 19, 29번째 요소 선택  ✓ (10 x 0) - 1 = -1 (선택 안함)  n-1 ✓ (10 x 1) - 1 = 9 (9번째 요소)  ✓ (10 x 2) - 1 = 19 (19번째 요소)  ✓ (10 x 3) - 1 = 29 (19번째 요소)	
- <b>n</b> +6	-n+6은 (-n) + 6로 계산되기 때문에 1 ~ 6번째 요소 선택  ✓ -0 + 6 = 6 (6번째 요소)  ✓ -1 + 6 = 5 (5번째 요소)  ✓  ✓ -5 + 6 = 1 (1번째 요소)  ✓ -6 + 6 = 0 (선택 안함)	

### ⋾ 計 か け か き 引 か ( Pseudo-classes)



#### 트리 구조적인 가상 클래스 - CSS2/CSS3/CSS4



#### :nth-last-child(an+b) 가상 클래스

✓ 끝에서부터 n번째에 있는 자식 요소를 선택하여 스타일을 적용

tr:nth-last-child(2)	HTML 표에서, 마지막에서 두 번째 행을 나타낸다.
tr:nth-last-child(-n+2)	HTML 표에서, 마지막 행에서 마지막 두 번째 행까지를 나타낸다.



#### :first-child, :last-child 가상 클래스

- ✓ :first-child 가상 클래스는 다른 일부 요소의 첫 번째(first) 자식인 특정 요소를 나타내는 것으로,:nth-child(1)와 동일하다.
- ✓ :last-child 가상 클래스는 다른 일부 요소의 마지막(last) 자식인 특정 요소를 나타내기 때문에 :nth-last-child(1)과 동일하다.

p:first-child mark { font-size: 14px; font-weight: bold; color: red }

첫 번째 자식인 p 요소의 하위 요소들 가운데 어떤 mark 요소

ol > **li:last-child** { font-size: 14px; font-weight: bold; color: red }

순서 목록 ol 요소의 마지막 자식인 목록 항목 li 요소

### ⋾ お → 라스(Pseudo-classes)





#### :first-child, :last-child 가상 클래스

#### CSS\_04-03\_PseudoClass\_firstChild.html

CSS	1 2 3 4	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 }	
HTML	1 2 3 4	<div>가상 선택자는 <mark>가상의 형용사</mark> 같은 것으로, 명시적으로 정의하지 않아도 가상으로 설정한 것처럼 동작한다.<first-child p="" 가상="" 번째="" 선택자이다.<="" 요소에="" 일치하는="" 자식인="" 첫="" 클래스는=""></first-child></div>	

가상 선택자는 <mark>가상의 형용사</mark> 같은 것으로, 명시적으로 정의하지 않아도 가상으로 설정한 것처럼 동작한다.

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



트리 구조적인 가상 클래스 - CSS2/CSS3/CSS4



#### :nth-of-type(an+b) 가상 클래스

- ✓ 문서 트리에서 양의 정수 또는 n의 제로 값에 대하여 앞에 동일한 확장 요소의 an+b-1 형제가 있고 부모 요소를 갖춘 요소를 나타낸다.
  - 즉, 동일한 부모 요소 안의 형제 요소들 중에서 다른 종류의 요소를 제외하고 지정한 요소를 선택한다.
  - 이 가상 클래스는 형제 요소들이 다양하게 여러 개 나열되어 있을 때, 특정한 위치의 지정한 요소만을 선택해서 스타일을 적용시킬 수 있다는 장점이 있다.

#### img:nth-of-type(2n+1) { float: left }

img:nth-of-type(2n) { float: right }

동일한 부모 요소 안에 있는 img 요소만을 계산하여 홀수((2n+1) 또는 odd) 번째에 있는 이미지들은 왼쪽으로 배치하고, 짝수(2n 또는 even) 번째에 있는 이미지들은 오른쪽으로 배치하도록 지정한 것이다.

p:nth-of-type(even) { background: yellow; color: green; }

동일한 부모 요소 안에 있는 p 요소만을 계산하여 짝수(2n 또는 even) 번째에 있는 단락의 배경색을 노란색으로, 글꼴의 색상을 녹색으로 지정

### ⋾ 計 か け か き 引 か ( Pseudo-classes)



#### 트리 구조적인 가상 클래스 - CSS2/CSS3/CSS4



#### :nth-last-of-type(an+b) 가상 클래스

body > h2:nth-of-type(n+2):nth-last-of-type(n+2) { color: blue }

첫 번째와 마지막을 제외한 body 요소의 h2 자식을 모두 나타내기 위한 선택자의 예를 나타낸 것으로 다음과 같이 :not() 가상 클래스를 사용해도 된다.

body > h2:not(first-of-type):not(:last-of-tpye)



#### :first-of-type, :last-of-type 가상 클래스

- ✓ :first-of-type 가상 클래스는 동일한 부모 요소 안에서 형제 관계 요소 중에서 첫 번째 형제 요소를 선택하고 :nth-of-type(1)과 동일하다.
- ✓ :last-of-type 가상 클래스 또한 동일한 부모 요소 안에서 형제 관계 요소 중에서 마지막 형제 요소를 선택하고 :nth-last-of-type(1)과 동일하다.
- ✓ 이 가상 클래스들은 :nth-child 선택자와는 다르게, 형제 요소들 중에서 다른 종류의 요소가 있어도 지정한 요소만을 대상으로 계산한다는 점이 다르다.

# **!!** 가상 클래스(Pseudo-classes)



### 트리 구조적인 기상 클래스 - CSS2/CSS3/CSS4



#### :first-of-type, :last-of-type 가상 클래스

#### p:first-of-type { color: red }

p:last-of-type { color: blue }

동일한 부모 요소 안에서 맨 처음 나타나는 첫 번째 P 요소의 색상을 빨간 색으로 지정하고, 마지막 P 요소의 색상을 파란색으로 지정한다.

#### tr > td:last-of-type { background: blue; color: white }

table 요소의 행 요소인 tr 중에서 마지막 데이터 셀인 td 요소의 배경색을 파란색으로, 글자의 색상은 흰색으로 지정한다.

#### CSS 04-03 PseudoClass firstChild.html

CSS	2	tr:nth-child(even) { background: blue; color: white } p:first-of-type { color: red; font-weight: bold } /* 첫 번째 p 요소 선택 */ p:last-of-type { color: blue; font-weight: bold } /* 마지막 P 요소 선택 */ p:nth-of-type(even) { background: yellow; color: green; } /*짝수 번째 p 요소 선택 */				
HTML	1 2 3 4 5 6 7	첫 번째 단락(P 요소) 입니다.   두 번째 단락(P 요소) 입니다.   エ의 1 번째 행입니다.	첫 번째 단락(P 요소) 입니다.			
		표의 1 번째 행입니다.    표의 2 번째 행입니다.	두 번째 단락(P 요소) 입니다.			
		세 번째 단락(P 요소) 입니다. 네 번째 단락(P 요소) 입니다.	표의 1 번째 행입니다.			
			표의 2 번째 행입니다.			

세 번째 단락(P 요소) 입니다.

네 번째 단락(P 요소) 입니다.

# ⋾ 計 か け か き 引 か ( Pseudo-classes)





- ✓ 부모 요소 안에서 자식 요소가 유일하게 하나일 때(다른 자식 요소가 있으면 안됨) 선택
  - :first-child, :last-child, :nth-child(1), :nth-last-child(1)와 동일

p:only-child { color: red; font-weight: bold; font-style: italic }

동일한 부모 요소 안에 있는 자식 요소들 중에서 p 요소가 유일한 자식 요소일 때, 글꼴을 빨간색의 굵은 이탤릭체로 지정한다.



- ✓ 동일한 부모 요소의 형제 요소 중에서 지정한 요소가 유일하게 하나밖에(다른 자식 요소들이 있어도 관계 없음) 없을 경우에 선택
  - :first-of-type, :last-of-type, nth-of-type(1), :nth-last-of-type(1)와 동일

p:only-of-type { background: yellow }

동일한 부모 요소 안에 있는 자식 요소들 중에서 지정한 P 요소가 유일하게 있을 때, 배경색을 노란색으로 지정한다.

# ⋾ 計 か け か き 引 か ( Pseudo-classes)



# Sempty 가상 클래스

✓ 자식이(텍스트가 아닌) 없는 요소를 나타낸다.

# Siblank 가장 클래스

✓ 내용이 없는(공백 제외) 요소를 나타낸다.

# ⋾ 計 か け か き 引 か ( Pseudo-classes)



부정(Negation) 가상 클래스 -CSS3/CSS4



- ✓ 단순 선택자(부정 가상 클래스 자체 제외)를 인자로 하는 기능적 표기로써, 인자로 전달된 단순 선택자 X로 표현되지 않는 특정 요소를 나타낸다.
  - 괄호 ()안에는 지정된 요소 또는 단순 선택자만 올 수 있기 때문에 :not(:not(···))와 같이 사용할 수 없다.
  - button:not([DISABLED]) /\* 해제되지 않은 HTML 문서의 모든 버튼 요소\*/
  - \*:not(FOO) /\* FOO 요소를 제외한 모든 요소 \*/
  - html | \*:not(:link):not(:visited) /\* 링크를 제외한 모든 html 요소 \*/

#### **:not(h3)** { font-size: 14px }

p:not(.sample) { color: red }

h3 이외의 모든 요소들의 글꼴 크기를 14픽셀로 지정하고, class="sample"이 아닌 P 요소에 글 꼴을 빨간색으로 지정하는 것이다.

input:not([type="text"]) { background: yellow }

input 요소 중에서 type값이 "text"가 아닌 요소들의 배경색을 노란색으로 지정하는 것이다.

✓ Selectors Level 4에서는 기능이 확장되어 다음과 같이 괄호 안에 여러 개의 가상 클래스를 지정할 수 있도록 허용하고 있다.

- 가상 요소는 문서 언어(HTML)에 의해 기록된 문서 트리에 관한 추상적 개념을 생성하는 것이다.
  - ✓ 예를 들어, 문서 언어는 요소의 내용(content)에서 첫 번째 문자나 첫 번째 줄에 접근하는 방법을 제공하지 않는다. 이렇게 가상 요소는 접근할 수 없는(존재하지 않기 때문에) 정보를 참조할 수 있도록 다양한 수단을 제공하기 때문에 CSS에 없는 내용에 스타일을 지정할 수 있도록 해준다.
- 가상 요소 이름 앞에 2개의 콜론(::) 기호를 사용하는데, 이는 가상 클래스와 가상 요소를 구별하기 위해서 CSS Selector Level 3에서 도입된 문법이다.
- 현재 스타일시트와의 호환성을 위해 사용자 브라우저는 CSS1과 CSS2에 도입된 1개의 콜론(:) 기호를 사용하는 문법을 허용(:first-line,:first-letter,:before,:after)하고 있다.
- 표준화가 완료되면, 1개의 콜론 기호는 호환성이 허용되지 않기 때문에 가능하면 2개의 콜론 기호를 사용하도록 하자.
- 선택자 당 한 개의 가상 요소만이 나타날 수 있기 때문에 반드시 선택자의 대상을 나타내는 단순 선택자 순서 다음에 나타나야 한다. 다음 버전인 Selector Level 4에서는 선택자 당 복수의 가상 요소를 허용하도록 하고 있다.

# ::first-line 가상 요소

- ✓ 어떤 문단에서 첫 번째 문자열에 일치하는 특정 요소에서만 사용할 수 있으며, 블록 박스, 인라인 블록, 테이블 캡션, 또는 테이블 셀 등의 블록 레벨(block-level)과 유사한 컨테이너에 붙을 때에만 효과를 가질 수 있다.
- ✓ 이 가상 요소에서 첫 번째 문자열의 길이는 페이지의 폭, 글꼴 크기 등 여러 요인들에 의해서 다르게 나타날 수 있다는 점에 유의해야 한다.

#### p::first-line { color: blue }

P 요소를 사용하는 문단의 첫 번째 줄의 텍스트 색상을 모두 파란색으로 변경하라는 뜻이다. 여기에서의 p:first-line은 실제 HTML의 어떤 요소에도 일치하지 않지만, 각 문단의 가상 요소에 일치하 는 시작 줄을 삽입한다.

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.

✓ ::first-line 가상 요소는 인라인 레벨 요소와 유사하지만, 일정한 제약이 따르기 때문에 다음의 CSS 속성들만이 이 가상 요소에 적용될 수 있다.

font, color, background, word-spacing, letter-spacing, text-decoration, vertical-align, text-transform, line-height, text-shadow, clear



### ::first-letter 가상 요소

- ✓ 문자열에서(이미지나 인라인 테이블과 같은) 다른 내용이 앞에 오지 않으면 특정 요소의 첫 번째 글자를 나타낸다.
  - 이 가상 요소의 특징은 일반적인 인쇄 효과에 해당하는 "Initial caps"와 "drop caps"에도 활용할 수 있다는 점이다.
  - 글자가 숫자일 경우에는 이 가상 요소가 적용 된다. (예를 들어, 6,700만원 이라고 한다면 6은 아주 많은 돈이기 때문이다.)

```
p { font-size: 14px }
p::first-letter { color: green; font-size: 200% }
p::firtst-line { color: blue }
```

P 요소를 사용하는 문단의 첫 번째 글자의 색상을 녹색으로, 글자의 크기를 P 요소의 글자 크기보다 2배로 변경한다. 그리고 첫 번째 문자열의 나머지는 파란색으로 지정하고, 나머지 단락은 14픽셀의 검은색 색상으로 나타난다.

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.

✓ 다음 속성들만이 이 가상 요소에 적용될 수 있다.

font, color, background, text-decoration, vertical-align(float이 none일 때만), text-transform, line-height, margin, padding, border, float, text-shadow, clear



### ::before, ::after 가상 요소

✓ ::before와 ::after 가상 요소들은 요소 내용의 앞(before) 또는 뒤(after)에 생성된 내용의 스타일과 위치를 지정한다. 그리고 content 속성은 이들 가상 요소들과 조합하여, 무엇이 삽입되는가를 지정한다.

```
p.note::before { content: "Read this – "; border: solid green; }
p.note::after { content: " – Remember this"; background-color: yellow; color: red }
이 예제는, 클래스 이름이 note인 P 요소의 내용 앞에 content 속성의 "Read this –" 문자열을 삽입하고 녹색 실선으로 테두리를 그린다. 또한 P 요소의 내용 뒤에 " – Remember this" 문자열을 추가하고 노란색 배경에 글자의 색상을 빨간색으로 지정한다.
```

Read this - :before 가상 요소는 내용의 앞(before)에 생성된 내용을 삽입합니다. - Remember this

- 선택자 개념은 선택자의 연결이라 할 수 있다. 즉, 앞에서 살펴본 선택자를 문맥이나 요소의 구조를 기반으로 하여 조합하는 방법이 바로 조합자이며, 결합자라고도 부른다.
  - ✓ 만일, 테이블 뿐만 아니라 목록에도 색상을 빨간색으로 적용시키고 싶다면 table, ul {color: red }와 같이 콤마(,)를 사용하면 된다. 이렇게 선택자는 CSS내의 단순한 선택자들을 조합하는 방법으로 지정한 구역에만 적용된다

구분	Level	조합자	설명		
<b>하위</b> 조합자	CSS1	선택자1 선택자2	선택자2가 선택자1에 반드시 포함되어 있을 경우 선택		
<b>자식</b> 조합자	CSS2	선택자1 > 선택자2	부모 선택자1의 직계 자식 선택자2를 선택		
<b>인접 형제</b> 조합자	CSS2	선택자1 + 선택자2	선택자1 바로 다음에 위치한 선택자2를 선택		
<b>일반 형제</b> 조합자	CSS3	선택자1 ~ 선택자2	선택자1 뒤에 인접하여 나타나는 모든 선택자2를 선택		
그룹 조합자	CSS1	선택자1, 선택자2	선택자1과 선택자2 모두 선택		
<b>참조</b> 조합자	CSS4	선택자1 <b>/속성</b> 선택자2	선택자1의 속성에 의해 참조된 선택자2 아이디 선택		



### 🔡 웹 문서의 요소들 관계

### CSS\_Study03\_Combinator.html

LI

.class

LI

.id

LI

attribute

.pclass

```
<div id="main">Div 요소의 아이디 #main
     선택자 연습1
    <div>아이디 #main 하위의 div 요소1</div>
     ul>
        cli class="class">클래스(Class) 선택자
        cli class="id">아이디(ID) 선택자
        cli class="attribute">속성(Attribute) 선택자
        cli class="pclass">가상 클래스(Pseudo-class) 선택자li class="pelement">가상 요소(Pseudo-elements) 선택자
10
     11
12
        2014러시아 소치 동계올림픽
        2018대한민국 평창 동계올림픽
13
14
     <div>아이디 #main 하위의 div 요소2
15
16
        선택자 연습3
        선택자 연습4
17
18
     </div>
19
     <div>
20
        < T > 조합자는 CSS내의 단순한 선택자들을 조합하는 방법으로 지정한 구역에만 적용된다
21
     </div>
                               BODY
   </div>
                  제일
                        1번째
                                DIV
                                    #main
                  큰형
                        동생
             형제
                          DIV
                                     TABLE
                                UL
                                           DIV
                                                 DIV
             관계
```

Р

LI

.pelement

P



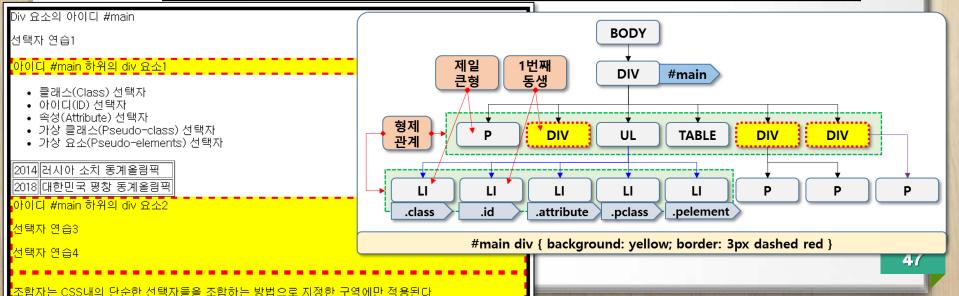
#### 하위 조합자(Descendant Combinator)

✓ (선택자1 선택자2)와 같이 선택자와 선택자의 조합을 공백으로 연결하여 선언한다. 이 공백의 의미는 모든 하위 요소를 나타내는 것이기 때문에 앞의 선택자1의 하위 요소 중에서 뒤의 선택자2에 해당하는 모든 요소를 선택하는 방법.

#### h1 mark { color: red }

#main div { border: 1px solid red }

h1 요소에 포함되는 하위 mark 요소의 색상을 지정하고, 아이디로 #main을 사용하는 요소에 포함되는 하위 div 요소의 테두리를 1픽셀의 굵기로 빨간 실선으로 그리도록 지정한 것이다.







#### 자식 조합자(Child Combinator)

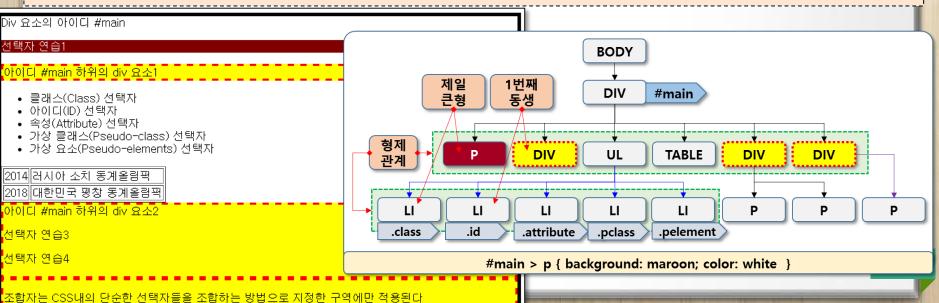
✓ (선택자1 > 선택자2)와 같이 선택자와 선택자의 조합을 > 기호로 연결하여 선언한다. 앞의 선택자1을 부모 요소로 하여 포함된 하위 직계 자식 선택자2 요소를 선택하는 방법.

#### body > p { font-size: 16px }

#main > p { color: red }]

body 요소의 직계 자식 p 요소에만 글꼴의 크기를 16픽셀로 지정하는 것으로, 만일 p 요소가 div 요소 같은 블록 안에 포함되어 있다면, 직계 자식이 아니기 때문에 스타일이 적용되지 않는다. 두 번째 예제 또 한 아이디를 #main으로 갖는 요소의 직계 자식 p 요소의 글자 색상을 빨간색으로 지정한 것이다.

※ 하위 조합자가 자식 요소 뿐만 아니라 자식의 자식까지 모두 포함하는 반면에, 자식 조합자는 자식 요소에만 스타일이 적용된다는 차이점이 있다.





# 인접 형제 조합자(Adjacent Sibling Combinator)

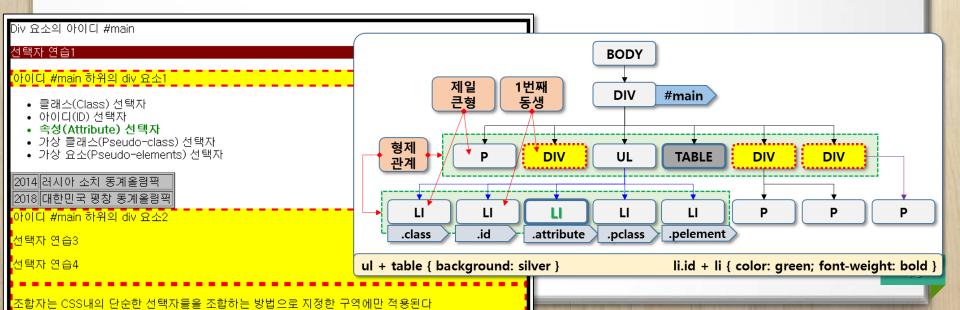
✓ (선택자1 + 선택자2)와 같이 선택자와 선택자의 조합을 + 기호로 연결하여 선언한다. 동일한 부모 요소를 갖는 형제 관계인 선택자1 <u>바로</u> 다음에 나타나는 동생 선택자2 요소만 선택하는 방법.

h1 + p { font-size: 16px }

h1 + h2 { font-style: italic }]

h1 요소의 바로 동생인 p 요소에만 글꼴의 크기를 16픽셀로 지정하는 것으로, 만일 h1 요소와 p 요소 사이에 다른 요소들이 존재하는 경우에는 적용되지 않는다.

두 번째 예제 또한, h1 요소와 h2 요소 사이에 다른 형제 요소가 있으면 스타일이 적용되지 않는다.







### 을 일반 형제 조합자(General Sibling Combinator)

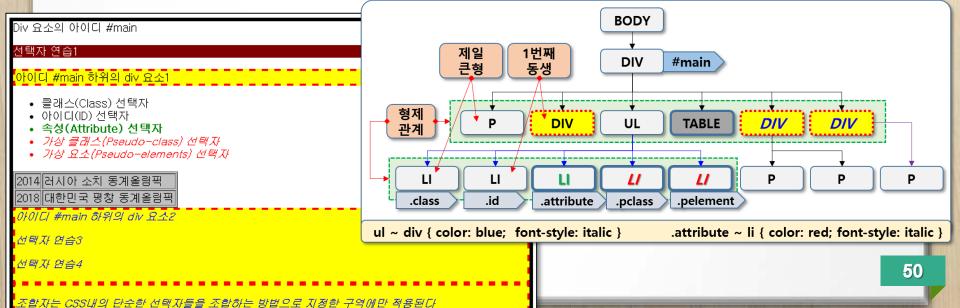
✓ (선택자1 ~ 선택자2)와 같이 선택자와 선택자의 조합을 ~ 기호로 연결하여 선언한다. 동일한 부모 요소를 갖는 형제 관계인 선택자1 다음에 나타나는 형제 관계의 선택자2 요소들을 선택하는 방법.

h1 ~ pre { font-size: 16px }

.id ~ li { font-style: italic }

h1 요소 다음에 나타나는 동생 pre 요소에 스타일이 적용된다.

마찬가지로, 클래스 이름이 id인 요소 다음에 나타나는 동생 li 요소들에 스타일이 적용된다.







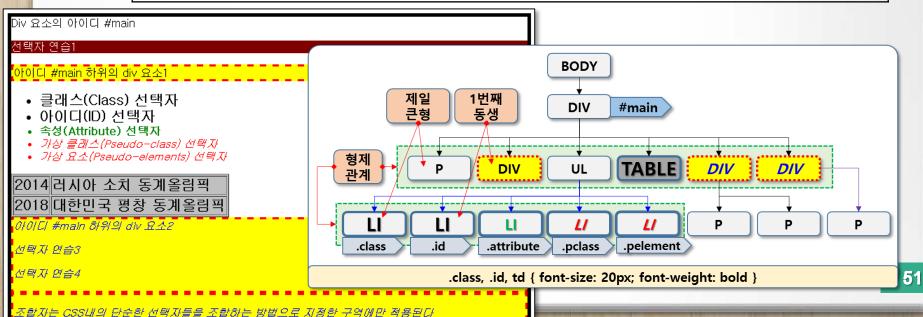
### 그룹 조합자(Group Combinator)

- ✓ (선택자1, 선택자2, 선택자3, …)와 같이 선택자와 선택자의 조합을 콤마(,) 기호로 연결하여 선언한다(콤마 앞뒤에는 공백이 나타날 수 있다).
- ✓ 여러 선택자 각각에 의해 선택된 모든 요소의 결합체를 선택하는 방법.
- ✔ 여러 요소에 동일한 스타일을 적용해야 할 경우에 매우 유용하다.

#### h1, h2, p { color: green }

.id, table { font-size: 20px }

h1, h2, p 요소 모두 동일하게 글자 색상을 녹색으로 변경하고, 클래스 이름이 id인 요소와 table 요소의 글꼴 크기를 동일하게 20픽셀로 지정한다.







#### 선택자의 우선순위

✓ CSS에서는 기본적으로 하나의 요소에 여러 스타일이 지정되는 경우를 대비해서, 선택자의 우선 순위를 정해서 관리하고 있다. 선택자의 우선 순위는 구체성(Specificity)을 기준으로 하기 때문에 다음과 같이 선택자의 구체적인 정도에 따라서 순위가 결정된다.

[W3C] Selector Level 3#specificity - http://www.w3.org/TR/selectors/#specificity



### 선택자의 구체성 계산

	구분	Α	В	С		
선택자 구성	선택자	아이디	클래스, 속성, 가상 클래스	형식, 가상 요소	계산값	우선 순위
	계산 단위	100단위	10단위	1단위		
*	전체	0	0	0	0	10
LI	형식	0	0	1	1	9
UL LI	형식 형식	0	0	2	2	8
UL OL+LI	형식 형식 + 형식	0	0	3	3	7
H1 + *[REL=up]	형식 + 속성	0	1	1	11	6
UL OL LI.red	형식 형식 형식.클래스	0	1	3	13	5
LI.red.level	형식.클래스.클래스	0	2	1	21	4
#x34y	아이디	1	0	0	100	3
#s12:not(LI)	아이디:가상클래스	1	0	1	101	2
UL#list1 LI	형식#아이디 형식	1	0	2	102	1



- 1 색상 이해하기
- 2 글꼴 스타일