

4강. CSS3 [1] : 기본 및 선택자

1. CSS3 사용

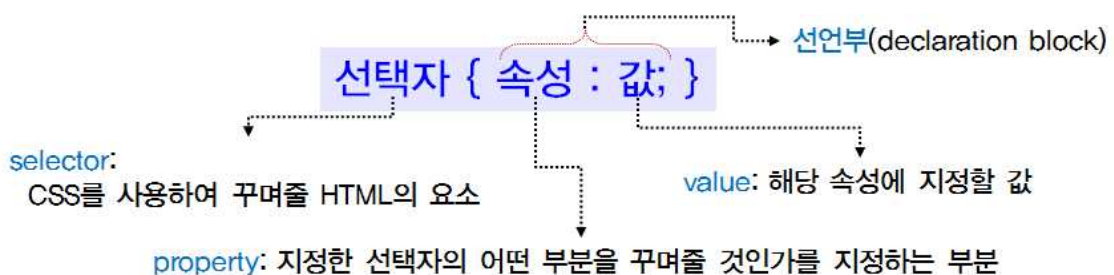
1.1 CSS ?

- CSS(Cascading Style Sheets)
 - ▶ 웹 문서를 만들기 위한 HTML 요소들이 문서를 만들기 위한 구조이고, CSS는 문서의 디자인 요소(여백, 배경색 등)를 담당한다.
 - ▶ CSS는 W3C에 의하여 표준화 작업이 진행되고 있으며, 현재는 CSS3(CSS Level 3)를 HTML5에서 기본으로 사용하고 있다.
- CSS의 레벨(Level)관리
 - ▶ CSS는 레벨(Level) 관리를 채택하여 기술 발전에 맞추어 기능들을 보완 및 추가하는 등으로 표준화를 진행하고 있다.

1.2 CSS 사용의 장점

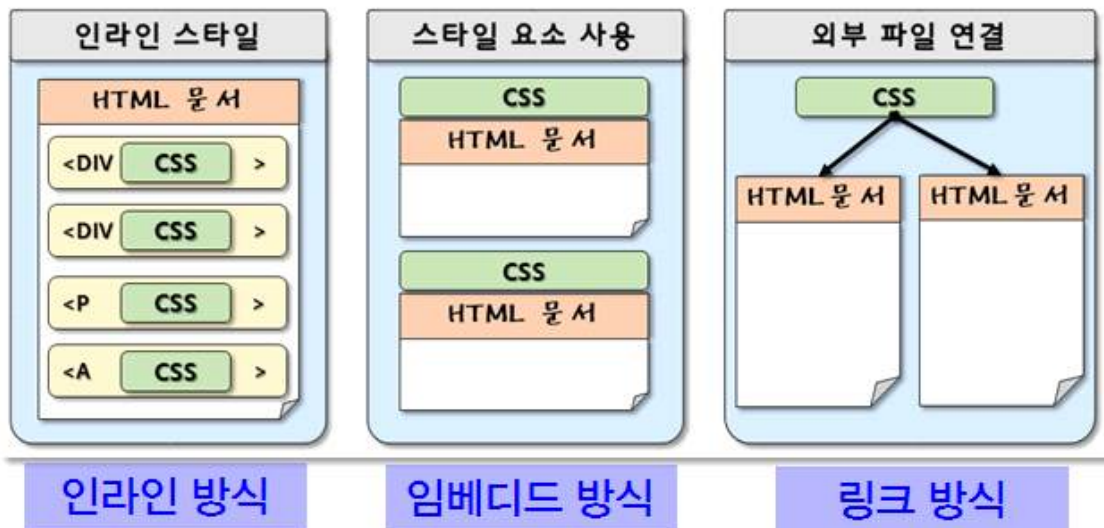
- 웹 표준에 의한 웹 문서는 내용과 디자인이 분리되어 있기 때문에 웹 문서의 내용을 수정해야 하는 경우, 디자인에 전혀 영향을 주지 않고 내용만 수정할 수 있거나, 내용에는 영향을 주지 않고 디자인을 수정하여 레이아웃을 변경하여 다른 느낌의 문서를 만들 수 있다.
- HTML문서는 단순한 기능만을 제공하기 때문에 CSS를 사용해야 다양한 기능을 추가하거나 변경시킬 수 있다. 다양한 글꼴의 사용이나 글자의 크기를 변경할 수 있고, 특정 요소를 원하는 모양으로 수정할 수도 있다.
- HTML 문서는 각 요소마다의 속성들을 하나하나 지정해 줘야 하지만, CSS를 사용하면 한 번의 속성을 정의함으로써 동시에 해당 요소에 속성들을 적용할 수 있다. 다른 문서에도 동일하게 적용할 수 있기 때문에, HTML 문서의 보수가 쉬어진다.
- 통일된 문서 양식 CSS를 사용함으로써 웹 브라우저가 읽어야 하는 문서의 내용도 작아지므로 웹 문서의 로딩 시간이 단축된다. 통일된 문서 양식이 한번 로딩이 되어 메모리에 적재되면, 이후에 이 CSS를 사용하는 HTML문서들이 로딩될 때는 문서의 양이 작기 때문에 로딩 시간이 단축된다.

1.3 스타일의 형식



예제	
<p>HTML 요소에서의 속성 표현</p> <pre><!DOCTYPE html> <html> <head> </head> <body> <p> Hello World! </p> </body> </html></pre>	<p>CSS에서의 속성 표현</p> <pre><!DOCTYPE html> <html> <head> <style> p { color: red; } </style> </head> <body> <p>Hello World!</p> </body> </html></pre>

1.4 CSS 적용방식



스타일이 정의되는 위치에 따른 구분

- 인라인 방식
 - HTML 문서의 각 태그 안에 직접 style 속성을 사용해서 적용하는 방법이다. 해당 요소에만 스타일을 적용하기 때문에 다른 동일한 요소에도 적용하려면 스타일 요소

를 사용하거나 외부 스타일로부터 스타일을 읽어 들여야 한다.

예제

```
<!DOCTYPE html>
<html>
  <head>
    <title> 인라인 스타일 </title>
  </head>
  <body>
    <h1 style="font-size:30pt; color:blue"> CSS 사용하기 </h1> ①
  </body>
</html>
```

① h1 요소에 font-size와 color 스타일을 적용하였다.

- 임베디드 방식
 - ▶ HTML 문서의 head 요소 안에 style 요소를 사용하여 스타일 적용하는 방법이다.

예제

```
<!DOCTYPE html>
<html>
  <head>
    <title> 스타일 요소 사용 </title>
    <style> ①
      h1 { font-size: 30pt; color: blue }
    </style>
  </head>
  <body>
    <h1> CSS 사용하기 </h1>
  </body>
</html>
```

① head 안에 h1(선택자)에 font-size와 color값을 지정하였다.

- 링크 방식
 - ▶ .css라는 확장자를 가지 파일에 스타일시트의 내용을 저장해서 HTML 파일에서 읽어들이는 방법이다. 이 방법은 여러 개의 HTML 파일에 동일한 스타일을 적용할 수

있는 장점이 있기 때문에 동일한 레이아웃을 유지하기 위한 방법으로 가장 많이 사용하고 있다.

예제

```
<!DOCTYPE html>
<html>
  <head>
    <title> 외부 파일 연결 </title>
    <link rel="stylesheet" href="sample.css"> ①
  </head>
  <body>
    <h1> CSS 사용하기 </h1>
  </body>
</html>
```

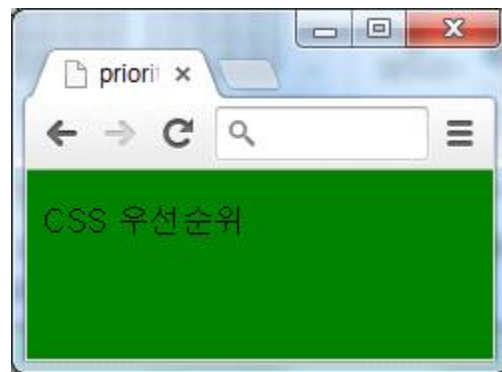
① sample.css (h1{font-size:30pt; color:blue;})파일을 연결시켜 스타일을 적용하였다.

1.5 CSS 적용의 우선순위

- 인라인 스타일 방법의 우선순위가 가장 높기 때문에 최우선적으로 적용되고, 다음으로 임베디드방식, 다음으로 링크 방식으로 적용된다.

예제

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css"> ①
    <style> ②
      body {background: red;}
    </style>
  </head>
  <body style="background: green">③
    <p>CSS 우선순위 </p>
  </body>
</html>
```



배경이 녹색으로 적용된 화면

③ 의 style이 html 백그라운드 배경으로 적용된다.

2. 단순 선택자

2.1 선택자

- 선택자는 조합자로 구분되는 하나 이상의 단순 선택자의 순서가 연결된 것이다. 그리고 하나의 가상요소는 단순 선택자의 마지막 순서에 추가될 수 있다. 여기서 단순 선택자의 순서는 조합자로 구분되지 않는 단순 선택자의 연결이다.
- CSS를 통해서 영향을 줄 것인지를 정의하는 선택자와 선택할 때 사용하는 선택자 또는 조합자 그리고 색상 속성, 텍스트 속성, 박스 속성 등이라 할 수 있다.
- 형식(type)선택자, 전체(*)선택자, 속성(attribute)선택자, 클래스(class)선택자, 아이디(id) 선택자 및 가상 클래스(Pseudo class)를 단순 선택자라고 한다.
- 공백(whitespace), 보다 크다 (>), 더하기 (+), 틸드(~)를 조합자라고 한다.
- 콤마로 구분되는 선택자 목록은 목록에 있는 개별 선택자 각각에 의해 선택된 모든 요소의 결합체를 나타낸다.

2.2 단순 선택자

- 기본 선택자의 종류는 다음과 같다.

Level	선택자	이름	설명
CSS2	*	전체universal선택자	모든 요소에 스타일 적용
CSS1	요소	형식type선택자	지정한 요소에 스타일 적용
CSS1	요소.클래스명	클래스 선택자	지정한 클래스에 스타일 적용
CSS1	요소#아이디명	아이디 선택자	지정한 아이디에 스타일 적용
CSS2	요소[속성]	속성 선택자	지정한 속성에 해당하는 요소에 적용

2.3 단순 선택자 – 전체 선택자

- 전체 선택자는 웹 문서에 있는 모든 요소들에 스타일을 적용시키는 것으로 모든 하위(자손)요소에 스타일을 적용시키고자 할 때 주로 사용한다.
- 사용법

```
[네임스페이스] | <*> { 속성: 값; 속성: 값; ... }
* { margin: 0; color: gray }
```

- *로 표시된 전체 선택자가 단순 선택자 순서의 유일한 구성 요소가 아니거나 바로 뒤에 가상 요소가 붙을 경우에는 별표 기호가 생략되고 전체 선택자의 존재가 암묵적으로 처리 된다.

```
* [hreflang|=en] = [hreflang|=en]
*.myclass = .myclass
```

- 네임스페이스 요소가 선택적으로 사용 가능하다.

네임스페이스 namespace

개체를 구분할 수 있는 범위를 나타내는 용어로서, 웹 문서에서 동일한 이름을 가진 요소들끼리 충돌을 방지하기 위해서 사용한다.

참고 사이트 CSS3 네임스페이스 모듈 <http://www.w3.org/TR/css3-namespace/>

```
@namespace ns "http://www.example.com"
```

```
@namespace ns url(http://www.example.com)
```

ns | * : 네임스페이스 ns 안에 있는 모든 요소들

* | * : 모든 요소들

| * : 네임스페이스가 없는 모든 요소들

* : 기본 네임스페이스가 지정되지 않은 경우는 * | *와 동일

ns가 기본 네임스페이스인 경우는 ns | *와 동일

2.4 단순 선택자 – 형식 선택자

- 스타일을 지정할 요소를 의미하는 것으로, HTML5 요소의 형식과 관련이 있기 때문에 형식 선택자라 부른다. HTML 요소에는 div, p, ul, table 등의 다양한 요소들이 존재하는데 이런 요소 이름에 따라 형식이 결정 되기 때문이다.
- 사용법

```
[네임스페이스] | <요소> { 속성: 값; 속성: 값; ... }
table { color: red }
ul { font-size: 18pt; font-style: italic }
```

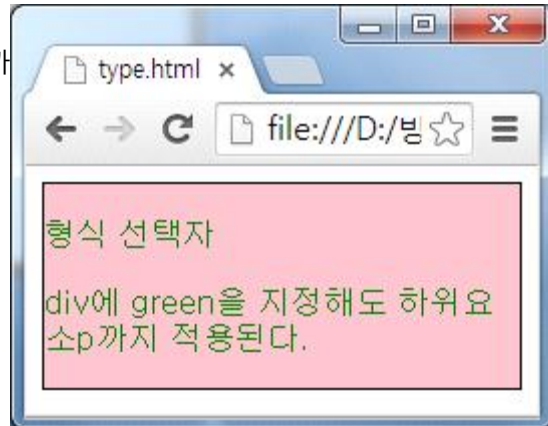
- 네임스페이스 요소가 선택적으로 사용 가능하다.

예제

```
<!DOCTYPE html>
<html>
<head>
<style>
div{ ① div 형식 선택자에 background, border, color를 지정한다.
background:pink; ② p요소에 상속된다.
border:1px solid black; ③ p요소에 상속되지 않는다.
color:green; ④ p 요소에 상속된다.
}
</style>
</head>
<body>
<div>
```

```

<p>형식 선택자</p>
<p>div에 green을 지정해도 하위요소까지 적용된다. </p>
</div>
</body>
</html>
    
```



결과화면으로 부모의 모든 스타일이 상속되는게 아니므로 주의해야 한다.

2.5 단순 선택자 – 클래스 선택자

- 같은 이름의 요소에 각각 다른 스타일을 적용하려면 각 요소들을 구별할 수 있는 클래스 선택자를 이용한다. 다른 요소에 동일한 스타일을 적용하는 경우에도 사용된다.
- 사용법

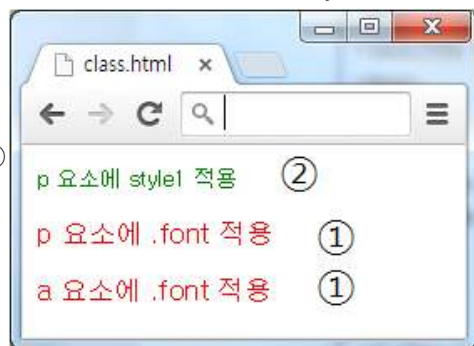
```

요소.클래스명 { 속성: 값; 속성: 값; ... }
.클래스명 {속성: 값; 속성: 값; ... }
p.fancy { color: red; font-weight: bold }
.select { font-size: 18pt; font-style: italic }
    
```

예제

```

<!DOCTYPE html>
<html>
<head>
<style>
.font { font-size:1em; color:red; } ① font 크기와 색 지정 (class="font1") 적용
.style1{ font-size:0.8em; color:green; } ② font 크기와 색 지정 (class="style1") 적용
</style>
</head>
<body>
<p class="style1">p 요소에 .style1 적용</p> ②
<p class="font"> p 요소에 .font 적용 </p> ①
<a class="font"> a 요소에 .font 적용 </a> ①
</body>
</html>
    
```



2.6 단순 선택자 – 아이디 선택자

- 특정 부분에 스타일을 적용하는데 문서에 두 개의 속성이 같은 값을 가질 수 없다. 문서 안에서 요소들을 배치(Layout) 시키는데 주로 사용한다.
- 사용법

```

요소#아이디명 { 속성: 값; 속성: 값; ... }
  #아이디명 {속성: 값; 속성: 값; ... }
p#font15 { font-size: 15pt }
  #font20 { font-size: 20pt }
    
```

예제

```

<!DOCTYPE html>
<html>
<head> <title>id test</title>
<style>
#font { font-size:1em; color:red;}①
#style1{ font-size:0.8em; color:green;}②
</style>
</head>
<body>
<p id="style1">p 요소에 #style1 적용</p>②
<p id="font"> p 요소에 #font 적용 </p>①
<a id="font"> a 요소에 #font 적용 </a>③
</body>
</html>
    
```



③브라우저에는 표시가 되나 HTML5 표준화 규정에 어긋나는 방법으로 다른 id값을 추가해 주어야 한다.

Markup Validation Service

다음 사이트(<http://validator.w3.org/>)는 Markup의 유효성을 검증하는 site이다. 상기 예제를 체크해 보면 동일한 id가 사용되었다는 오류를 표시하고 있다. 다음 사이트에서 HTML문서가 유효한지 검증해 볼 수 있다.



2.7 단순 선택자 - 속성 선택자

- 특정한 속성의 이름을 지정할 수 있는 것으로 대괄호 '[' 와 ']'로 묶여 있으며 대괄호 안에 찾고자 하는 속성 값을 지정하면 된다.
- 사용법

```
요소[네임스페이스]속성 연산자 값 { 속성: 값; ... }
p[title^="hello"]
```

- 하나의 요소에 복수의 속성 선택자가 사용가능하고 네임스페이스의 구성 요소를 선택적으로 허용하고 있다.
- 속성 선택자의 연산자 종류

- | | | |
|------|----------------|---|
| CSS2 | • 요소[속성] | ⇒ 속성 값과 상관없이 해당 속성을 사용하는 요소에 적용 |
| | • 요소[속성="값"] | ⇒ 속성 값과 일치하는 요소에 적용 |
| | • 요소[속성~="값"] | ⇒ 속성 값을 공백으로 구분한 요소에 적용 |
| | • 요소[속성 = "값"] | ⇒ 속성 값이 정확히 일치하거나 값으로 시작하고 바로 뒤에 - 기호로 구분한 요소에 적용 |
| CSS3 | • 요소[속성^="값"] | ⇒ 속성 값의 시작 부분의 문자와 일치하는 요소에 적용 |
| | • 요소[속성\$="값"] | ⇒ 속성 값의 끝 부분의 문자와 일치하는 요소에 적용 |
| | • 요소[속성*="값"] | ⇒ 속성 값 전체 중 어떤 일부분이라도 일치하는 요소에 적용 |

예제

```
<!DOCTYPE html>
<html>
<head>
<style>
a { font-family: "굴림체"; font-size:50px; } ① a요소에 font 서체와 크기를 지정한다.
a[title="daum"] { font-weight: bold; } ② title이 daum인 a요소
a[title~="cyworld"] { color: red; } ③ title이 공백 cyworld인 a요소
a[title="google"] { color: green; } ④ title이 google- 인 a요소
a[title^="face"] { font-style: italic; } ⑤ title이 face로 시작하는 a요소
a[title$="ter"] { font-size:40px; } ⑥ title이 ter로 끝나는 a요소
a[title*="utu"] { font-size:30px; } ⑦ title이 utu가 있는 a요소
</style>
</head>
<body>
<a title="naver" href="http://www.naver.com">네이버</a><br> ① 스타일이 적용
<a title="daum" href="http://www.daum.net">다음</a><br>②
<a title="nate cyworld" href="http://www.cyworld.com">싸이월드</a><br> ③
```

네이버
다음
싸이월드
구글 번역
페이스북
트위터
유튜브

```

<a title="google-translate" href="http://translate.google.co.kr">구글 번역</a> <br> ④
<a title="facebook" href="http://www.facebook.com">페이스북</a> <br> ⑤
<a title="twitter" href="http://www.twitter.com">트위터</a> <br> ⑥
<a title="youtube" href="http://kr.youtube.com">유튜브</a> ⑦
</body>
</html>

```

- table요소의 유효한 속성 들 중에서 직접 속성을 만들 수 있다. "data-"를 앞에 두기만 하면 된다.
- 사용법

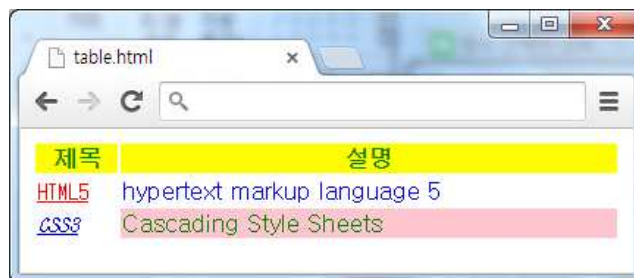
```
table[data-author^=John] { color: red }
```

예제

```

<!DOCTYPE html> <html><head>
<style>
table[data-made=LEE]{width:600px; color:green;} ① data-made=LEE인 table요소 적용
th{background:yellow; } ② th 요소에 적용
a[title~="HTML5"] { color: red; } ③ title=공백HTML5인 a 요소에 적용
a[title^="CSS"] { font-style: italic; } ④ title=CSS로 시작하는 a요소에 적용
td[title$="er"] { color:blue; } ⑤ title=er로 끝나는 td요소에 적용
td[title*="cad"] { background:pink; } ⑥ title=cad가 포함된 td요소에 적용
</style>
</head>
<body>
<table data-made="LEE"> ①
  <tr ><th>제목</th><th>설명</th></tr> ②
  <tr><td><a href="#" title="W3C HTML5">HTML5</a></td> ③
    <td title="hyper">hypertext markup language 5 </td> </tr> ⑤
  <tr> <td> <a href="#" title="CSS3">CSS3 </a></td> ④
    <td title="Cascading">Cascading Style Sheets </td> </tr>⑥
</table>
</body>
</html>

```



3. 가상 선택자

3.1 가상선택자

- 명시적으로 정의하지 않아도 가상으로 설정한 것처럼 동작하도록 하는 것이다. 즉 웹 문서상에는 존재하지 않지만 임의로 가상 선택자를 지정하여 사용하는 것을 의미한다.

3.2 가상클래스

- 문서 트리를 벗어나거나 단순 선택자를 활용하여 표현할 수 없는 정보를 토대로 한 선택이 가능하도록 한 것이 바로 가상 클래스의 개념으로, Pseudo-class 슈도 클래스라고 부른다. 가상 클래스의 이름은 대소문자를 구별하지 않고 항상 콜론(:) 뒤에 가상 클래스 이름과 괄호 사이에 값으로 이루어진다.

3.3 링크 가상 클래스, 사용자 조치 가상 클래스

- :link 가상 클래스는 방문한 적이 없는 링크로 적용되며, 기본적으로 파란색 글자와 밑줄로 표시되기 때문에, 밑줄을 없애거나 색상을 바꾸려고 할 때 사용한다. 방문했던 (:visited) 링크는 방문하지 않았던 (:link) 링크로 되돌릴 수 있다.
- :hover 가상클래스는 사용자가 포인팅 장치로 특정 요소를 지정하는 동안 적용이 되지만 반드시 활성화 되는 것은 아니다. 예를 들면 요소에 의하여 생성된 박스 위에 마우스 포인터를 올려놓으면 적용된다. 그러나 상호작용을 지원하지 않는 브라우저나 일부 브라우저는 이 가상 클래스를 지원하지 못할 수도 있다.
- :active 가상 클래스는 사용자에게 의하여 요소가 활성화되는 동안(사용자가 마우스를 눌렀다 놓는 동안) 적용된다.
- :focus 가상 클래스는 특정 요소에 초점이 있는 동안(텍스트 입력 양식에 입력했을 때, 텍스트 필드 안에 마우스 커서를 놓기 위해 Tab키를 눌렀을 때) 적용된다.

예제

```

<!DOCTYPE html> <html> <head>
<style> /* a:link ~ a:focus 순서 중요*/
a:link { color: green; } ① 방문한 적이 없는 링크
a:visited { color: blue; } ② 방문한 적이 있는 링크
a:hover { color: red; font-size: 18px } ③ 마우스가 링크 위로 오버되었을 경우
a:active { background: yellow } ④ 마우스로 클릭 상태(+ 마우스오버된 상태)
a:focus { background: yellow; color: black; font-weight: bold } ⑤ 포커스 있는 상태
a:focus:hover { background: blue; color: white } ⑥ 포커스 + 마우스 오버되었을 경우
</style>
</head>
<body>
<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> |

```



3.4 목적지 가상 클래스

- 동일한 문서 안에서 다른 위치로 이동하고자 할 때 이용하다.

예제

```

<!DOCTYPE html> <html><head>
<style>
p:target { background: yellow; border: 1px dotted red } ①
#sample1:target { background: blue; color: white } ②
</style>
</head>
<body>
<a href="#sample1">새로운 콘텐츠로 이동하기 1</a> | ① 클릭하면 ③ 으로 이동
<a href="#sample2">새로운 콘텐츠로 이동하기 2</a> ② 클릭하면 ④ 으로 이동
<p>위 링크를 클릭하면 아래 콘텐츠로 이동하면서 스타일이 변경됩니다.</p>
<p id="sample1"><b>새로운 콘텐츠1...</b></p> ③
<p id="sample2"><b>새로운 콘텐츠2...</b></p> ④
</body>
</html>
    
```

3.5 언어적 가상 클래스

- :lang 가상 클래스는 지정된 요소 중 language로 지정된 언어로 시작되는 요소를 선택하는 것이다.

3.6 사용자 인터페이스

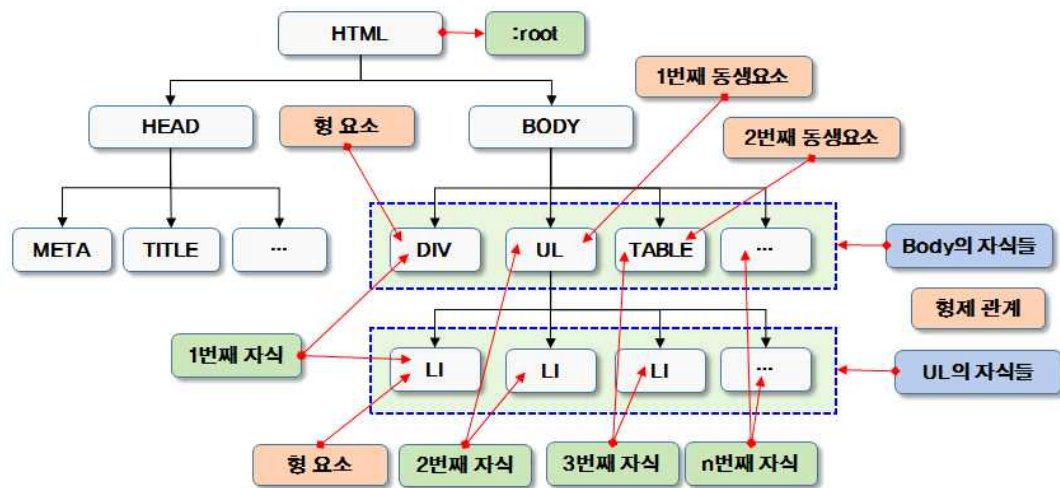
- :enable 가상 클래스는 요소 중에서 활성화된 인터페이스 요소를 나타낸다.
- :disabled 가상 클래스는 요소 중에서 활성화가 해제된 상태의 인터페이스 요소를 나타낸다.
- :checked 가상 클래스는 라디오 버튼이나 체크박스 요소를 클릭해서 상태를 반전(토글) 시킬 수 있다.

3.7 부정 가상 클래스

- :not(x) 가상 클래스는 단순 선택자(부정 가상 클래스 자체 제외)를 인자로 하는 기능적 표기로서, 인자로 전달된 단순 선택자 X로 표현되지 않는 특정 요소를 나타낸다. * :not(h3) 이면 h3요소를 제외하는 모든 요소를 나타낸다.

3.8 트리 구조적 가상 클래스

- 문서 트리 구조



- :root 가상 클래스는 웹 문서 안의 최상위 요소(루트)를 선택하여 스타일을 적용하는 것으로서, HTML 전체에 동일한 스타일 적용 가능하다.
- :nth-child(an+b) 가상 클래스는 동일한 부모 요소 안에서 n번째 있는 자식 요소를 선택하는 것이다.
- :nth-last-child(an+b) 가상 클래스는 끝에서 n번째에 있는 자식 요소를 선택하는 것이다.

예제
<pre><!DOCTYPE html> <html> <head></pre>

```

<style>
p:nth-child(odd) { background: blue;} ① 홀수 번째 = nth-child(2n+1)
p:nth-child(2n+0) { background: red;} ② 짝수 번째 = nth-child(2n+0)
p:nth-last-child(1){ font-style:italic;} ③ 끝에서부터 1첫번째
</style>
</head>
<body>
<p>nth-child(odd)</p>
<p>p:nth-child(even).</p>
<p>nth-child(odd) </p>
<p>p:nth-child(even)
p:nth-last-child(1)</p>
</body>
</html>

```



- :first-child 가상 클래스는 다른 일부 요소의 첫 번째 자식 요소를 나타내는 것으로, :nth-child(1)와 동일하다.
- :last-child 가상 클래스는 다른 일부 요소의 마지막 자식 요소를 자식인 특정 요소를 나타내기 때문에 :nth-last-child(1)과 동일하다.
- :nth-of-type(an+b) 가상 클래스는 동일한 부모 요소 안의 형제 요소들 중에서 지정한 요소만을 계산하여 n번째 요소를 선택한다. 특정한 위치의 지정한 요소를 선택할 때 용이하다.
- nth-of-last-type(an+b) 가상 클래스는 동일한 부모 요소 안의 형제 요소들 중에서 지정한 요소만을 계산하여 끝에서 n번째 요소를 선택한다.
- :first-of-type 가상 클래스는 동일한 부모 요소 안에서 형제 관계요소 중에서 첫 번째 형제 요소를 선택하고 :nth-of-type(1)과 동일하다.
- :last-of-type 가상 클래스는 동일한 부모 요소 안에서 형제 관계요소 중에서 마지막 형제 요소를 선택하고 :nth-last-of-type(1)과 동일하다.

예제

```

<!DOCTYPE html> <html> <head>
<style>
tr:nth-child(1) { color: white }
tr:nth-last-child(2) { color: red; font-style: italic }
tr:nth-child(2n+1) { background: orange }
tr:nth-child(even) { background: lightgreen }
tr:nth-child(-n+3) { font-size: 20px; font-weight: bold }
</style> </head>
<body>
<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>

```



- :only-child 가상 클래스는 부모 요소 안에서 자식 요소가 유일하게 하나일 때 선택한다. first-child = :last-child = :nth-child(1) = :nth-last-child(1)인 경우와 동일하다.
- :only-of-type 가상 클래스는 동일한 부모 요소 안의 형제 요소들 중에서 지정한 요소가 유일하게 하나일 때 선택하는 것으로 :first-of-type = :last-of-type = :nth-of-type(1) = :nth-of-last-type(1)인 경우와 동일하다.
- :empty 가상 클래스는 자식(텍스트가 아닌)이 없는 요소를 나타내는 것이다.

3.9 가상 요소

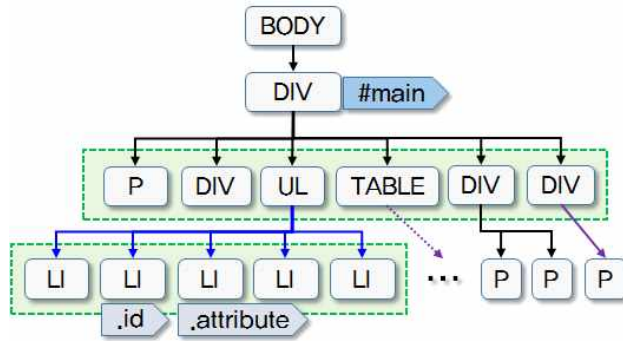
- 가상요소는 문서 언어에 의해 기록된 문서 트리에 관한 추상적 개념을 생성하는 것이다. 문서 언어는 요소의 내용에서 첫 번째 문자나 첫 번째 줄에 접근 하는 방법을 제공하지 않는다. 이렇게 가상 요소는 접근할 수 없는 정보를 참조할 수 있도록 다양한 수단을 제공하기 때문에 CSS에 없는 내용에 스타일을 지정할 수 있도록 해준다.
- ::first-line 가상요소는 어떤 문단에서 첫 번째 문자열에 일치하는 특정 요소에서만 사용할 수 있다. 페이지의 폭, 글꼴 크기 등의 다양한 요인에 의해 텍스트의 길이가 달라진다.
- ::first-letter 가상요소는 문자열에서 다른 내용(이미지, 인라인 테이블 등)이 앞에 오지 않는 경우 특정 요소의 첫 번째 글자를 선택한다. 첫 번째 글자는 반드시 첫 번째 줄의 문자열에서 나타나야 한다.
- ::before 가상요소는 요소 내용의 앞에 생성된 내용의 스타일과 위치를 지정한다.
- ::after 가상요소는 요소 내용의 뒤에 생성된 내용의 스타일과 위치를 지정한다.

4. 조합자

4.1 조합자

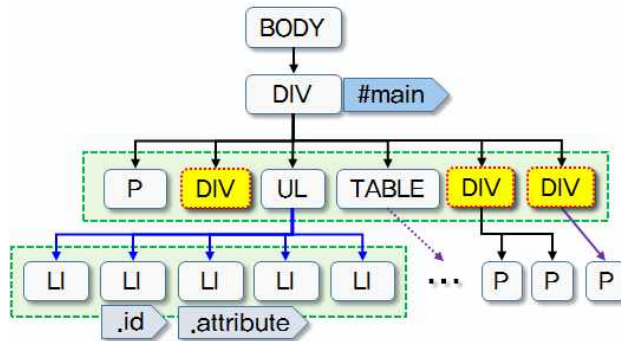
- 선택자를 문맥이나 구조를 기반으로 하여 조합하는 방법이 조합자이며 결합자라고도 부른다.

4.2 웹 문서의 트리 구조



4.3 하위 조합자: "선택자1 선택자2"

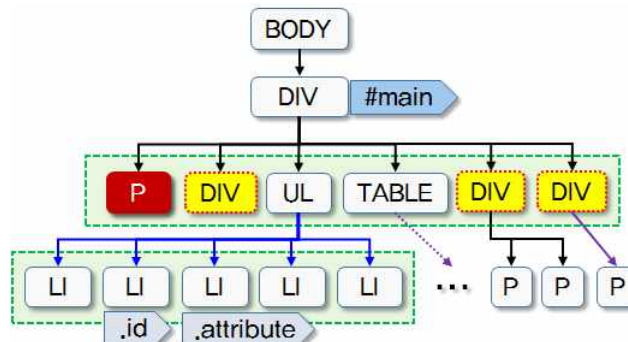
- 선택자와 선택자의 조합을 공백으로 연결하여 선언한다. 선택자1의 하위 요소 중에서 뒤의 선택자2에 해당하는 모든 요소를 선택하는 방법이다.



#main div { background: yellow; border: 3px dashed red }
 => #main의 하위 요소 중에서 모든 div를 선택한다.

4.4 자식 조합자: "선택자1 > 선택자2"

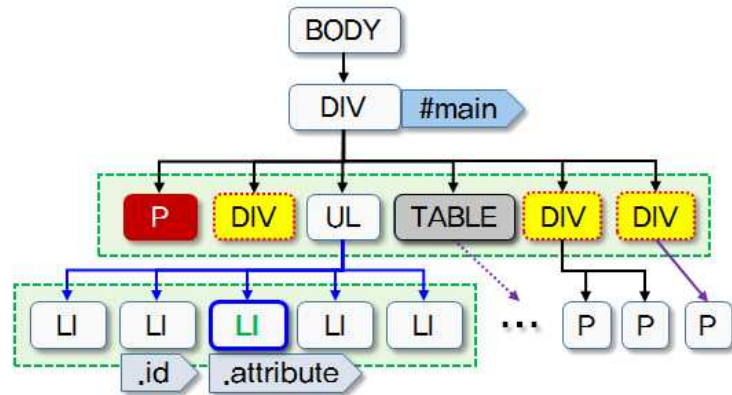
- 선택자와 선택자의 조합을 > 기호로 연결하여 선언한다. 선택자1을 부모 요소로 하여 포함된 하위 자식 선택자2 요소를 선택하는 방법이다. 하위 조합자와는 달리 부모 선택자의 1의 직계자식 선택자2 요소만 선택하는 방법이다.



#main > p { background: maroon; color: white }
 => #main요소의 직계 자식 P 선택

4.5 인접 형제 조합자: "선택자1 + 선택자2"

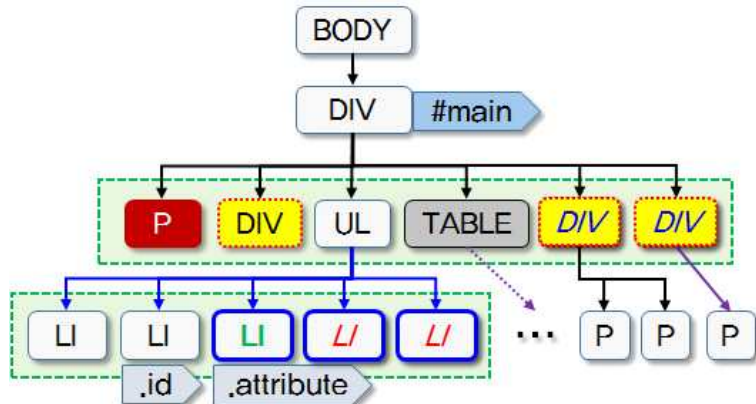
- 선택자와 선택자의 조합을 + 기호로 연결하여 선언한다. 동일한 부모 요소를 갖는 형제 관계인 선택자 1 바로 다음에 나타나는 동생 선택자2의 요소를 선택하는 방법이다.



ul + table { background: silver } => ul 옆의 TABLE 형제를 선택
 liid + li { color: green; font-weight: bold } => liid 옆의 li 형제를 선택

4.6 일반 형제 조합자: "선택자1 ~ 선택자2"

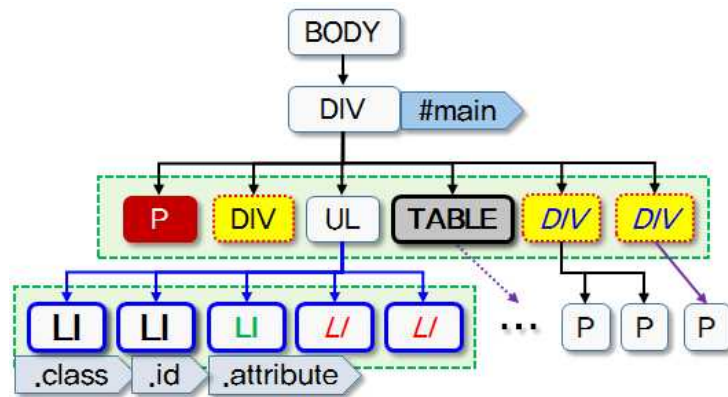
- 선택자와 선택자의 조합을 ~ 기호로 연결하여 선언한다. 선택자1 다음에 나타나는 선택자2 형제들을 선택하는 방법이다.



ul~div{color: blue; font-style: italic} => ul 다음에 나타나는 div 2개를 선택
 .attribute~li{color: red; font-style: italic } =>.attribute 다음의 li 2개를 선택

4.7 그룹 조합자: "선택자1, 선택자2, ..."

- 선택자와 선택자의 조합을 , 기호로 연결하여 선언하며 콤마 앞뒤에는 공백이 나타날 수 있다. 여러 개의 선택자가 동일한 선언을 공유하는 경우에는 한 번에 묶어서 스타일을 적용할 경우에 사용한다.



.class, .id, table { font-size: 20px; font-weight: bold } =>
TABLE 요소와, UL의 .class와 .id 에 동일한 스타일이 적용한
다.

4.8 선택자의 우선순위

