CSS (Cascading Style Sheets) #1

2014-07-23

이승진

# 소개

## 학습목표

HTML 태그의 서식 지정 기능인 CSS를 학습한다.

### 예제코드

프로젝트명: css01

유형: Dynamic Web Project

### 목차

[CSS (Cascading Style Sheets) 1](#_Toc393919874)

[1. 소개 1](#_Toc393919875)

[2. CSS (Cascading Style Sheet) 2](#_Toc393919876)

[3. CSS 실렉터 (selector) 3](#_Toc393919877)

[4. CSS 서식의 위치 8](#_Toc393919878)

[5. 서식 적용 우선 순위 규칙 10](#_Toc393919879)

[6. 서식 적용 예 13](#_Toc393919880)

[7. margin과 padding 속성 15](#_Toc393919881)

[8. display 속성 17](#_Toc393919882)

[9. table 19](#_Toc393919883)

[10. menu 22](#_Toc393919884)

# CSS (Cascading Style Sheet)

## 문서의 내용과 서식의 분리

HTML 문서의 구성 요소는 다음과 같다.

- HTML 태그

- CSS

- Javascript

### HTML 태그

HTML 태그로 문서의 내용을 작성한다.

문서에 포함된 자료(data)가 문서의 내용이다.

문서의 제목, 문단 내용, 표의 내용, 자료 목록 등이 문서의 내용이다.

문서의 내용이 그려지는 형태를 지정하는 서식(style)은 문서의 내용이 아니다.

### CSS

CSS는 HTML 태그가 그려지는 형태 즉 서식(style)을 지정한다.

문자의 크기와 색, 경계선의 굵기, 단락 사이의 여백, 표의 위치 등이 서식이다.

HTML 문서의 본문(body)에는 HTML 태그와 문서의 내용만 넣고,

문서의 서식은 본문과 분리해서 따로 CSS로 정의하는 것이 바람직하다.

### Javascript

Javascript 코드는 웹브라우저에서 실행된다.

HTML 문서가 웹브라우저에 로드되어 표시된 후에, 웹브라우저창에서 문서의 형태나 내용이 동적으로 변화해야 한다면 Javascript로 이를 구현해야 한다.

## style 태그

CSS 서식은 별도의 CSS 파일로 작성할 수 있고, HTML 문서 내부에 작성할 수도 있다.

HTML 문서 내부에 CSS 서식을 작성하려면 <style> 태그로 묶어주어야 한다.

<style> 태그는 HTML 문서에서 <head> 태그 사이에 위치하는 것이 보통이다.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <style>  *태그선택* {  *서식*;  *서식*;  *서식*;  ...  }  </style> |

(줄2) 서식을 적용할 HTML 태그를 선택한다.

서식을 적용할 태그를 선택하는 이 부분을 CSS 실렉터(selector)라고 부른다.

(줄3~6) 선택된 태그에 적용할 서식을 정의한다.

CSS 문법에서 줄 바꿈 위치에는 제약이 없다.

CSS 예

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <style>  h1 {  color : blue;  font-size: 12pt;  }  /\* 이것은 주석(comment) \*/  </style> |

(줄2) h1 → 서식을 적용할 html 태그를 선택하는 부분.

이 문서의 모든 h1 태그에 서식을 적용한다.

(줄3) color : blue; → 서식을 정의하는 부분.

이 문서의 모든 h1 태그 내부 문자의 색은 blue.

(줄4) font-size: 12pt; → 서식을 정의하는 부분. 글자 크기는 12pt

이 문서의 모든 h1 태그 내부 문자의 크기는 12pt

CSS 문법에서 주석은 C언어의 주석과 같이 /\* 주석 \*/ 형태이다.

# CSS 실렉터 (selector)

서식을 적용할 HTML 태그를 선택하는 부분을 CSS 실렉터(selector)라고 부른다.

서식을 적용할 HTML 태그를 선택하는 방법은 다음과 같다.

- 태그명으로

- 태그의 id 속성값으로

- 태그의 class 속성값으로

- 위 3방법을 결합하여

## CSS 실렉터의 예

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <style>  **h1** { color : blue; font-size: 12px; }  **#para1** { text-align: center; color: red; }  **.center** { text-align: center; }  **p#para2** { color: blue; }  **p.left** { text-align: left; }  </style> |

위에서 h1 #para1 .center p#para2 p.left 부분이 CSS 실렉터이다.

중괄호 { } 부분이 CSS 서식 정의이다.

CSS 실렉터로 선택된 HTML 태그들에 CSS 서식이 적용된다.

(줄2) 이 문서의 모든 h1 태그에 서식을 적용한다. 즉 여기서 h1은 서식을 적용할 태그명이다.

예: <h1>안녕하세요</h1>

(줄3) id="para1"인 모든 태그에 서식을 적용한다. 여기서 "#" 문자 뒤에 오는 것이 id 속성값이다.

예: <p id="para1">안녕하세요</p>

(줄4) class="center"인 모든 태그에 서식을 적용한다. 여기서 "." 문자 뒤에 오는 것은 class 속성값이다.

예: <div class="center">안녕하세요</div>

(줄5) id="para2"인 모든 p 태그에 서식을 적용한다. "#" 자 앞은 태그명이고 뒤는 id 속성값이다.

예: <p id="para2">안녕하세요</p>

(줄6) class="left"인 모든 p 태그에 서식을 적용한다. "." 문자 앞은 태그명이고 뒤는 class 속성값이다.

예: <p class="left">안녕하세요</p>

### css01/test01.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 01</title>  <style>  **h1** { color : blue; font-size: 20px; }  #para1 { text-align: center; color: red; }  .center { text-align: center; }  **p**#para2 { color: blue; }  **p**.right { text-align: right; }  </style>  </head>  <body>  <h1>파란색 12px 크기 제목.</h1>  <div id="para1">  빨간색 가운데 정렬. 빨간색 가운데 정렬. 빨간색 가운데 정렬. <br />  빨간색 가운데 정렬. 빨간색 가운데 정렬. <br />  </div>  <p id="para1">빨간색 가운데 정렬.</p>  <h1 class="center">가운데 정렬, 파란색 12px 크기 제목.</h1>  <p id="para2" class="center"> 파란색 문단. 파란색 문단. 파란색 문단.</p>  <p> 보통 문단. 보통 문단. 보통 문단.</p>  <p class="right"> 오른쪽 정렬 문단. 오른쪽 정렬 문단. 오른쪽 정렬 문단.</p>  <p id="para2" class="right"> 파란색 문단. 파란색 문단. 파란색 문단.</p>  <p> 보통 문단. 보통 문단. 보통 문단.</p>  <p class="right"> 오른쪽 정렬 문단. 오른쪽 정렬 문단. 오른쪽 정렬 문단.</p>  </body>  </html>  </body>  </html> |

(줄7)의 서식은 모든 h1 태그에 적용되므로 (줄16)과 (줄25)에 적용된다.

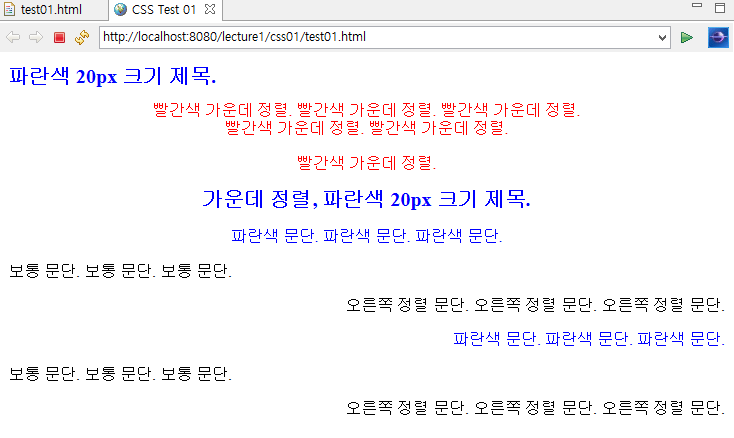
(줄8)의 서식은 (줄18~21)와 (줄23)에 적용된다.

(줄9)의 서식은 (줄25)과 (줄27)에 적용된다.

(줄10)의 서식은 (줄27)과 (줄31)에 적용된다.

(줄11)의 서식은 (줄29), (줄31), (줄33)에 적용된다.

### css01/test01.html 실행결과



## CSS 실렉터의 예

어떤 태그 내부에 있는 자식 태그들에 서식을 지정하는 형태의 구현도 가능하다.

아래의 예에서 살펴보자.

### css01/test02.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 02</title>  <style>  .div1 **h1** { color : blue; }  .div1 #para1 { color: red; border: solid 1px gray; padding: 10px; }  .center { text-align: center; }  **p** { border: solid 1px gray; padding: 10px; }  **p** **span** { font-size: 30px; }  **div** **p** **span** { color: green; }  **div** **span** { background-color: #FFFFAA; }  .div1 { margin-bottom: 90px; }  </style>  </head>  <body>  <div class="div1">  <h1>파란색 제목</h1>    <div id="para1">빨간색 텍스트</div>    <h1 class="center">가운데 정렬, 파란색 제목</h1>    <p><span>강조</span> 보통 텍스트</p>  </div>  <h1>보통 제목</h1>  <p><span>강조</span> 보통 텍스트</p>  </body>  </html> |

(줄7)은 .div1 태그 내부에 들어있는 h1 태그에 서식을 적용한다.

즉 class="div1" 속성을 가진 어떤 태그 내부에 들어있는 h1 태그들에 서식을 적용한다.

따라서 (줄20), (줄24)에 서식이 적용된다.

(줄8)는 class="div1" 속성을 가진 태그 내부의, id="para1" 속성을 가진 태그에 서식을 적용한다.

따라서 (줄22)에 서식이 적용된다.

(줄9)은 class="center" 속성을 가진 태그에 서식을 적용한다.

따라서 (줄24)에 서식이 적용된다.

(줄10)은 모든 p 태그에 서식이 적용된다. 따라서 (줄26), (줄31)에 서식이 적용된다.

(줄11)는 p 태그 내부의 span 태그인 (줄26), (줄31)에 서식이 적용된다.

(줄12)은 div 태그 내부의 p 태그 내부의 span 태그인 (줄26)에 서식이 적용된다.

(줄31)의 span 태그는 div 내부가 아니므로 (줄12)의 서식 적용 대상이 아니다.

(줄13)는 div 태그 내부의 span 태그인 (줄26)에 서식이 적용된다.

span 태그가 div 태그의 바래 아래 자식은 아니고 중간에 p 태그가 있지만 상관없이 서식 적용 대상이다.

(줄31)의 span 태그는 div 내부가 아니므로 (줄13)의 서식 적용 대상이 아니다.

(줄14)는 (줄19~27)의 div 태그에 서식을 적용한다. 이 div 태그와 그 바로 아래인 (줄29)의 h1 태그 사이에 90 픽셀의 여백이 만들어진다.

### css01/test02.html 실행결과



# CSS 서식의 위치

CSS 서식은 아래의 3곳에 위치할 수 있다.

- 확장자가 css인 별도의 파일에 CSS 서식을 정의한다.

- html 문서의 내부에 <style> 태그에 CSS 서식을 정의한다.

- html 태그의 style 속성값에 CSS 서식을 정의한다.

위 3가지 방법을 아래의 예에서 살펴보자.

### css01/test03.css

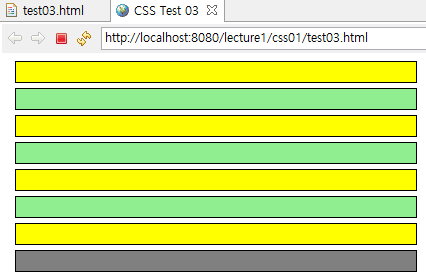
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **div** {  background-color: red;  height: 20px;  width: 400px;  border: solid 1px black;  margin: 5px;  } |

확장자가 css인 파일에 서식을 정의할 때는 <style> 태그가 필요 없다.

### css01/test03.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 03</title>  <link rel="stylesheet" type="text/css" href="test03.css">  <style>  **div** { background-color: yellow; }  **div**.even { background-color: lightgreen; }  </style>  </head>  <body>  <div></div>  <div class="even"></div>  <div></div>  <div class="even"></div>  <div></div>  <div class="even"></div>  <div></div>  <div class="even" style="background-color: gray;"></div>  </body>  </html> |

### css01/test03.html 실행결과



## 외부 서식 파일

확장자가 css인 파일에 서식을 정의하여, css 서식과 html 태그를 별도의 파일로 분리한다.

css 파일에 정의된 서식이 적용되려면 html 파일은 이 css 파일에 대한 링크 태그가 있어야 한다.

css01/test03.html 파일 (줄6)의 link 태그에 의해서 test03.css 파일에 정의된 서식이 이 html 파일에 적용된다.

<link rel="stylesheet" type="text/css" href="test03.css">

link 태그의 href 속성은 css 파일에 대한 URL 이다.

css 파일과 html 파일이 같은 폴더에 있다면 link 태그의 href 속성값으로 css 파일명만 적어주면 된다.

css 파일이 html 파일과 같은 폴더에 있지 않다면 URL을 적어야 한다.

## URL의 형태

URL을 적어주는 방식을 상대 경로와 절대 경로로 구분할 수 있다.

상대 경로란 현재 위치에서 시작하여 어디에 있는 어떤 파일이라는 형태를 말한다.

예)

"css01/test03.css" → 현재 폴더의 아래 "01" 폴더에 있는 "test03.css"

"../css01/test03.css" → 현재 폴더의 위 폴더의 아래 01 폴더에 있는 "test03.css"

"../../css01/test03.css" → 현재 폴더의 위 폴더의 위 폴더의 아래 01 폴더에 있는 "test03.css"

절대 경로 URL은 "http://서버주소" 로 시작하거나 "/컨택스트명" 으로 시작해야 한다.

예)

"http://localhost/lecture1/css01/test03.css"

"/css01/01/test03.css"

"lecture1"은 이클립스에서는 웹프로젝트명이고, 톰캣 웹서버에서는 컨택스트(context)명이다.

## 내부 서식

css01/test03.html 파일 내부 (줄7)~(줄10)의 <style> 태그에 서식이 정의되어 있다.

<style> 태그로 서식을 작성하는 방법의 장점은, html 태그와 css 서식을 같은 파일에 작성하기 때문에 처음 개발할 때 간편하다는 점이다.

별도의 css 파일에 서식을 따로 정의하는 방법의 장점은, 웹 페이지의 디자인을 개편하고 싶을 때 css 파일만 수정하거나 교체하면 된다는 점이다. 즉 웹사이트 스킨이나 테마를 여러 개 만들어 두고 바꿔가며 적용하려면 별도의 css 파일로 서식을 정의하는 쪽이 간편하다.

## 인라인 서식 (inline style)

css01/test03.html에서 (줄21)의 div 태그에는 style 속성으로 서식이 정의되어 있다.

이렇게 html 태그의 속성으로 서식을 정의하는 것은 바람직하지 않다.

문서의 내용에 해당하는 HTML 태그 부분과, 서식 정의 부분은 분리되는 것이 좋다.

# 서식 적용 우선 순위 규칙

여기 저기에 서식이 정의되고 이들 서식이 중복된다면 우선 순위가 중요하다.

서식 정의가 중복된다면 우선 순위가 높은 것이 적용되고 낮은 것은 무시된다.

## 규칙1 - 인라인 서식 정의

css01/test03.html 파일의 (줄21) 인라인 서식 정의의 우선순위가 그 무엇보다 가장 높다.

html 태그에 style 속성으로 정의된 서식은 다른 어떤 서식 보다도 가장 높은 우선순위를 갖는다.

따라서 css01/test03.html 파일 (줄12) div 태그의 background-color 속성은 다른 서식이 어떻게 정의되어 있더라도 상관없이 무조건 gray이다.

## 규칙2 - 상세한 실렉터

그 다음으로 우선시 되는 규칙은 상세하게 정의된 css 실렉터(selector)의 우선 순위가 높다는 것.

우선 순위가 높은 상세한 실렉터의 형태부터 시작하여 순서대로 나열하면 다음과 같다.

### 실렉터의 형태에 따른 우선순위 표

|  |  |  |
| --- | --- | --- |
| **등수** | **실렉터의 형태** | **예** |
| 1 | 태그명과 id 속성값을 명시 | div#a |
| 2 | id 속성값만 명시 | #b |
| 3 | 태그명과 class 속성값을 명시 | div.c |
| 4 | class 속성값만 명시 | .d |
| 5 | 태그명만 명시 | div |

css 서식 정의가 중복되는 사례를 살펴보자.

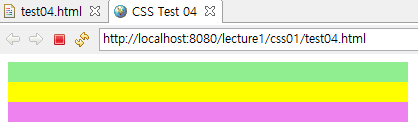
### css01/test04.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 04</title>  <style>  div#a { background-color: lightgreen; }  #b { background-color: yellow; }  div.c { background-color: red; }  .d { background-color: violet; }  div { background-color: blue; width: 400px; height: 20px; }  </style>  </head>  <body>  <div id="a" class="c"></div>  <div id="b" class="c"></div>  <div class="d"></div>  </body>  </html> |

(줄7~11)에서 background-color 서식이 중복되어있다.

(줄16~18)의 div 태그들은 각각 어떤 background-color 서식을 적용 받을까?

### 실행결과



css01/test04.html에서 (줄7~11) 순서가 실렉터의 형태에 따른 우선순위 표의 순서와 같다.

즉 (줄7)의 우선순위가 가장 높고, (줄11)의 우선순위가 가장 낮다.

(줄11)의 서식은 모든 div 태그에 다 적용된다. background-color 속성은 다른 서식과 중복되기 때문에 우선순위를 따져봐야 하지만, 중복되지 않는 width와 height 속성값은 모든 div 태그에 그대로 적용된다.

그래서 위 실행결과에서 3개의 사각형의 폭과 높이가 모두 같다.

(줄16)의 div 태그에는 (줄7), (줄9), (줄11)의 서식이 적용된다.

div 태그이므로 (줄11)의 적용 대상이고, class="c" 이므로 (줄9)의 적용 대상이고,

id="a" 이므로 (줄7)의 적용 대상이다.

background-color는 이 3줄에 다 정의되어 있어서 중복되는데, 이들 중 가장 우선순위가 높은 (줄7)의 lightgreen 속성값이 적용된다. 그래서 실행결과에서 첫 사각형은 lightgreen 색이다.

(줄17)의 div 태그에는 (줄8), (줄9), (줄11)의 서식이 적용된다.

이들 중에서 우선순위가 높은 (줄8)의 yellow 색이 background-color 속성에 적용된다.

(줄18)의 div 태그에는 (줄10), (줄11)의 세 서식이 적용된다.

이들 중에서 우선순위가 높은 (줄10)의 violet 색이 background-color 속성에 적용된다.

### 부모 태그를 명시한 형태의 실렉터가 높은 우선순위를 갖는다.

부모 태그 내부에 있는 자식 태그를 지정하는 형태의 실렉터는, 부모 태그를 명시하지 않는 실렉터보다 상세하므로 우선 순위가 높다.

예를들어 규칙2의 우선순위 표의 실렉터들 앞에 부모 태그 실렉터를 적어주고 우선순위를 비교하면

다음과 같다.

부모 태그 실렉터 유무에 다른 우선순위 표

|  |  |  |  |
| --- | --- | --- | --- |
| **등수** | **부모 태그 실렉터** | **자식 태그 실렉터 형태** | **예** |
| 1 | O | 태그명과 id 속성값을 명시 | div div#a |
| 2 | id 속성값만 명시 | div #b |
| 3 | 태그명과 class 속성값을 명시 | div div.c |
| 4 | class 속성값만 명시 | div .d |
| 5 | 태그명만 명시 | div div |
| 6 | X | 태그명과 id 속성값을 명시 | div#a |
| 7 | id 속성값만 명시 | #b |
| 8 | 태그명과 class 속성값을 명시 | div.c |
| 9 | class 속성값만 명시 | .d |
| 10 | 태그명만 명시 | div |

부모 태그 실렉터의 형태에 따라 우선순위가 다르다.

당연히 부모 태그 실렉터가 상세할 수록 우선순위가 높다.

우선순위를 결정할 때 자식 태그 실렉터의 상세함보다 부모 태그 실렉터의 상세함이 더 중요하다.

|  |  |  |
| --- | --- | --- |
| **등수** | **부모 태그 실렉터 형태** | **예** |
| 1 | 태그명과 id 속성값을 명시 | div#a span |
| 2 | id 속성값만 명시 | #a span |
| 3 | 태그명과 class 속성값을 명시 | div.c span |
| 4 | class 속성값만 명시 | .c span |
| 5 | 태그명만 명시 | div span |

## 규칙3 - 서식 정의 위치

규칙1, 규칙2에 의한 우선순위가 동일하다면, 그 다음 우선순위 규칙은 서식 정의 위치이다.

서식 정의 위치에 따라 우선순위가 다르다.

서식 정의 위치에 따른 우선순위 규칙은 다음과 같다

- html 파일 내부의 서식 정의가 외부 서식 파일보다 우선순위가 높다.

- 파일 위치가 동일하다면 마지막으로 줄번호를 비교한다. 아래 줄의 서식 정의가 윗 줄보다 우선순위가 높다.

css01/test03.html 예제를 다시 자세히 살펴보자.

### css01/test03.css

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **div** {  background-color: red;  height: 20px;  width: 400px;  border: solid 1px black;  margin: 5px;  } |

### css01/test03.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 03</title>  <link rel="stylesheet" type="text/css" href="test03.css">  <style>  **div** { background-color: yellow; }  **div**.even { background-color: lightgreen; }  </style>  </head>  <body>  <div></div>  <div class="even"></div>  <div></div>  <div class="even"></div>  <div></div>  <div class="even"></div>  <div></div>  <div class="even" style="background-color: gray;"></div>  </body>  </html> |

(줄14)의 div 태그에는 test03.css (줄2), test03.html (줄8)의 서식이 둘 다 적용된다.

이 두 서식의 실렉터의 상세함이 태그명만 적어준 형태로 동일하므로, 그 다음 기준인 서식 정의 위치를 비교해서 우선순위를 결정하게 된다.

외부 파일인 test03.css보다 같은 파일인 test03.html 파일의 서식 정의 우선순위가 높기 때문에

(줄14)의 background-color는 "yellow"가 된다.

(줄15)의 div 태그는 test03.html (줄9)의 우선순위가 가장 높기 때문에 배경색이 lightgreen이 된다.

(줄21)의 div 태그는 우선순위가 그 무엇보다도 높은 인라인 서식으로 배경색을 지정했으므로, 배경색은 무조건

gray가 된다.

# CSS 예제

## css01/test05.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 05</title>  <style>  body > div { border: solid 1px black; background-color: yellow; }  div div { padding: 30px; }  div.c1 { padding: 20px; }  div.c1 div { background-color: #aaffaa; }  div.c2 {  margin-top: 20px;  margin-bottom: 20px;  }  div.c2 div {  margin: 20px;  background-color: #ffaaaa;  }  div.c3 { width: 500px; }  div.c3 div {  margin: 20px;  background-color: #dddddd;  }  </style>  </head>  <body>  <div class="c1">  <div>  내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용  내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용  </div>  </div>  <div class="c2">  <div>  내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용  내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용  </div>  </div>  <div class="c3">  <div>  내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용  내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용 내용  </div>  </div>  </body>  </html> |

### padding과 margin

이 두 css 속성은 여백의 크기를 지정한다.

태그의 경계선(border)을 기준으로, padding은 태그의 안쪽 여백이고, margin은 태그의 바깥쪽 여백이다.

즉 padding은 태그의 경계선과 그 태그 안쪽 내용 사이의 여백이고, margin은 태그 외부의 다른 태그 사이의 여백이다.

### 실행결과



(줄7) body > div → body 태그의 자식이고 손자나 증손자가 아닌 div 태그가 선택된다.

(줄28), (줄35), (줄42)의 div 태그에 서식이 적용된다.

이 세 div 태그들에 검정 실선 경계선과 노란색 배경이 적용된다.

body div → body 태그 내부의 모든 div 태그가 전부 선택된다.

(줄28), (줄29), (줄35), (줄36), (줄42), (줄43) div 태그들이 다 선택된다.

(줄8) div div → div 태그의 내부에 들어있는 div 태그가 선택된다.

(줄29), (줄36), (줄43)의 div 태그에 서식이 적용된다.

노란색 div들의 안쪽 div들은 내부 내용과의 사이에 30 픽셀 여백이 적용된다.

위 실행결과 그림에서 "내용 내용 내용..." 문자열과 그 주위 사각형과 사이의 여백이 30 픽셀이다.

(줄9) div.c1 → (줄28)의 사각형에 20 픽셀 padding이 적용된다.

위 실행결과 그림에서 첫번째 노란색 사각형의 노란색 여백이 이 padding에 해당한다.

(줄10) div.c1 div → (줄29)의 div 태그가 선택된다. 이 div의 배경색이 밝은 녹색이 된다.

(줄11) div.c2 → (줄35)의 div 태그가 선택된다.

(줄12) 실행결과 그림에서 첫번째 노란 사각형과 두번째 노란 사각형 사이의 여백이 20px이다.

(줄13) 실행결과 그림에서 두번째 노란 사각형과 세번째 노란 사각형 사이의 여백이 20px이다.

(줄15) div.2 div → (줄36)의 div 태그가 선택된다.

(줄16) 실행결과 그림에서 두번째 노란 사각형의 노란색 여백이 이 margin에 해당한다.

(줄17) 실행결과 그림에서 두번째 노란 사각형 내부의 사각형이 분홍색이 된다.

(줄19) div3.c3 → (줄42)의 div 태그가 선택된다. 이 태그의 가로 폭이 500 픽셀이 된다.

div 태그의 디폴트 width 값은 100%이어서 전체 영역을 다 채운다.

(줄20) div.c3 div → (줄43)의 div 태그가 선택된다.

(줄21) 실행결과 그림에서 세번째 노란 사각형의 노란색 여백이 이 margin에 해당한다.

## block 태그, inline 태그, inline-block 태그

### block 태그

block 태그들의 디폴트 width 값은 100%이어서 혼자 가로 폭을 다 차지한다.

그렇기 때문에 block 태그 앞뒤로 줄바꿈이 일어나서, block 태그 좌우에 다른 태그가 없다.

black 태그들은 margin, padding, width, height 속성값이 잘 적용된다.

예: div, table, form, p, ul, ol, h1

### inline 태그

inline 태그들의 width 값은 그 내용물의 width 이다. 그리고 좌우의 다른 태그들과 붙어있다.

inline 태그들은 padding과 좌우 margin 속성값은 적용되는데,

위아래 margin 속성값과 width, height 속성값은 상황에 따라 무시될 수 있다.

예: input, span, font, b, label, a

### inline-block 태그

inline-block 태그들은,

block 태그처럼 padding, margin, width, height 속성이 잘 적용되고,

inline 태그처럼 좌우의 다른 태그들과 붙어있다.

### display CSS 속성

html 태그에는 display CSS 속성이 있다.

이 속성의 값은 block, inline, inline-block, none 이다.

이 속서값이 none이면 웹브라우저 창에 그려지지 않는다.

block 태그/inline태그/inline-block 태그의 display 속성값은 block/inline/inline-block 이다.

태그의 이 속성값을 다음과 같이 수정할 수 있다.

|  |  |
| --- | --- |
| 1  2  3 | div { display: inline-block; }  span { display: block; }  div#error { display: none; } |

(줄1) div 태그의 display 속성의 디폴트 값은 block 이지만, 이 문서의 모든 div 태그는 inline-block이 된다.

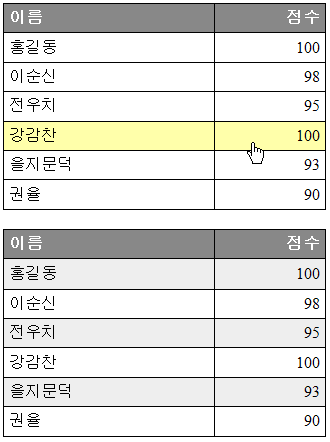
(줄2) span 태그의 display 속성의 디폴트 값은 inline 이지만, 이 문서의 모든 span 태그는 block이 된다.

(줄3) id="error" 속성을 가진 div 태그는 화면 그려지지 않고 숨겨진다. 나중에 javascript 코드로 display 속성값을 block으로 수정하면 화면에 그려진다.

## css01/test06.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 06</title>  <style>  table { border-collapse:collapse; }  td { border:1px solid black; padding: 5px; }    tr:first-child { background-color: #888888; color: white; font-weight: bold; }    td:nth-child(1) { width: 200px; }  td:nth-child(2) { width: 100px; text-align: right; }    table.t1 tr:hover { background-color: #ffffaa; cursor: pointer; }  table.t2 tr:hover:not(:first-child) { background-color: #ffffaa; cursor: pointer; }  table.t2 tr:nth-child(even) { background-color: #eeeeee; }  table.t2 { margin-top: 20px; }  </style>  </head>  <body>  <table class="t1">  <tr><td>이름</td><td>점수</td></tr>  <tr><td>홍길동</td><td>100</td></tr>  <tr><td>이순신</td><td>98</td></tr>  <tr><td>전우치</td><td>95</td></tr>  <tr><td>강감찬</td><td>100</td></tr>  <tr><td>을지문덕</td><td>93</td></tr>  <tr><td>권율</td><td>90</td></tr>  </table>  <table class="t2">  <tr><td>이름</td><td>점수</td></tr>  <tr><td>홍길동</td><td>100</td></tr>  <tr><td>이순신</td><td>98</td></tr>  <tr><td>전우치</td><td>95</td></tr>  <tr><td>강감찬</td><td>100</td></tr>  <tr><td>을지문덕</td><td>93</td></tr>  <tr><td>권율</td><td>90</td></tr>  </table>  </body>  </html> |

### 실행결과



(줄7) 테이블의 경계선을 한줄로 표시한다

(줄8) 테이블 각 칸에 1px 굵기의 검정색 경계선을 그린다.

(줄10) tr:first-child → tr 태그들 중에서 부모로부터 첫번째 자식인 tr 태그만 선택한다.

테이블의 첫 행에만 이 서식이 적용되어, 회색 배경에 흰색 굵은 글씨로 그려진다.

(줄11) td:nth-child(1) → td 태그들 중에서 부모로부터 첫번째 자식인 td 태그만 선택한다.

테이블의 첫 열에만 이 서식이 적용되어 가로 폭이 200 픽셀이 된다.

:first-child 실렉터와 :nth-child(1) 실렉터는 둘 다 부모로부터 첫번째 자식인 태그만 선택한다. 둘이 동일해서 아무런 차이 없다.

(줄15) table.t1 tr:hover → class="t1" 속성의 table 태그 내부의 tr 태그들 중에서   
 현재 마우스 포인터가 그 위에 있는 태그만 선택한다.

마우스 포인터가 위치한 행의 background-color는 밝은 노란색이고

마우스 포인터는 손가락(pointer) 아이콘으로 표시된다.

(줄16) table.t2 tr:hover:not(:first-child) → class="t2" 속성의 table 태그 내부의 tr 태그들 중에서   
 현재 마우스 포인터가 그 위에 있는 태그만 선택한다.

단 부모로부터 첫번째 자식인 태그는 제외한다.

(줄17) table.t2 tr:nth-child(even) → class="t2" 속성의 table 태그 내부의 tr 태그들 중에서   
 부모로부터 짝수번재 자식인 tr 태그만 선택하다.

실행 결과에서 두 번째 테이블의 짝수 행의 배경색이 밝은 회색이다.

(줄18) 실행 결과에서 두 테이블 사이에 20 픽셀 여백이 생긴다.

## 실렉터 목록

앞의 예제에 새로 등장한 실렉터들을 정리하자

### :first-child 실렉터

부모 태그로부터 첫번째 자식인 태그들만 선택한다.

### :nth-child(*n*) 실렉터

부모 태그로부터 *n*번째 자식인 태그들만 선택한다.

### :nth-child(even) 실렉터

부모 태그로부터 짝수번째 자식인 태그들만 선택한다.

### :nth-child(odd) 실렉터

부모 태그로부터 홀수번째 자식인 태그들만 선택한다.

### :hover 실렉터

현재 마우스 포인터가 그 태그 위에 있는 태그만 선택한다.

마우스 위치는 실시간으로 이동하기 때문에 이 실렉터가 선택하는 태그도 실시간으로 달라진다.

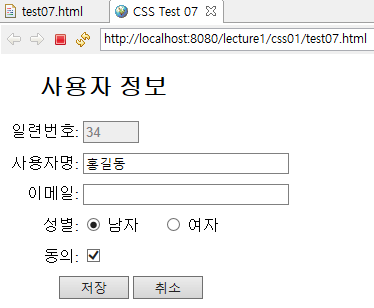
### :not(*실렉터*)

괄호안에 주어진 실렉터에 해당하지 않는 태그들만 선택한다

## css01/test07.html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <title>CSS Test 07</title>  <style>  body { font-family: 굴림; font-size: 10pt: }  #form1 h2 { margin-left: 30px; }  #form1 div { margin-bottom: 10px; }  #form1 label:first-child { display: inline-block; width: 70px; text-align: right; }  #form1 label:not(:first-child) { display: inline-block; margin-right: 20px; }  input[type=text] { border: solid 1px gray; padding: 2px; width: 200px; }  input[type=text][readonly] { background-color: #eeeeee; color:gray; width:50px }  input[type=text]:focus{ background-color: #eeffee; }  input[type=submit] { margin-left: 50px; width: 70px; }  input[type=reset] { width: 70px; }  </style>  </head>  <body>  <form id="form1" method="post">  <h2>사용자 정보</h2>  <div>  <label>일련번호:</label>  <input type="text" name="no" readonly value="34" />  </div>  <div>  <label>사용자명:</label>  <input type="text" name="username" value="홍길동" />  </div>  <div>  <label>이메일:</label>  <input type="text" name="email" />  </div>  <div>  <label>성별:</label>    <input id="r1" type="radio" name="sex" value="male" checked />  <label for="r1">남자</label>    <input id="r2" type="radio" name="sex" value="female" />  <label for="r2">여자</label>  </div>  <div>  <label for="c1">동의:</label>  <input id="c1" type="checkbox" name="check" checked />  </div>  <div>  <input type="submit" value="저장" />  <input type="reset" value="취소" />  </div>  </form>  </body>  </html> |

### 실행결과



(줄7) body 태그에 폰트를 설정한다. 폰트 서식 설정은 body 태그 내부의 자식 태그들에도 적용된다.

따로 폰트 설정을 하지 않은 태그는 부모 태그의 폰트 설정을 적용 받는다.

(줄8) id="form1"인 태그 내부의 h2 태그를 선택한다. (줄22)의 h2 태그에 왼쪽 여백 30 픽셀이 적용된다.

(줄9) id="form1"인 태그 내부의 div 태그를 선택한다. (줄24,29,34, 49, 49, 54) div 태그들이 선택된다.

이 div 태그의 아래쪽 여백 10 픽셀이 적용된다.

(줄10) id="form1"인 태그 내부의 label 태그들 중에서, 부모로부터 첫번째 자식인 label 태그를 선택한다.

이 실렉터에 의해서 (줄25, 30, 35, 40, 50) label 태그들은 선택되지만, (줄43, 46) label 태그들은 1번 자식이 아니고 3, 5번 자식이기 때문에 선택되지 않는다. 디폴트 display 속성값이 inline인 label 태그는 width 속성 설정을 무시한다. width 속성 설정이 무시되지 않도록, label 태그의 display 속성값을 inline-block으로 수정했다.

(줄11) id="form1"인 태그 내부의 label 태그들 중에서, 부모로부터 첫번째 자식이 아닌 label 태그를 선택한다.

이 실렉터에 의해서 (줄43, 46) label 태그들이 선택된다.

(줄12) type="text" 속성을 가진 input 태그를 선택한다. (줄26, 31, 36) input 태그들이 선택된다.

(줄13) type="text" 속성과 readonly 속성을 가진 input 태그를 선택한다. (줄26) input 태그가 선택된다.

(줄14) type="text" 속성을 가진 input 태그 중에서 현재 입력 포커스(focus)를 가진 태그를 선택한다.

입력 포커스는 실시간으로 변화하기 때문에 이 서식 적용 대상 태그도 실시간으로 달라진다.

(줄15) type="submit" 속성을 가진 input 태그를 선택한다. (줄55) input 태그가 선택된다.

(줄16) type="submit" 속성을 가진 input 태그를 선택한다. (줄56) input 태그가 선택된다.

## 실렉터 목록

### :focus

현재 입력 포커스를 소유하는 태그를 선택하는 실렉터이다. 텍스트 박스에 문자를 입력하고 있다면, 그 텍스트 박스가 입력 포커스를 소유하고 있다. 탭 키를 누르면 다음 태그로 입력 포커스가 넘어간다.

### [속성명=값]

태그의 속성명과 그 값으로 선택하는 실렉터이다. 주의할 점은 태그의 CSS 속성이 아니라 그냥 속성이라는 것.

<input id="id1 " name="email" type="text" value="some@nn.com" style="width: 100px;" />

위에서 태그의 속성은 id, name, type, value, style 이다. width는 CSS 속성이다.

이 태그를 선택하는 실렉터의 예는 다음과 같다.

input[id=id1] /\* 태그명와 속성값으로 선택 \*/

[id=id1] /\* 태그명 생략하고 속성값만으로 선택 \*/

[name=emai][type=text] /\* 여러 속성을 AND로 결합 \*/

### [속성명]

속성값은 상관 없이 그 속성의 존재 유무로 태그를 선택한다.

<input type="text" name="no" readonly value="34" />

이 태그를 선택하는 실렉터의 예는 다음과 같다.

input[readonly] /\* 태그명와 속성의 존재로 선택 \*/

[type=text][value] /\* 여러 속성을 AND로 결합 \*/

[value][readonly] /\* 여러 속성을 AND로 결합 \*/

### [속성명^=속성값]

### [속성명$=속성값]

### [속성명\*=속성값]

태그의 속성명과 그 값의 일부로 선택하는 실렉터들이다. 값 전체가 아니라 값의 일부분만 비교한다.

^= 속성값의 앞부분만 비교

$= 속성값의 끝부분만 비교

\*= 속성값의 일부분만 비교

<input id="id1 " name="email" type="text" value="some@nn.com" style="width: 100px;" />

이 태그를 선택하는 실렉터의 예는 다음과 같다.

input[value=some@n.com] /\* 값 전체 비교 \*/

input[value^=some] /\* 값의 앞부분만 비교 \*/

input[value$=com] /\* 값의 끝부분만 비교 \*/

input[value\*=nn] /\* 값의 일부분만 비교 \*/

### :checked

체크 표시된 <input type="checkbox" /> <input type="radio" /> 태그들만 선택함.

### :nth-child(숫자)

### :nth-of-type(숫자)

부모 태그로부터 숫자 번째 자식 태그만 선택하는 실렉터이다.

:nth-child는 부모의 모든 자식들 중에서 숫자 번째 자식을 선택

:nth-of-type은 부모의 자식들을 태그 종류로 그룹을, 그 태그 그룹에서 숫자 번째 자식을 선택

div:nth-child(2) /\* 부모의 모든 자식들 중 2번째인 div 태그를 선택 \*/

div:nth-of-child(2) /\* 부모의 div 자식들 중 2번째인 div 태그를 선택 \*/