

3장은 배워볼게 없음. 전부 다 쓱다 이해하고, 표 완성 00

## ch03. CSS3\_01

---

# 학습 목차

---

1. CSS3 소개
2. CSS3 단위
3. CSS3선택자
4. 기본 선택자
5. 속성 선택자
6. 조합 선택자
7. 가상 클래스 선택자
8. 가상 원소 선택자

\* CSS에서는 선택자가 가장 중요함.  
and box model

# 학습 목표

---

- CSS3 언어가 무엇인지 안다.
- CSS3 스타일 시트를 작성하는 방법을 안다.
- CSS3 선택자를 이해하고 적절한 선택자를 활용 할 수 있다
- CSS3 단위를 이해하고 적절한 단위를 선택하여 활용할 수 있다.
- CSS3 적용 순서를 이해하고 활용할 수 있다

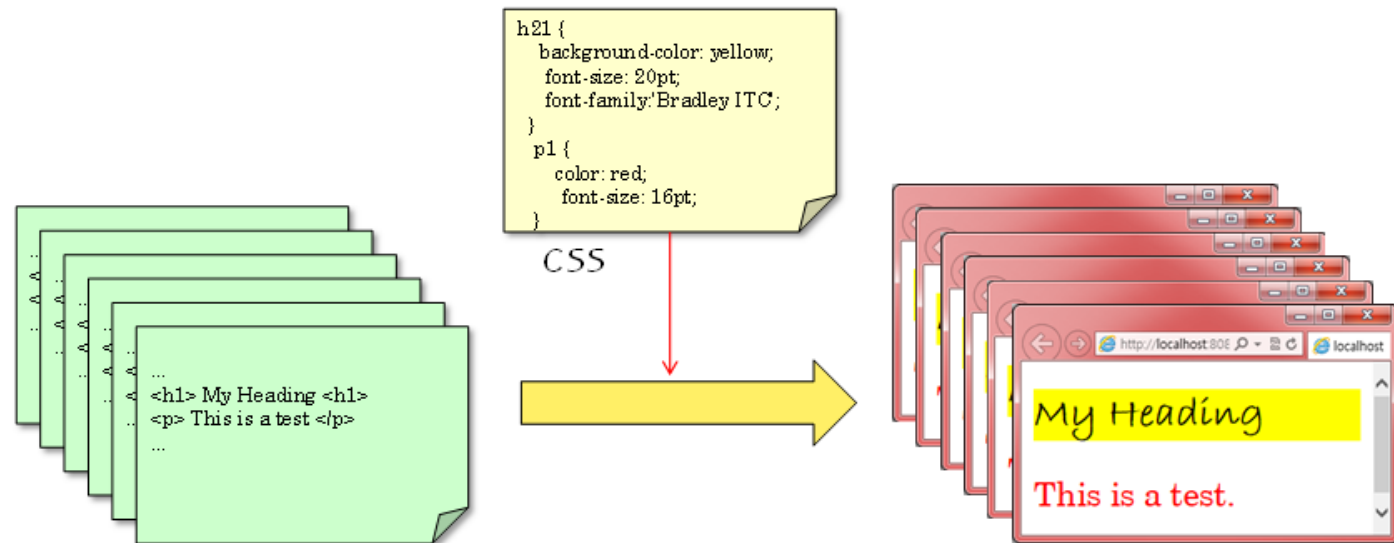
# 1. CSS3 소개

---

- CSS3(Cascading Style Sheet)
  - HTML 문서의 색이나 모양 등 외관을 꾸미는 언어
  - CSS로 작성된 코드를 스타일 시트(style sheet)라고 부름
- 발전 과정
  - CSS1
    - 웹 문서의 단순한 글꼴, 텍스트 정렬 방식, 마진 등을 정의하는 데 사용
  - CSS2
    - 1998년에 발표되어 거의 모든 브라우저에서 사용
    - 글꼴 규정 및 현재 사용되고 있는 CSS의 모든 규격 등이 이 버전에서 시작됨
  - CSS3
    - text, fonts, color, backgrounds & borders, transforms, transitions, animations과 같은 종류의 모듈을 추가로 지원
    - 기존의 CSS2가 갖지 못했던 화려하고 역동적인 표현 추가하여 자바스크립트 같은 기술에만 의존하던 영역을 지원

# 1. CSS3 소개

- 필요성
  - 문서 작성과 디자인을 분리
    - 하나의 웹 문서에서 문서 작성은 HTML이, 디자인은 CSS가 담당
- 장점
  - 내용과 디자인 수정이 용이
  - 다양한 기능으로 확장 가능
  - 통일된 문서 양식 제공 ☆
  - 전송 및 로딩 시간 단축



# 1. CSS3 소개

- 스타일 시트 구성

- 선택자

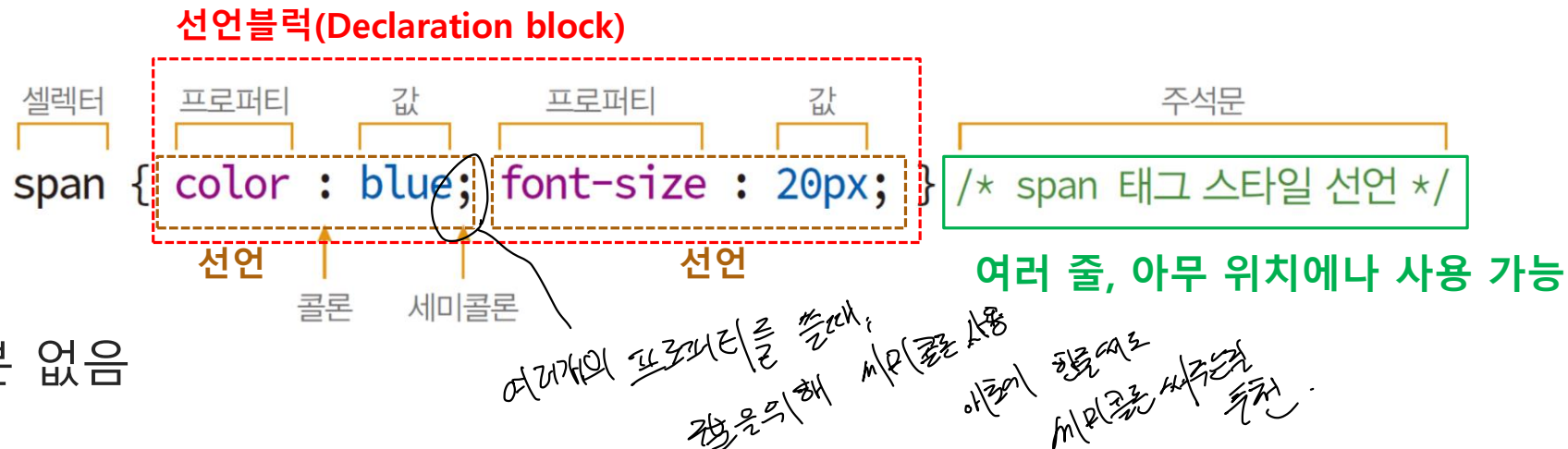
- 스타일을 적용하고자 하는 HTML 요소를 선택하기 위해 CSS에서 제공하는 수단

- 프로퍼티(속성)

- 선택자로 HTML 요소를 선택하고 {, } 내에 프로퍼티(속성)와 값을 지정하는 것으로 다양한 style을 정의

- 속성값

- 해당 프로퍼티에 사용할 수 있는 값을 "키워드"나 "크기 단위" 또는 "색상 표현 단위" 등의 특정 단위로 지정

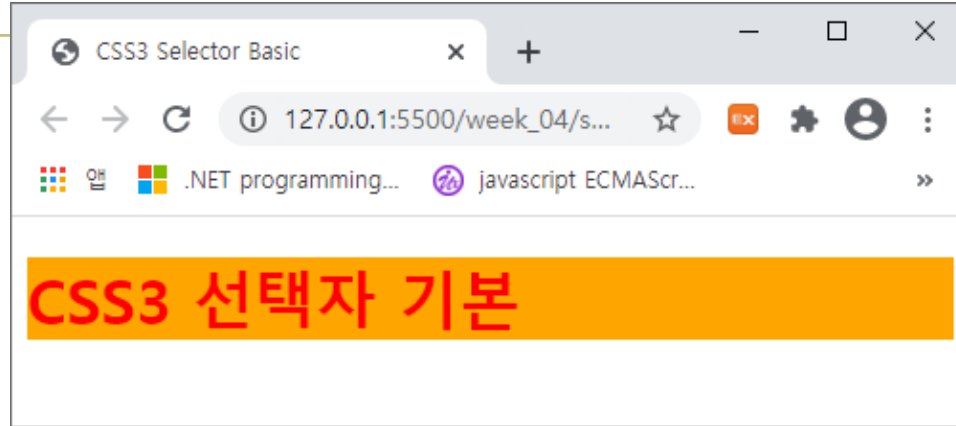


- 대소문자 구분 없음

# 1. CSS3 소개

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    h1 {
      color: red;
      background-color: orange;
    }
  </style>
</head>
<body>
  <h1>CSS3 선택자 기본</h1>
</body>
</html>
```

선택자 (h1)  
표크지리 (color: red; background-color: orange;)  
속성값 (color: red; background-color: orange;)



## 기본선택자

종류	형태	설명
전체 선택자	*	HTML 페이지 내부의 태그를 모두 선택
태그 선택자	태그	HTML 페이지 내부의 특정 태그를 모두 선택
아이디 선택자	#아이디	특정 id 속성이 있는 태그 선택, 웹 표준에 id 속성은 웹 페이지 내부에서 중복되면 안 된다는 규정이 있으므로 아이디 선택자는 특정 태그 하나를 선택할 때 사용
클래스 선택자	.클래스	특정 클래스가 있는 태그 선택

# 1. CSS3 소개

- HTML 문서에 CSS3 스타일 시트 만드는 방법 3 가지
  - **Embedding style(1)**
    - <style></style> 태그에 스타일 시트 작성
    - <style> 태그는 <head> 태그 내에서만 사용
    - <style> 태그는 여러 번 작성 가능
    - <style> 태그에 작성된 스타일 시트는 웹 페이지 전체에 적용

```
<!DOCTYPE html>
<html>
<head> <title>&lt;style&gt; 태그로 스타일 만들기</title>
<style>
  body {
    background-color : linen;
    color : blueviolet;
    margin-left : 30px;
    margin-right : 30px;
  }
  h3 {
    text-align : center; color : darkred;
  }
</style>
</head>
<body>
<h3>css 소개</h3>
<hr>
<p>HTML 문서의 색이나 모양 등 외관을 꾸미는 언어
</p>
</body>
</html>
```

외관  
이걸  
메뉴

수평 가운데 정렬.



# 1. CSS3 소개

- HTML 문서에 CSS3 스타일 시트 만드는 방법 3 가지

- **Inline style(2)**

- 태그의 style 속성에 스타일 시트 작성
    - 해당 태그에만 스타일 적용

style속성      이 태그에만 적용되는 CSS3 스타일 시트

```
<p style = "color : magenta; font-size : 30px">
    축구를 좋아합니다.
</p>
```

```
<!DOCTYPE html>
<html>
  <head> <title>&lt;style&gt; 속성에 스타일 만들기</title>
    <style>
      p { color : red; font-size : 15px; } /* 모든 p 태그에 적용 */
    </style>
  </head>
  <body>
    <h3>css3 구성</h3>
    <hr>
    <p>선택자</p>
    <p>프로퍼티</p>
    <p style="color:blue">속성값</p>
    <p style="color:magenta; font-size:30px">대소문자 구분 없음</p>
  </body>
</html>
```

```
<body>
  <h1>Hello WEB!!!!</h1>
  <script>
    let $ele = document.querySelector('h1');
    $ele.style.color = 'red';
  </script>
</body>
```

자바 스크립트를 사용하여 스타일 지정하면 inline style 형태로 적용

`<h1 style="color: red;">Hello WEB!!!!</h1>`

# 1. CSS3 소개

- HTML 문서에 CSS3 스타일 시트 만드는 방법 3 가지

- **Link style(3)**

- 스타일 시트를 **별도 파일**로 작성(\*.css) – 가장 일반적인 방법

```
<style> /* mystyle.css */  
  body { background-color:linen; color:blueviolet; margin-left:30px; margin-right:30px; }  
  h3 { text-align:center; color:darkred; }  
</style>
```

외부 스타일 시트 파일에 HTML 태그 내용은 존재할 수 없음

- **<link> 태그나 @import**로 불러 사용

```
<head> <!-- 1) link 태그 이용 -->  
  <link href="mystyle.css" type="text/css" rel="stylesheet">  
</head>
```

```
<style> <!-- 2) @import 이용 -->  
  @import url(mystyle.css);  
  /* @import url('mystyle.css'); 로 해도 됨 */  
  /* @import "mystyle.css";로 해도 됨 */  
</style>
```

- 동일한 스타일 시트를 웹 페이지마다 중복 작성 해소
- 웹 사이트의 전체 웹 페이지 모양의 일관성 확보

화려함

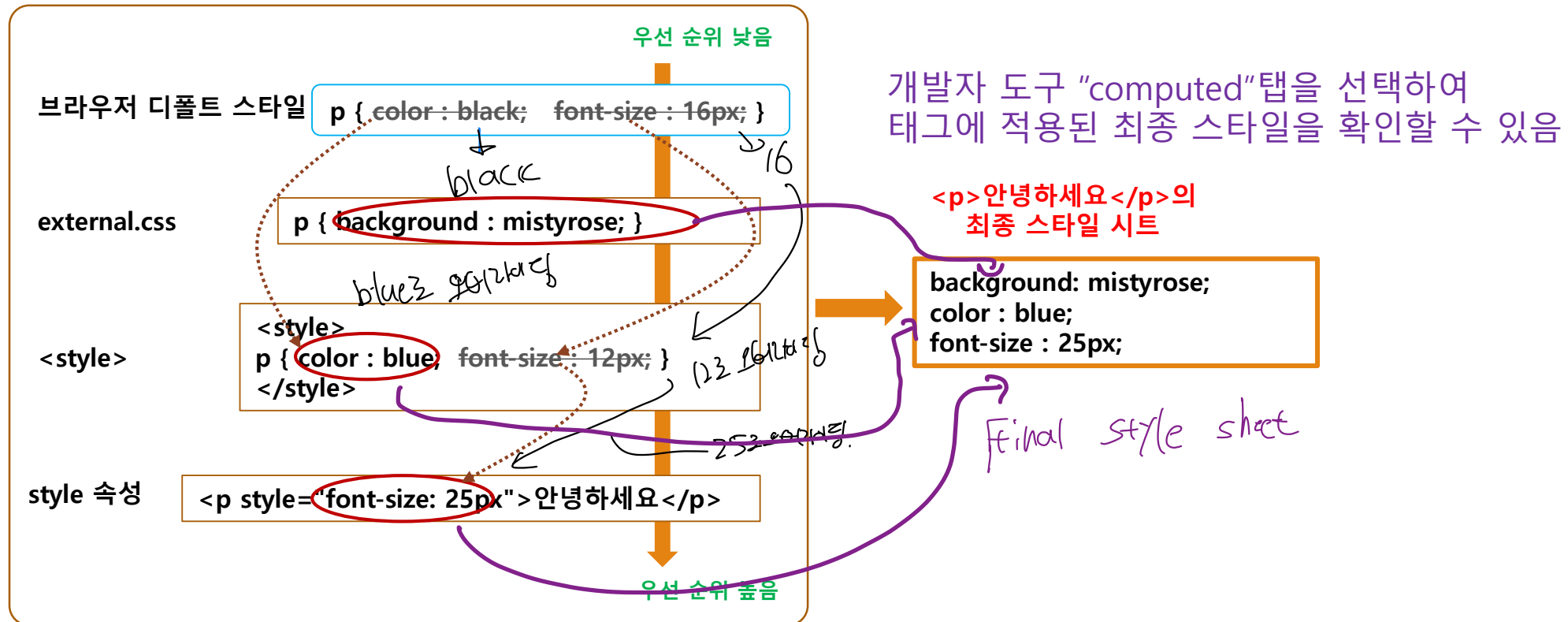
있으면 큰일남

방법 ①

방법 ②

# 1. CSS3 소개

- css3 규칙
  - 스타일 합치기(cascading)와 오버라이딩(overriding)
  - 태그에 적용되는 모든 스타일이 합쳐지고, 동일한 스타일은 순위가 높은 스타일이 우선 적용되는 규칙



# 1. CSS3 소개

\*우선순위의 규칙을 한안더 하기

```
<head>
  <style>
    .classdemo{ color :■ chocolate; } /* class 선택자, 우선순위:10점 */
    #idexam{ color :■ gold; } /* id선택자, 우선순위:100점 */
    p#idexam.classdemo {color:■ magenta;} /* 우선순위:1+100+10 */
    p { color:■ blueviolet; } /* 요소선택자, 우선순위: 1점 */
    p { color:■ green; } /* 우선순위:1점 */
  </style>
</head>
<body>
  <p style="color:■ blue">css 우선 순위</p> <!-- 우선순위:1000점 -->
  <p id="idexam" class="classdemo">css 우선 순위</p>
</body>
```

## ◎css3 적용 순위

1. Inline styles
2. Id 선택자
3. Classes, pseudo-classes(:hover), attribute([href]) 선택자
4. Elements(태그), pseudo-elements(::before) 선택자

◦ 동일한 우선순위의 경우 마지막에 선언한 스타일을 적용

## 2. CSS3 단위

- 키워드 단위
  - W3C에서 미리 정의한 단어
  - 키워드를 입력하면 해당하는 스타일이 자동으로 적용

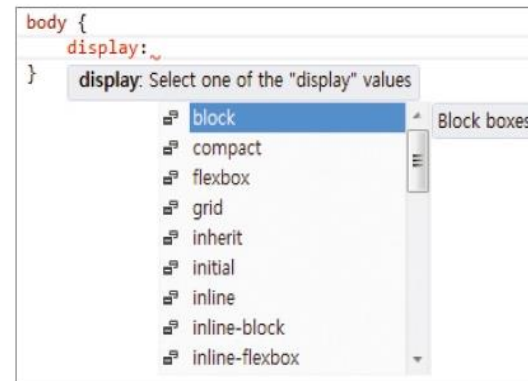


그림 5-6 개발 환경을 이용한 속성별 자동 완성 기능

- URL 단위
  - 이미지나 글꼴 파일을 불러올 때 사용
  - url('경로')

/\* 현재 폴더의 desert.JPG \*/

background-image : url('desert.JPG');

/\* other 폴더의 desert.JPG \*/

background-image : url('other/desert.JPG');

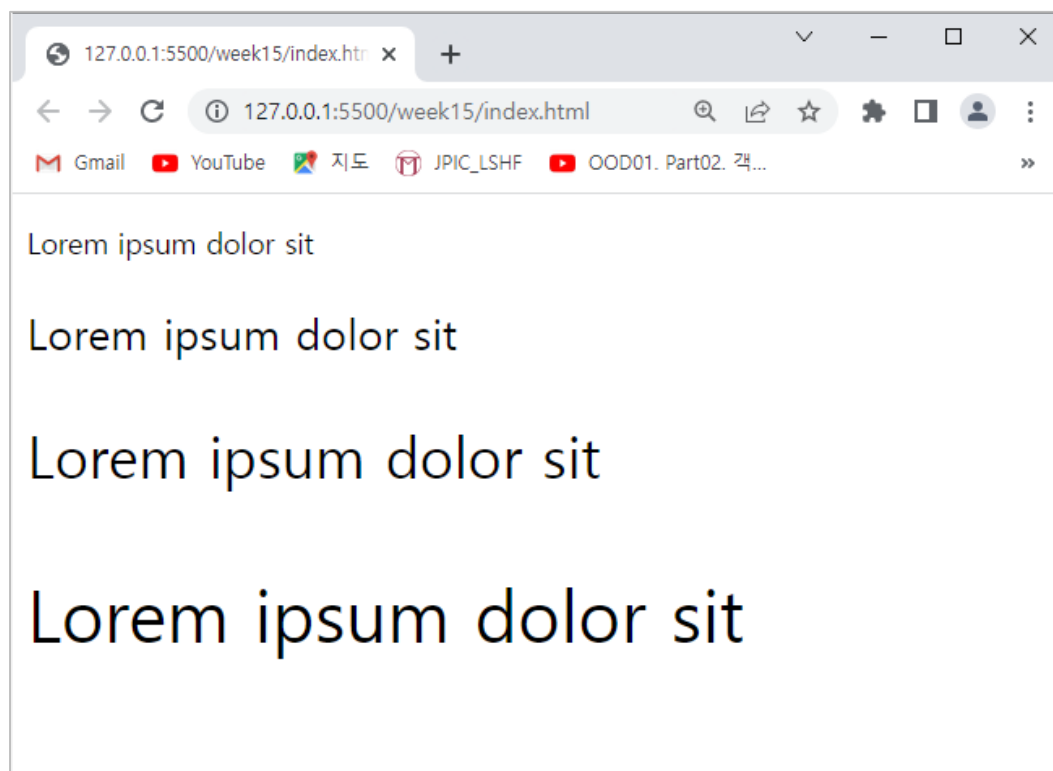
- 크기 단위

- 절대 길이 : px
- 상대 길이 : %, em, rem, vw, vh
  - ↳ 부모 기준 (상대적인 처리)
  - ↳ 뷰포트 단위 (반응형)

## 2. CSS3 단위

- % 단위 적용하기
  - % 단위는 부모 요소를 기준으로 **상대적인 크기를 지정**.
  - 초기 설정 크기: 100%

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    #aa{font-size: 100%;}
    #bb{font-size: 150%;}
    #cc{font-size: 200%;}
    #dd{font-size: 250%;}
  </style>
</head>
<body>
  <p id="aa">Lorem ipsum dolor sit</p>
  <p id="bb">Lorem ipsum dolor sit</p>
  <p id="cc">Lorem ipsum dolor sit</p>
  <p id="dd">Lorem ipsum dolor sit</p>
</body>
</html>
```



## 2. CSS3 단위

- em 단위
  - 배수(倍數) 단위
  - 부모 요소를 기준으로 상대적인 사이즈를 설정
  - 1배=1em=100%
  - 1.5배=1.5em=150%

Font size: 1.2em → 14px \* 1.2 = 16.8px

Font size: 1.2em ⇒ 14px \* 1.2 = 16.8px

Font size: 1.2em ⇒ 16.8px \* 1.2 = 20.16px

Font size: 1.2em ⇒ 20.16px \* 1.2 = 24.192px

- rem 단위
  - 최상위 요소(root)의 크기 기준
  - rem의 r은 root를 의미

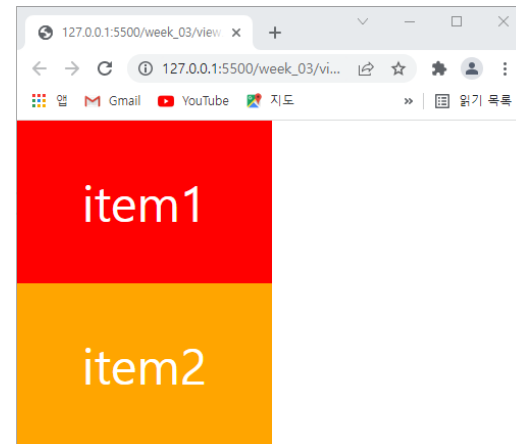
```
<head>
  <style>
    body {
      font-size: 14px;
    }
    div {
      font-size: 1.2em; /* 14px * 1.2 = 16.8px */
      font-weight: bold;
      padding: 2em; /* 16.8px * 2 = 33.6px */
    }
  </style>
</head>
<body>
  <div>Font size: 1.2em → 14px * 1.2 = 16.8px</div>
  <div class='box1'>
    Font size: 1.2em ⇒ 14px * 1.2 = 16.8px
    <div class='box2'>
      Font size: 1.2em ⇒ 16.8px * 1.2 = 20.16px
      <div class='box3'>
        Font size: 1.2em ⇒ 20.16px * 1.2 = 24.192px
      </div>
    </div>
  </div>
</body>
```

## 2. CSS3 단위

- viewport 단위
  - viewport - 웹 페이지의 가시 영역
  - viewport 비율에 따라 크기 결정
  - 예 : viewport 너비가 1000px, 높이가 600px인 경우,
    - 1vw : viewport 너비 1000px의 1%인 10px
    - 1vh : viewport 높이 600px의 1%인 6px
    - vmin : viewport 높이 600px의 1%인 6px
    - vmax : viewport 너비 1000px의 1%인 10px

단위	설명
vw	viewport 너비의 1/100
vh	viewport 높이의 1/100
vmin	viewport 너비 또는 높이 중 작은 쪽의 1/100
vmax	viewport 너비 또는 높이 중 큰 쪽의 1/100

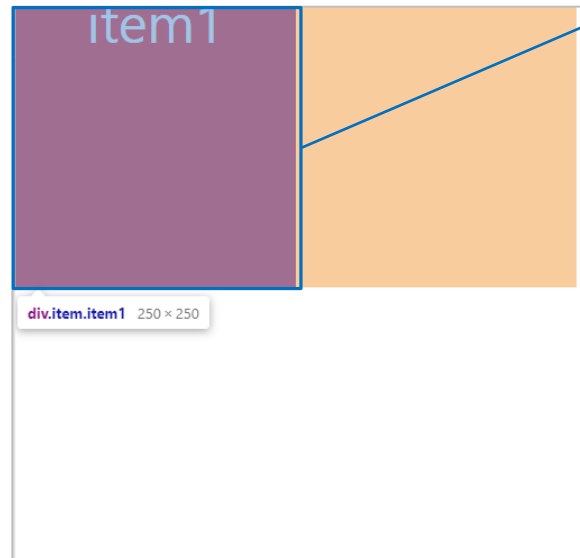
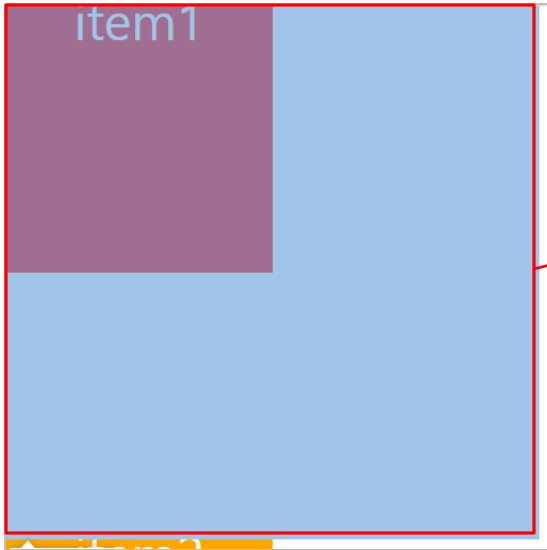
```
<head>
  <style>
    :root{ font-size: 12px;}
    body { margin: 0px; }
    .item {
      width: 50vw;
      height: 50vh;
      text-align: center;
      line-height: 50vh;
      font-size: 4rem;
      color: white;
    }
    .item1 { background-color: red; }
    .item2 { background-color: orange; }
  </style>
</head>
<body>
  <div class='item item1'>item1 </div>
  <div class='item item2'>item2 </div>
</body>
```





## 2. CSS3 단위

- em, % 단위와 viewport 단위 차이점
  - em, % 단위는 상속에 의해 부모 요소에 영향을 받음
  - viewport 단위는 viewport 기준



```
<head>
<style>
  body { margin: 0px; }
  :root{
    font-size: 12px;
  }
  :first-child {
    width: 500px;
    height: 500px;
  }
  .item {
    width: 50%;
    height: 50%;
    text-align: center;
    line-height: 50%;
    font-size: 4rem;
    color: white;
  }
  .item1 { background-color: red; }
  .item2 { background-color: orange; }
</style>
</head>
<body>
  <div>
    <div class='item item1'>item1</div>
  </div>
  <div class='item item2'>item2</div>
</body>
```

## 2. CSS3 단위

- 색 표현 3 가지 방법

- 16진수 코드로 표현

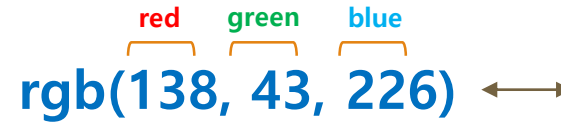
# 8A2BE2



빨간색(r) 성분 0x8A(138),  
초록색(g) 성분 0x2B(43),  
파란색(b) 성분 0xE2(226) 이  
혼합된 보라색(blueviolet)

- 10진수 코드와 RGB()로 표현

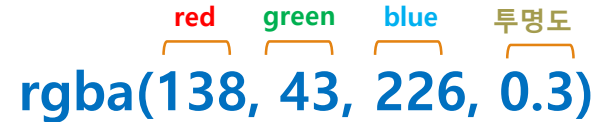
rgb(138, 43, 226)



빨간색(r) 성분 138,  
초록색(g) 성분 43,  
파란색(b) 성분 226 이  
혼합된 보라색(blueviolet)

- RGBA (Red, Green, Blue, Alpha/투명도)

rgba(138, 43, 226, 0.3)



투명도는 0.0~1.0  
0.0 은 완전 투명  
1.0 은 완전 불투명

- 색 이름으로 표현

- CSS3 표준에서는 140개 색의 이름을 정하고 있음

### CSS 스타일

```
div {  
  color : rgb(138, 43, 226);      /* 글자색 blueviolet */  
  background-color : gold;        /* 배경색 gold */  
  border-color : #6B8E23;        /* 테두리색 olivedrab(#6B8E23) */  
}
```

<div>CSS에서 r, g, b로 구성됩니다. </div>

### HTML 코드

# 3. CSS3 선택자

종류	형태	사용 예
전체 선택자	*	*
태그 선택자	태그	h1
아이디 선택자	#아이디	#id
클래스 선택자	.클래스	.header
속성 선택자	선택자[속성 = 값]	input[type = text]
	선택자[속성 ~= 값]	div[data-role ~= row]
	선택자[속성  = 값]	div[data-role  = row]
	선택자[속성 ^= 값]	div[data-role ^= row]
	선택자[속성 \$= 값]	div[data-role \$= 9]
	선택자[속성 *= 값]	div[data-role *= row]
후손 선택자	선택자 선택자	header h1
자손 선택자	선택자 > 선택자	header > h1
반응 선택자	선택자:active	div:active
	선택자:hover	div:hover
상태 선택자	선택자:checked	input:checked
	선택자:focus	input:focus
	선택자:enabled	input:enabled
	선택자:disabled	input:disabled

중요함

종류	형태	사용 예
구조 선택자	선택자:first-child	li:first-child
	선택자:last-child	li:last-child
	선택자:nth-child(수열)	li:nth-child(2n+1)
	선택자:nth-last-child(수열)	li:nth-last-child(2n+1)
	선택자:first-of-type	h1:first-of-type
	선택자:last-of-type	h1:last-of-type
동위 선택자	선택자:nth-of-type(수열)	h1:nth-of-type(2n+1)
	선택자:nth-last-of-type(수열)	h1:nth-last-of-type(2n+1)
	선택자 + 선택자	h1 + div
	선택자 ~ 선택자	h1 ~ div
링크 선택자	선택자:link	a:link
	선택자:visited	a:visited
문자 선택자	선택자::first-letter	p::first-letter
	선택자::first-line	p::first-line
	선택자::after	p::after
	선택자::before	p::before
부정 선택자	선택자::selection	p::selection
	선택자:not(선택자)	li:not(.item)

# 4. 기본 선택자

- 태그(요소) 선택자

- 태그 이름이 선택자로 사용되는 유형
- 선택자와 같은 모든 태그에 CSS3 스타일 시트 적용

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Selector Basic Page</title>
  <style><!-- content 글자 색 변경 -->
    h1 { color: red; }
    p { color: blue; }
  </style>
</head>
<body>
  <h1>제목 글자</h1>
  <p>Lorem ipsum dolor sit amet.</p>
  <p>Etiam lacus felis, ornare non lobortis ac, vehicula non diam.</p>
</body>
</html>
```

여러 개의  
태그에 적용

## 선택자 여러 개에 속성 적용하기: 쉼표 사용

```
<style>
body, p, h1, h2, h3, h4, h5, h6 { margin: 0; padding: 0; }
</style>
```

그림 5-2 선택자를 여러 개 사용

# 4. 기본 선택자

- **class 선택자**

- 점(.)으로 시작하는 선택자
- HTML 태그의 class 속성으로만 지정 가능
  - class 속성이 같은 모든 태그에 적용

```
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    .select { color: red; }
  </style>
</head>
<body>
  <ul>
    <li class="select">사과</li>
    <li>바나나</li>
    <li class="select">오렌지</li>
    <li>감</li>
  </ul>
</body>
```

- 클래스 명 작성 규칙

- 1.문자와 숫자 사용 - 하이픈이나 밑줄 사용 가능
- 2.첫 문자로 숫자,밑줄, 하이픈 사용 불가
- 3.공백 사용 불가

- HTML 요소에 class 속성 값은 공백으로 구분하여 여러 개 지정 가능
- HTML 요소에 대하여 이미 정의되어 있는 속성과 함께 필요한 스타일을 지정 - 재사용에 유용

```
<head>
  <style>
    .text-center { text-align: center; }
    .text-large { font-size: 200%; }
    .text-red { color: red; }
    .text-blue { color: blue; }
  </style>
</head>
<body>
  <p class="text-center">Center</p>
  <p class="text-large text-red">Large Red</p>
  <p class="text-center text-large text-blue">Center Large Blue</p>
</body>
```

# 4. 기본 선택자

---

- class 선택자

- 클래스 속성은 중복 가능
- 클래스 속성을 서로 다른 태그에 적용
  - 태그 선택자와 클래스 선택자를 함께 사용, 정확한 태그 선택을 할 수 있다

```
<head>
  <title>CSS3 Selector Basic Page</title>
  <style>
    li.select {
      color :chocolate;
    }
    h1.select{
      color :darkgoldenrod;
    }
  </style>
</head>
<body>
  <h1 class="select">클래스 속성 중복</h1>
  <ul>
    <li class="select">사과</li>
    <li>바나나</li>
    <li>오렌지</li>
    <li>감</li>
  </ul>
</body>
```

---

# 4. 기본 선택자

- id 선택자

- #으로 시작하는 이름의 선택자
- HTML 태그의 id 속성으로만 지정 가능
- id 속성은 중복되지 않도록 해야 함(웹 표준)



```
<head>
  <title>CSS3 Selector Basic Page</title>
  <style>
    #header {
      background: red;
    }
    #list{
      background-color: cadetblue;
    }
  </style>
</head>
<body>
  <div id="header">
    <h1>#header 태그</h1>
  </div>
  <ul id="list">
    <li>HTML5</li>
    <li>CSS3</li>
    <li>JavaScript</li>
  </ul>
</body>
```

# 4. 기본 선택자

---

- id 선택자와 class 선택자 비교
  - id 선택자
    - id 속성의 목적은 각 태그를 유일하게 구분
      - 동일한 id 속성을 갖지 않도록 HTML 파일 작성하는 것이 바람직
      - 자바스크립트 코드에서 id 값을 가진 태그 객체를 찾을 때 문제됨
    - 적합한 활용
      - id 선택자는 여러 태그 중 특정 태그에만 CSS 스타일을 적용할 때 적합
  - class 선택자
    - 적합한 활용
      - 여러 태그를 하나의 그룹으로 묶어 단체로 동일한 CSS 스타일을 적용할 때 적합
      - class 속성 값이 같은 태그에 모두 CSS 스타일 적용
    - 태그의 종류에 관계없이 class 선택자 활용 가능
    - 사용 빈도가 높음



# 4. 기본 선택자

---

- 전체 선택자(universal selector)
  - 와일드 문자(\*)를 사용하여 모든 태그에 적용시키는 선택자

---

```
<head>
  <title>CSS3 Selector Basic Page</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      color: deeppink;
    }
  </style>
</head>
<body>
  <label for="name">이름:</label>
  <input type="text" id="name">
  <label for="pass">패스워드:</label>
  <input type="password" id="pass">
</body>
```

---

# 5. 속성 선택자

- HTML 태그의 특정 속성(attribute)에 대해 값이 일치하는 태그에만 스타일을 적용하는 선택자
- 속성 선택자 형식

형식	설명	사용 예
[속성]	해당 속성이 정의된 모든 태그를 선택한다. 속성의 값과는 무관하다.	p[text]
[속성=값]	정의된 속성과 속성값이 동일한 태그를 선택한다.	p[text="red"]
[속성~=값]	공백으로 구분된 속성값 목록 중 하나가 주어진 값과 동일한 태그를 선택한다.	p[text~="red blue"]
[속성 =값]	속성값이 해당 값과 동일하거나, 또는 주어진 값으로 시작하고 '-' 기호로 이어지는 값을 가진 태그를 선택한다.	p[text ="red-blue"]
[속성^=값]	속성값이 주어진 값으로 시작하는 태그를 선택한다.	p[text^="img"]
[속성\$=값]	속성값이 주어진 값으로 끝나는 태그를 선택한다.	p[text\$=".png"]
[속성*=값]	속성값이 주어진 값을 부분 문자열로 가지는 태그를 선택한다.	p[text*="ong"]

# 5. 속성 선택자 – 사용 예

```
<head>
<style>
* {
  margin: 2px;
  padding: 2px;
}
label[for]{
  background-color:gold;
}
input[type="text"] {
  background-color: aqua;
}
input[type="password"] {
  background-color: darkgray;
}
</style>
</head>
```



```
<body>
<label for="name">이름:</label>
<input type="text" id="name">
<br>
<label for="pass">패스워드:</label>
<input type="password" id="pass">
</body>
```

```
<head>
<style>
/* div 요소 중에서 class 속성 값에 "test"를 포함하는 요소 */
div[class *= "test"] { color: red; }

/* div 요소 중에서 class 속성 값에 "test"를 단어로 포함하는 요소 */
div[class ~= "test"] { background-color: yellow; }

/* a 요소 중에 href 속성 값이 ".html"로 끝나는 요소 */
a[href $=".html"] { color: brown; }

/* a 요소 중에 href 속성 값이 "test"로 시작하는 요소 */
a[href ^="test"] { background-color: gold }

/* h1 요소 중에 title 속성 값이 "first"와 일치하거나 "first-"로 시작하는 요소 */
h1[title |= "first"] { color: red; }

</style>
</head>
```

```
<body>
<div class="first_test">첫번째 블럭요소</div>
<div class="second">두번째 블럭요소</div>
<div class="first test">세번째 블럭요소</div>
<a href="test.html">test.html</a> <br>
<a href="test.jsp">test.jsp</a>
<h1 title="first">Heading first</h1>
<h1 title="first-heading">Heading-first</h1>
<h1 title="heading first">Heading second</h1>
</body>
```

첫번째 블럭요소  
두번째 블럭요소  
세번째 블럭요소  
test.html  
test.jsp

Heading first

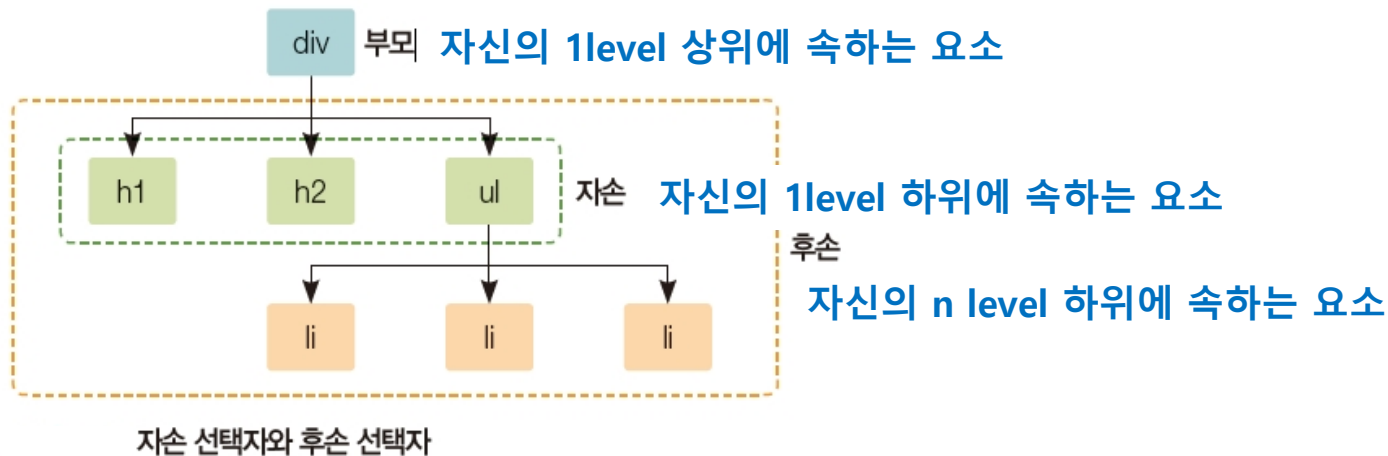
Heading-first

Heading second

# 6. 조합 선택자

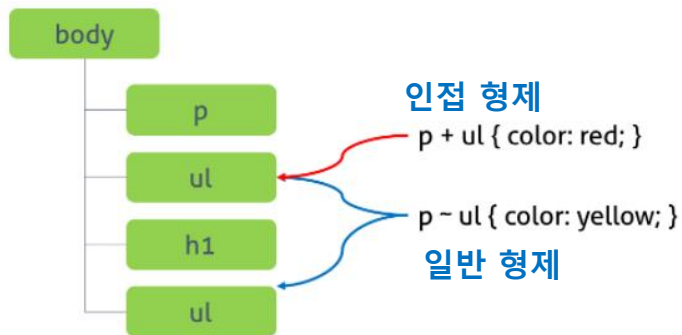
- “자손(자식)은 무엇이고 또 후손(하위)은 무엇인가요?”

```
<body>
  <div>
    <h1>CSS3 선택자 기본</h1>
    <h2>기본 선택자</h2>
    <ul>
      <li>전체 선택자</li>
      <li>태그 선택자</li>
      <li>아이디 선택자와 클래스 선택자</li>
    </ul>
  </div>
</body>
```



조합 선택자의 종류

구분	조합 방법	설명
후손 선택자	선택자 A 선택자 B	선택자 B가 선택자 A에 반드시 포함되어 있을 경우에 선택한다.
자손 선택자	선택자 A > 선택자 B	부모 선택자 A의 직계 자손인 선택자 B를 선택한다.
인접 형제 선택자	선택자 A + 선택자 B	선택자 A 바로 다음에 위치한 선택자 B를 선택한다.
일반 형제 선택자	선택자 A ~ 선택자 B	선택자 A 뒤에 인접하여 나타나는 모든 선택자 B를 선택한다.
그룹 선택자	선택자 A, 선택자 B	선택자 A과 선택자 B를 모두 선택한다.



# 6. 조합 선택자

- 후손 선택자

Lorem ipsum

Navigation

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

공백.

```
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    #header h1 { color: red; } /* #header * h1 { color: red; } */
    #section h1 { color: orange; }
  </style>
</head>
<body>
  <div id="header">
    <h1 class="title">Lorem ipsum</h1>
    <div id="nav">
      <h1>Navigation</h1>
    </div>
  </div>
  <div id="section">
    <h1 class="title">Lorem ipsum</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
  </div>
</body>
```

헤더의 라스트 h1태그

섹션의 라스트 h1태그

Lorem ipsum

Navigation

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

# 6. 조합 선택자

- 후손 선택자 여러 개를 함께 사용할 경우 주의 사항
  - ① <header> 태그의 후손인 <h1> 태그와 일반적인 <h2> 태그 선택



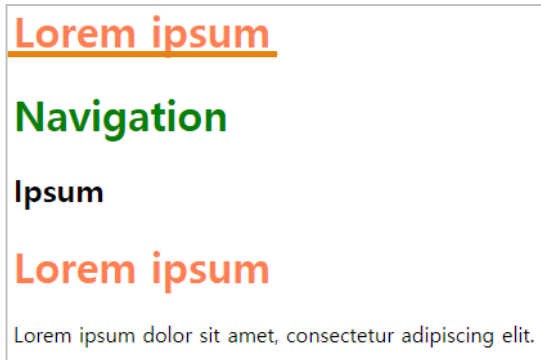
```
<div id="header">
  <h1>Lorem ipsum</h1>
  <div>
    <h1>Navigation</h1>
  </div>
  <h2>Ipsum</h2>
</div>
<h2>Ipsum</h2>
```

- ② <header> 태그의 후손인 <h1> 태그와 header 태그의 후손인 <h2> 태그 선택



# 6. 조합 선택자

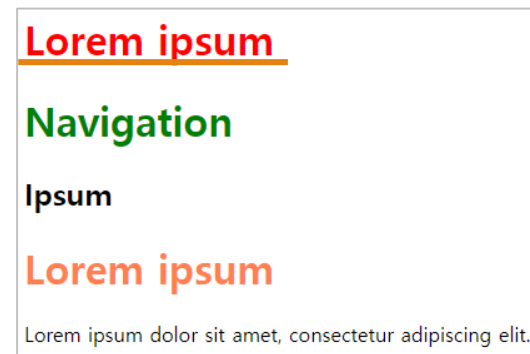
- 자손(자식) 선택자



```
<style>
  div.header>div>h1 { color: green; }
  div>h1 { color: coral; }
</style>
```

```
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    div.header>h1 { color: red; } /* 선택자 우선순위 : 아이디 > 클래스 > 태그 */
    div.header>div>h1 { color: green; }
    div>h1 { color: coral; }
  </style>
</head>
```

```
<body>
  <div class="header">
    <h1>Lorem ipsum</h1>
    <div>
      <h1>Navigation</h1>
      <h2>Ipsum</h2>
    </div>
  </div>
  <div class="section">
    <h1 class="title">Lorem ipsum</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</body>
```



# 6. 조합 선택자

- <table> 태그 요소 선택할 때 자손 선택자 주의사항
  - 웹 브라우저가 <tbody> 태그를 자동으로 추가하므로 스타일 속성이 적용되지 않음
  - table 태그에 스타일을 적용할 때는 자손 선택자를 사용하지 않음

이름	지역
hallym	춘천시 한림대학길 1

이름	지역
hallym	춘천시 한림대학길 1

← CSS 스타일을 주석으로 처리된 선언문으로 변경

```
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    table> tr > th { color: red; }
    /* table> tbody> tr > th { color: red; } */
  </style>
</head>
<body>
  <table border="1">
    <tr>
      <th>이름</th>
      <th>지역</th>
    </tr>
    <tr>
      <td>hallym</td>
      <td>춘천시 한림대학길 1</td>
    </tr>
  </table>
</body>
```

```
▼ <table border="1">
  ▼ <tbody>
    ▶ <tr>...</tr>
    ▶ <tr>...</tr>
  </tbody>
</table>
```



# 6. 조합 선택자

- 인접 형제 선택자

```
<head>
  <style>
    h1 + h2 + ul { color: blue; }
    div + h3 { color: red; }
    h3 + p { color: purple; background-color: yellow; }
  </style>
</head>
<body>
  <div>
    <h1>인접 형제 선택자1</h1>
    <h2>인접 형제 선택자2</h2>
    <ul>목록
      <li>주제1</li>
      <li>주제2</li>
    </ul>
  </div>
  <h3>Adjacent Selector_1</h3>
  <p>인접 형제 선택자에 의한 스타일 적용</p>
  <h3>Adjacent Selector_2</h3>
</body>
```

## 인접 형제 선택자1

## 인접 형제 선택자2

- 목록
- 주제1
- 주제2

## Adjacent Selector\_1

인접 형제 선택자에 의한 스타일 적용

## Adjacent Selector\_2

# 6. 조합 선택자

- 일반 형제 선택자

```
<head>
  <style>
    h1 ~ ul { color: blue; }
    div ~ h3 { color: red; }
    h3 ~ p { color: green; background-color: yellow; }
  </style>
</head>
<body>
  <div>
    <h1>형제 선택자1</h1>
    <h2>형제 선택자2</h2>
    <ul>목록
      <li>주제1</li>
      <li>주제2</li>
    </ul>
  </div>
  <h3>Sibling Selector-1</h3>
  <h4>같은 레벨 형제</h4>
  <p>일반 형제 선택자에 의한 스타일 적용</p>
  <h3>Sibling Selector_2</h3>
  <h3>Sibling Selector_3</h3>
</body>
```

형제 (h1, h2, ul, li, h3, h4, p, h3, h3)

## 형제 선택자1

### 형제 선택자2

목록

- 주제1
- 주제2

### Sibling Selector-1

같은 레벨 형제

일반 형제 선택자에 의한 스타일 적용

### Sibling Selector\_2

### Sibling Selector\_3

# 7. 가상 클래스(pseudo-class) 선택자

- 요소의 특정 상태에 따라 스타일을 정의할 때 사용하는 선택자
  - : 기호 사용
  - 특정 상태
    - 마우스가 올라와 있을 때
    - 링크를 방문했을 때와 아직 방문하지 않았을 때
    - 포커스가 들어와 있을 때

JS를 안쓰고도,  
동적 요소? }  
pseudo-class

- 링크 및 동적 셀렉터(User action pseudo-classes)

사용 방법	설명	사용 예
:link 선택자	선택자가 방문하지 않은 링크일 때	a : link { color: red; text-decoration: none; }
:visited 선택자	선택자가 방문한 링크일 때	a : visited { color: blue; }
:active 선택자	선택자가 클릭 된 상태일 때	a : active { color: black; }
:hover 선택자	선택자 위에 마우스가 있는 경우	a : hover { color: green; }
:focus 선택자	선택자에 포커스가 있는 경우	a : focus { color: yellow; }

# 7. 가상 클래스(pseudo-class) 선택자

- 링크 및 동적 셀렉터(User action pseudo-classes) 사용 예

```
<head>
  <style>
    /* 가상 클래스 선택자 */
    a:link { color: blue; text-decoration: underline; } /* a 요소가 방문하지 않은 링크일 때 */
    a:visited { color: red; } /* a 요소가 방문한 링크일 때 */
    a:hover { text-decoration: overline; } /* a 요소에 마우스가 올라와 있을 때 */
    a:active { background-color: yellow; } /* a 요소가 클릭 된 상태일 때 */

    div.d1 { border: 1px dashed red; width: 400px; padding: 5px; }
    div.d1:hover { background-color: yellow; }

    div.d2 { border: 1px dashed green; width: 400px; padding: 5px; }
    div.d2:hover { background-color: green; }
  </style>
</head>
<body>
  <h2>Pseudo Class</h2>
  <p><a href="http://www.w3.org" target="_blank">W3C 방문</a> : 마우스 이벤트에 따른 링크의 변화를 잘 보세요.</p>
  <div class="d1">
    <h3>가상 클래스 1 영역</h3>
    마우스 위치에 따른 박스의 스타일 변화를 보세요.
  </div>
  <div class="d2">
    <h3>가상 클래스 2 영역</h3>
    마우스 위치에 따른 박스의 스타일 변화를 보세요.
  </div>
</body>
```

## Pseudo Class

W3C 방문 : 마우스 이벤트에 따른 링크의 변화를 잘 보세요.

### 가상 클래스 1 영역

마우스 위치에 따른 박스의 스타일 변화를 보세요.



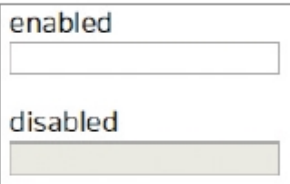
### 가상 클래스 2 영역

마우스 위치에 따른 박스의 스타일 변화를 보세요.

# 7. 가상 클래스(pseudo-class) 선택자

- 상태 선택자
  - 입력 양식의 상태 선택

형태	설명
:checked	체크 상태의 input 태그 선택
:focus	포커스를 맞춘 input 태그 선택
:enabled	사용 가능한 input 태그 선택
:disabled	사용 불가능한 input 태그 선택

상태 선택자의 상태	
상태	설명
	type 속성이 checkbox 또는 radio인 input 태그가 선택된 상태
	사용자가 바로 입력할 수 있도록 입력 양식에 포커스를 둔 상태, 웹 페이지 하나당 input 태그 하나에만 포커스를 둘 수 있음
	input 태그에 값을 입력할 수 있는 상태고 disabled는 input 태그에 값을 입력할 수 없는 상태, input 태그에 disabled 속성을 입력해 적용 <pre>&lt;body&gt;   &lt;input value="enabled"&gt;   &lt;input value="disabled" disabled="disabled"&gt; &lt;/body&gt;</pre>

# 7. 가상 클래스(pseudo-class) 선택자

- 상태 선택자

```
<head>
<style>
  /* input 요소가 사용 가능한 상태일 때, input 요소 바로 뒤에 위치하는 인접 형제 span 요소를 선택 */
  input:enabled+span { color: blue; }

  /* input 요소가 사용 불가능한 상태일 때, input 요소 바로 뒤에 위치하는 인접 형제 span 요소를 선택 */
  input:disabled+span { color: gray; text-decoration: line-through; }

  /* input 요소가 체크 상태일 때, input 요소 바로 뒤에 위치하는 인접 형제 span 요소를 선택 */
  input:checked+span { color: red; }

  input:focus { background-color: red; }

  input:checked + div.d1 { color: coral }
  input:checked + div.d2 { color: black; }

  .d1, .d2 { color: whitesmoke; }

</style>
</head>
```

```
<body>
  <input type="radio" checked="checked" value="male" name="gender"> <span>Male</span> <br>
  <input type="radio" value="female" name="gender"> <span>Female</span> <br>
  <input type="radio" value="neuter" name="gender" disabled> <span>Neuter</span>
  <hr>
  <h3>문제) 대한민국 수도는?</h3>
  <p>정답 작성 : <input type="text"> </p>
  <h2>answer</h2>
  힌트 보기 : <input type="checkbox">
  <div class="d1"> 남대문이 있는 곳이죠. </div>
  정답 보기 : <input type="checkbox">
  <div class="d2"> 서울 </div>
</body>
```

☒ Male  
☐ Female  
☐ Neuter

문제) 대한민국 수도는?

정답 작성 :

answer

힌트 보기 : ☒

남대문이 있는 곳이죠.

정답 보기 : ☐

서울

\* 형제 판단이  
상당히 중요해보임  
같은태그라면 이  
(형식) 문제해결.

# 7. 가상 클래스(pseudo-class) 선택자

- 구조 선택자 – 특정 위치의 태그 선택

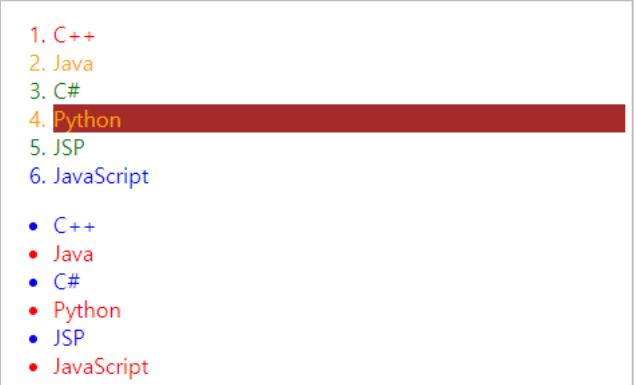
형태	설명
:first-child	셀렉터에 해당하는 모든 요소 중 <u>첫번째 자식인</u> 요소를 선택
:last-child	셀렉터에 해당하는 모든 요소 중 <u>마지막 자식인</u> 요소를 선택
:nth-child(n)	셀렉터에 해당하는 모든 요소 중 <u>앞에서 n번째 자식인</u> 요소를 선택
:nth-last-child(n)	셀렉터에 해당하는 모든 요소 중 <u>뒤에서 n번째 자식인</u> 요소를 선택
:first-of-type	셀렉터에 <u>해당하는 요소의</u> 부모 요소의 자식 요소 중 <u>첫번째 등장하는</u> 요소를 선택
:last-of-type	셀렉터에 <u>해당하는 요소의</u> 부모 요소의 자식 요소 중 <u>마지막에 등장하는</u> 요소를 선택
:nth-of-type(n)	셀렉터에 <u>해당하는 요소의</u> 부모 요소의 자식 요소 중 <u>앞에서 n번째에</u> 등장하는 요소를 선택
:nth-last-of-type(n)	셀렉터에 <u>해당하는 요소의</u> 부모 요소의 자식 요소 중 <u>뒤에서 n번째에</u> 등장하는 요소를 선택
:not(셀렉터)	셀렉터에 해당하지 않는 모든 요소를 선택

:first-child => 형제 요소의 그룹 중 첫번째 요소를 나타낸다.

:first-of-type => 형제 요소의 그룹 중 해당 타입의 첫번째 요소를 나타낸다.

# 7. 가상 클래스(pseudo-class) 선택자

- 구조 선택자



```
1. C++
2. Java
3. C#
4. Python
5. JSP
6. JavaScript
```

- C++
- Java
- C#
- Python
- JSP
- JavaScript

```
<head>
  <style>
    /* ol 요소의 자식 요소인 li 요소 중에서 짝수 번째 요소만을 선택 */
    ol > li:nth-child(2n) { color: orange; }
    /* ol 요소의 자식 요소인 li 요소 중에서 홀수 번째 요소만을 선택 */
    ol > li:nth-child(2n+1) { color: green; }
    /* ol 요소의 자식 요소인 li 요소 중에서 첫번째 요소만을 선택 */
    ol > li:first-child { color: red; }
    /* ol 요소의 자식 요소인 li 요소 중에서 마지막 요소만을 선택 */
    ol > li:last-child { color: blue; }
    /* ol 요소의 자식 요소인 li 요소 중에서 4번째 요소 요소만을 선택 */
    ol > li:nth-child(4) { background: brown; }
    /* ul 요소의 모든 자식 요소 중에서 뒤에서부터 시작하여 홀수 번째 요소만을 선택 */
    ul > :nth-last-child(2n+1) { color: red; }
    /* ul 요소의 모든 자식 요소 중에서 뒤에서부터 시작하여 짝수 번째 요소만을 선택 */
    ul > :nth-last-child(2n) { color: blue; }
  </style>
</head>
<body>
  <ol>
    <li>C++ </li> <li>Java </li> <li>C# </li> <li>Python </li> <li>JSP </li> <li>JavaScript </li>
  </ol>
  <ul>
    <li>C++ </li> <li>Java </li> <li>C# </li> <li>Python </li> <li>JSP </li> <li>JavaScript </li>
  </ul>
</body>
```



# 7. 가상 클래스(pseudo-class) 선택자

- 구조 선택자

This is a heading

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

This is a heading

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

<body>

<h1>This is a heading</h1>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

<p>The third paragraph.</p>

<p>The fourth paragraph.</p>

<div>

<h1>This is a heading</h1>

<p>The first paragraph.</p>

<p>The second paragraph.</p>

<p>The third paragraph.</p>

<p>The fourth paragraph.</p>

</div>

</body>

<head>

<style>

/\* p 요소의 부모 요소의 자식 요소 중 첫번째 등장하는 p 요소 \*/

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

/\* p 요소의 부모 요소의 자식 요소 중 마지막 등장하는 p 요소 \*/

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

/\* p 요소의 부모 요소의 자식 요소 중 앞에서 2번째에 등장하는 p 요소 \*/

p:nth-of-type(2) { color: green; }

/\* p 요소의 부모 요소의 자식 요소 중 뒤에서 2번째에 등장하는 p 요소 \*/

p:nth-last-of-type(2) { color: orange; }

/\* p 요소가 첫번째 자식이면 배경색으로 brown 적용, p요소는 첫번째 자식이 아니므로 선언된 스타일이 적용되지 않음 \*/

p:first-child { background: brown; }

/\* div요소의 자식 요소인 p 요소 중에 마지막 자식이 아닌 요소 선택 \*/

div>p:not(:last-child) { background: yellow; }

</style>

</head>

/\* 아래의 CSS3 선언문 추가 \*/

h1:first-child {color :lightseagreen;}

This is a heading

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

This is a heading

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

# 7. 가상 클래스(pseudo-class) 선택자

- 구조 선택자 사용시 주의 사항

```
<head>
  <style>
    /*
    li>a:first-child { color : red;}
    */
    li:first-child >a { color : red;}
  </style>
</head>
<body>
  <li><a href="#">구조 선택자 사용 주의 사항</a></li>
  <li><a href="#">구조 선택자 사용 주의 사항</a></li>
  <li><a href="#">구조 선택자 사용 주의 사항</a></li>
  <li><a href="#">구조 선택자 사용 주의 사항</a></li>
</body>
```

- 구조 선택자 사용 주의 사항
- 구조 선택자 사용 주의 사항
- 구조 선택자 사용 주의 사항
- 구조 선택자 사용 주의 사항

- 구조 선택자 사용 주의 사항
- 구조 선택자 사용 주의 사항
- 구조 선택자 사용 주의 사항
- 구조 선택자 사용 주의 사항

# 8. 가상 요소(pseudo-elements) 선택자

- 문자 선택자
  - 요소 콘텐츠의 첫 글자 또는 첫 줄
  - 요소 콘텐츠의 앞 또는 뒤
  - 가상 요소에는 두개의 콜론(::)을 사용

형태	설명
::first-letter	콘텐츠의 첫 글자 선택 - 블록 요소에만 적용
::first-line	콘텐츠의 첫 줄 선택 - 블록 요소에만 적용
::after	콘텐츠의 뒤에 위치하는 공간 선택 - 일반적으로 content 속성과 함께 사용
::before	콘텐츠의 앞에 위치하는 공간 선택 - 일반적으로 content 속성과 함께 사용
::selection	해당 요소에서 사용자가 드래그한 글자 선택

# 8. 가상 요소(pseudo-elements) 선택자

- 문자 선택자

HTML!!! 문자 선택자 CSS3!!!

Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
tempore, repellat rerum et ea laborum voluptatum! Quisquam

가상 요소 ⇒ :: (더블 콜론) 사용

```
<head>
<style>
  /* p 요소 콘텐츠의 첫 글자를 선택 */
  p::first-letter { font-size: 3em; }
  /* p 요소 콘텐츠의 첫 줄을 선택 */
  p::first-line { color: red; }
  p::selection { background-color: aqua; color: green; }

  /* h1 요소 콘텐츠의 앞 공간에 content 어트리뷰트 값을 삽입한다 */
  h1::before {
    content: " HTML!!! ";
    color: blue;
  }
  /* h1 요소 콘텐츠의 뒷 공간에 content 어트리뷰트 값을 삽입한다 */
  h1::after {
    content: " CSS3!!!";
    color: red;
  }
</style>
</head>
<body>
  <h1>문자 선택자</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.<br>
    tempore, repellat rerum et ea laborum voluptatum! Quisquam
  </p>
</body>
```

# 학습 정리

---

- 스타일 시트 구성 요소는 반드시 정리하세요
  - 선택자(셀렉터) { 프로퍼티(속성): 값; } /\* 주석문 \*/
  - 스타일 선언 후 세미콜론
  - 콤마로 선택자를 나열하여 선언하면 다수의 요소에 동일한 스타일 적용
- 스타일 시트 만드는 3가지 방법을 정리하세요
- 스타일 적용 우선순위도 정리하세요
- 선택자는 한번 정리하면 여러 곳에서 유용하게 활용할 수 있습니다.
- CSS3 단위 중 em, vw, vh는 반응형 웹 구현을 위해 필요합니다.

# Q & A

---

- CSS3 소개와 선택자에 대한 학습이 모두 끝났습니다.
- 모든 내용을 이해 하셨나요?
- 아직 이해가 안되는 내용이 있다면 다시 한번 복습하시기 바랍니다.
- 질문은 한림 SmartLEAD 쪽지 또는 e-mail 또는 전화상담을 이용하시기 바랍니다.
- 다음 시간에는 “CSS3 속성”에 대하여 공부하도록 하겠습니다.
- 수고하셨습니다.^^