

# 06. CSS 기본

06-1 웹 문서에 디자인 입히기

06-2 스타일과 스타일 시트

06-3 CSS 기본 선택자

06-4 캐스케이딩 스타일 시트



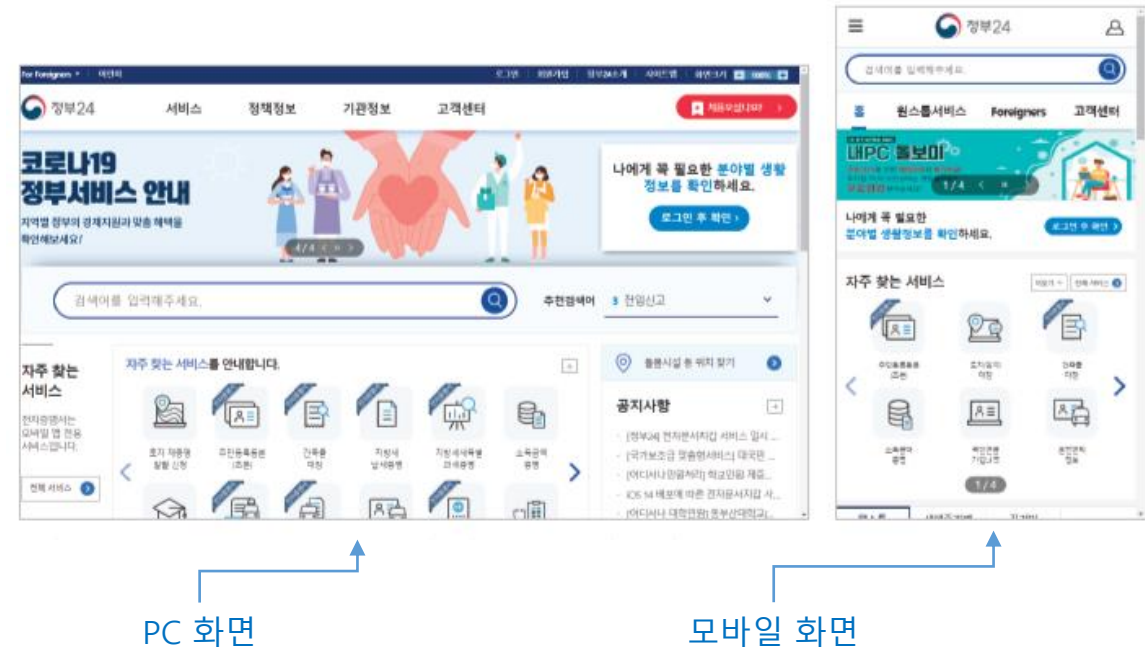
# 웹 문서에 디자인 입히기

## 왜 스타일을 사용할까

- HTML로는 웹 사이트의 내용을 나열하고 CSS로는 웹 문서의 디자인을 구성  
→ 스타일을 사용하면 웹 문서의 내용과 상관없이 디자인만 바꿀 수 있다



- 다양한 기기에 맞게 탄력적으로 바뀌는 문서를 만들 수 있다  
→ 반응형 웹 디자인



# 스타일과 스타일 시트

## 스타일 형식 알아보기

기본형 선택자 { 속성1: 속성값1; 속성2: 속성값2 }

스타일 규칙

텍스트 단락의  
글자를 가운데로 정렬  
글자색을 파랑으로  
지정하고 싶다면

### <p> 태그에 스타일 적용하기

선택자 → p {  
텍스트 정렬 속성 → text-align: center;  
글자색 속성 → color: blue;  
값 → center, blue

## 스타일 규칙 작성 방법

- 중괄호( { } ) 사이에 스타일 규칙 나열
- 규칙이 여러 개일 경우 세미콜론(;)으로 구분



### CSS 여러 줄로 표기하기

```
p {  
  text-align: center;  
  color: blue;  
}
```



### CSS 한 줄로 표기하기

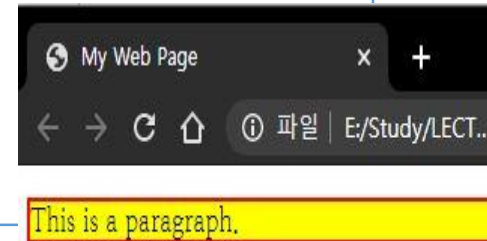
```
p { text-align: center; color: blue; }
```

## 스타일 주석

- /\*와 \*/ 사이에 주석 내용 입력
- 한 줄 또는 여러 줄을 입력 가능

# CSS의 형태

```
<!doctype html>
<html>
<head>
  <title>My Web Page</title>
  <style>
    p {
      background-color: yellow;
      border: 2px solid red;
    }
  </style>
</head>
<body>
  <p>This is a paragraph.</p>
</body>
</html>
```



# 스타일과 스타일 시트

## 스타일 시트



## 브라우저 기본 스타일

- 브라우저에서 기본으로 적용하는 스타일
- 웹 문서에서 아무 스타일도 적용하지 않고 HTML만 사용해도 그 기능에 따라 크기에 맞게 보여줌

## 인라인 스타일

- 스타일 시트를 사용하지 않고 스타일을 적용할 대상에 직접 표시
- 스타일을 적용하고 싶은 태그에 style 속성을 사용해 **style="속성: 속성 값;"** 형태로 스타일 적용

## 내부 스타일 시트

- 웹 문서 안에서 사용할 스타일을 문서 안에 정리한 것
- 모든 스타일 정보는 <head> 태그와 </head> 태그 안에서 정의
- <style> 태그와 </style> 태그 사이에 작성

## 외부 스타일 시트

- 여러 웹 문서에서 사용할 스타일을 별도 파일로 저장해 놓고 필요할 때마다 파일에서 가져와 사용
- <style> 태그 없이 <link> 태그만 사용해 미리 만들어 놓은 외부 스타일 시트 파일 연결

## CSS 삽입 위치

- 인라인 스타일 시트(inline)
  - 각각의 HTML요소마다 스타일 지정
  - 2개이상 선언이 있다면 반드시 끝에 ;을 적어준다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h1 style="color: red; text-decoration: underline;">This is a headline.</h1>
  <p style="color: #0026ff">This is a paragraph.</p>
</body>
</html>
```



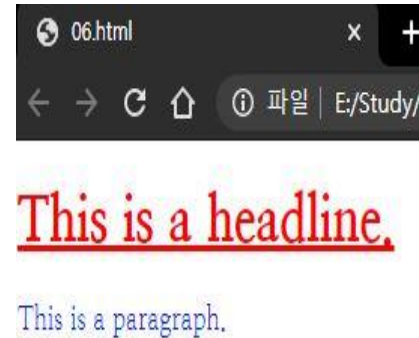
This is a headline.

This is a paragraph.

# CSS 삽입 위치

- 내부 스타일 시트 (internal style sheet)
  - 내부 스타일 시트는 HTML안에 CSS를 정의하는 것이다.

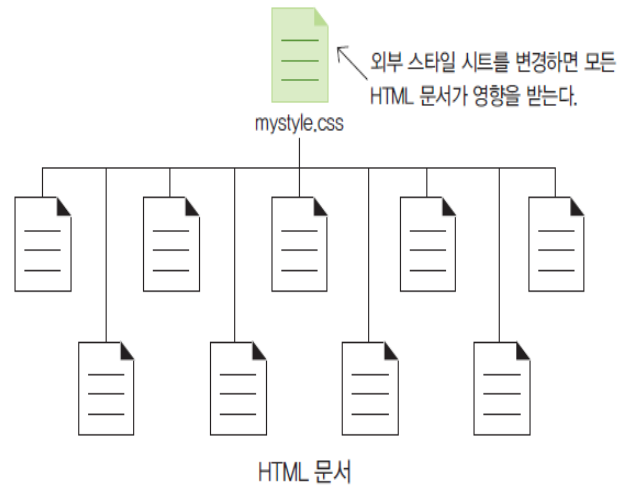
```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 {
      color: red;
      text-decoration: underline;
    }
    p {
      color: #0026ff;
    }
  </style>
</head>
<body>
  <h1>This is a headline.</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```



# CSS 삽입 위치

- 외부 스타일 시트 (external style sheet)
  - 스타일 시트를 HTML문서 외부, 즉 별도의 파일로 저장하는 방식이다.
  - 유지보수라는 측면에서 가장 권장하는 방식이다.

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



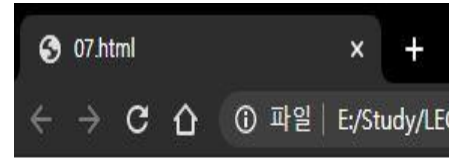


# CSS 삽입 위치

- 외부 스타일 시트 (external style sheet)

external.css

```
h1 {  
  color: red;  
  text-decoration: underline;  
}  
p {  
  color: #0026ff;  
}
```



This is a headline.  
This is a paragraph.

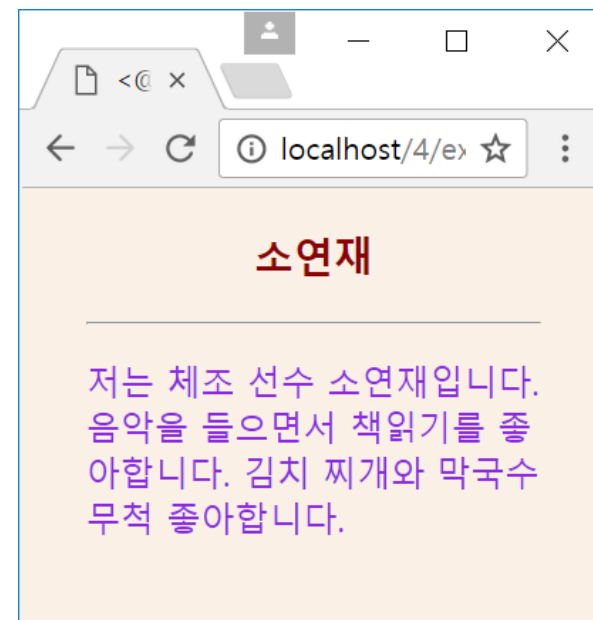
```
<!DOCTYPE html>  
<html>  
<head>  
  <link type="text/css" rel="stylesheet" href="external.css">  
</head>  
<body>  
  <h1>This is a headline.</h1>  
  <p>This is a paragraph.</p>  
</body>  
</html>
```

# MyStyle.html @import로 CSS3 파일 불러오기

## mystyle.css

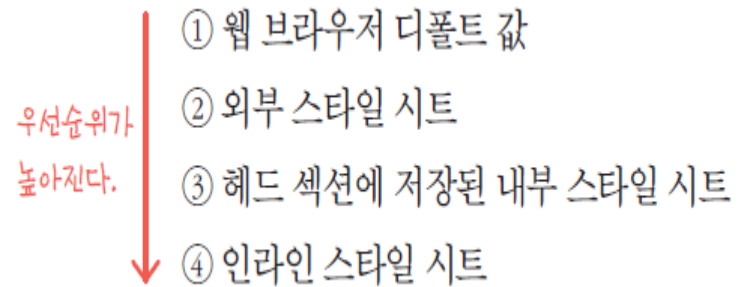
```
/* mystyle.css */
body { background-color : linen; color : blueviolet;
      margin-left : 30px; margin-right : 30px; }
h3 { text-align : center; color : darkred; }
```

```
<!DOCTYPE html>
<html>
<head>
<title>&lt;@import&gt;로 외부 스타일 불러오기</title>
<style>
  @import url(mystyle.css);
</style>
</head>
<body>
<h3>소연재</h3>
<hr>
<p>저는 체조 선수 소연재입니다. 음악을 들으면서 책읽기를
좋아합니다. 김치 찌개와 막국수 무척 좋아합니다.</p>
</body>
</html>
```



# CSS 삽입 위치

- 다중 스타일 시트
  - 하나의 요소에 대해서 내부, 외부, 인라인이 서로 다르게 적용되면 어떻게 될까??



- 가급적 외부 스타일 시트로 빼는 것이 좋다.



# CSS 기본 선택자

## 전체 선택자

- 페이지에 있는 모든 요소를 대상으로 스타일을 적용할 때 사용
- 웹 브라우저의 기본 스타일을 초기화할 때 자주 사용

기본형 \* { 속성: 값; ..... }

```
<style>
* {
  margin: 0;
}
</style>
</head>
<body>
  
</body>
```



## 타입 선택자

- 문서에서 특정 태그를 사용한 모든 요소에 스타일이 적용됨

기본형 태그명 { 스타일 규칙 }

```
<style>
p {
  font-style: italic; /* 이탤릭체 */
}
</style>
<div>
  <h1>레드향</h1>
  <p>검질에 붉은 빛이 돌아 레드향이라 불린다.</p>
  <p>레드향은 한라봉과 귤을 교배한 것으로 ...</p>
  <p>비타민 C와 비타민 P가 풍부해 ...</p>
</div>
```

### 레드향

검질에 붉은 빛이 돌아 레드향이라 불린다.

레드향은 한라봉과 귤을 교배한 것으로 일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.

비타민 C와 비타민 P가 풍부해 혈액순환, 감기예방 등에 좋은 것으로 알려져 있다.

# CSS 기본 선택자

## class 선택자

- 요소의 특정 부분에만 스타일 적용
- 마침표(.) 다음에 클래스 이름 지정
- 문서 안에서 여러 번 반복할 스타일이라면 클래스 선택자로 정의

기본형 .클래스명 { 스타일 규칙 }

```
<style>
  p {
    font-style: italic; /* 이탤릭체 */
  }
  .accent {
    border: 1px solid #000; /* 테두리 */
    padding: 5px; /* 테두리와 내용 사이의 여백 */
  }
  .bg {
    background-color: #ddd; /* 배경색 */
  }
</style>
</head>
<body>
  <div>
    <h1 c = "accent bg">레드향</h1>
    <p>껍질에 붉은 빛이 돌아 <span class="a">레드향</span>이라 불린다.</p>
    <p>레드향은 한라봉과 귤을 교배한 것으로 ...</p>
    <p>비타민 C와 비타민 P가 풍부해 ...</p>
  </div>
```

accent와 bg 클래스 선택자를 사용

accent 클래스 선택자만 사용

### 레드향

껍질에 붉은 빛이 돌아 레드향이라 불린다.

레드향은 한라봉과 귤을 교배한 것으로 일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.

비타민 C와 비타민 P가 풍부해 혈액순환, 감기예방 등에 좋은 것으로 알려져 있다.

## id 선택자

- 요소의 특정 부분에만 스타일 적용
- 파운드(#) 다음에 id 이름 지정
- 문서 안에서 한번만 사용한다면 id 선택자로 정의

기본형 #아이디명 { 스타일 규칙 }

```
(... 생략 ...)
```

```
<style>
  #container {
    width: 500px; /* 너비 */
    margin: 10px auto; /* 중앙 배치 */
    padding: 10px; /* 테두리와 내용 사이 여백 */
    border: 1px solid #000; /* 테두리 굵기와 색깔 */
  }
</style>
.....
<div id="c">
  <h1>레드향</h1>
  (... 생략 ...)
```

### 레드향

껍질에 붉은 빛이 돌아 레드향이라 불린다.

레드향은 한라봉과 귤을 교배한 것으로 일반 귤보다 2~3배 크고, 과육이 붉고 통통하다.

비타민 C와 비타민 P가 풍부해 혈액순환, 감기예방 등에 좋은 것으로 알려져 있다.

# CSS 기본 선택자

## class 선택자와 id 선택자의 차이

**공통점** : 요소의 특정 부분에만 스타일 적용

### 차이점

- 클래스 선택자 : 문서 안에서 여러 번 반복
- id 선택자 : 문서 안에서 한번만 사용

## 그룹 선택자

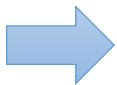
- 같은 스타일을 사용하는 선택자를 한꺼번에 정의
- 쉼표(,)로 구분해 여러 선택자를 나열

기본형 선택자1, 선택자2 { 스타일 규칙 }



<h1>, <p> 태그를 중앙에 정렬하기

```
h1 {  
  text-align: center; /* 중앙 정렬 */  
}  
p {  
  text-align: center; /* 중앙 정렬 */  
}
```



그룹 선택자를 사용하여 <h1>, <p> 태그를 중앙에 정렬하기

```
(... 생략 ...)  
h1, p {  
  text-align: c; /* 중앙 정렬 */  
}  
(... 생략 ...)
```

text-align:center

# <style> 태그에 스타일 시트 만들기

- 전형적인 모양

```
<head>
  <style>
    body { background-color : mistyrose; }
    h3 { color : purple; }
  </style>
  <style>
    hr { border : 5px solid yellowgreen; }
    span { color : blue; font-size : 20px; }
  </style>
</head>
```

- <style> 태그는 <head> 태그 내에서만 사용
- <style> 태그는 여러 번 작성 가능
  - 스타일 시트들이 합쳐 사용
- <style> 태그에 작성된 스타일 시트는 웹 페이지 전체에 적용



# 캐스케이딩 스타일 시트

## 캐스케이딩의 의미

- 캐스케이딩(Cascading) : '위에서 아래로 흐른다'는 뜻. 즉 계단식으로 적용된다는 의미로 사용.
- 선택자에 여러 스타일이 적용될 때 스타일 충돌을 막기 위해 우선순위에 따라 적용할 스타일을 결정함.

## 스타일 충돌을 막는(캐스케이딩)의 원칙

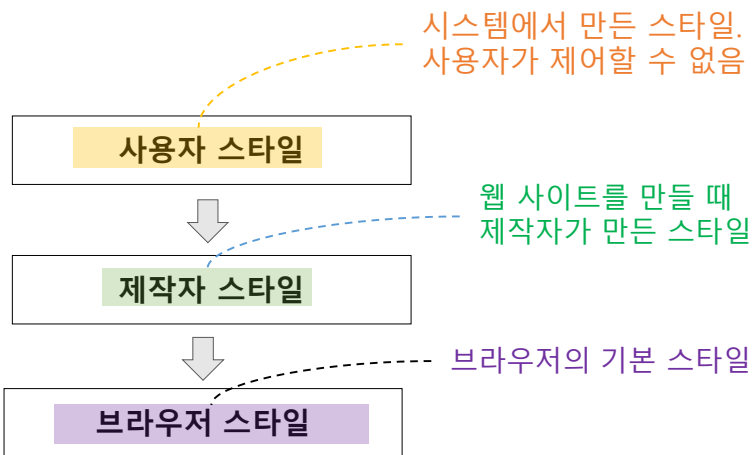
- ① 스타일 우선순위 - 스타일 규칙의 중요도와 적용 범위에 따라 우선순위가 결정되고 그 우선순위에 따라 위에서 아래로 스타일 적용
- ② 스타일 상속 - 태그들의 포함 관계에 따라 부모 요소의 스타일을 자식 요소로, 위에서 아래로 전달

※ 스타일 시트에서 '캐스케이딩'은 가장 기본적인 개념이기 때문에 일반적으로 '스타일 시트'는 '캐스케이딩 스타일 시트(CSS)'와 같은 의미로 사용됨

# 캐스케이딩 스타일 시트

## 원칙 1: 스타일 우선 순위

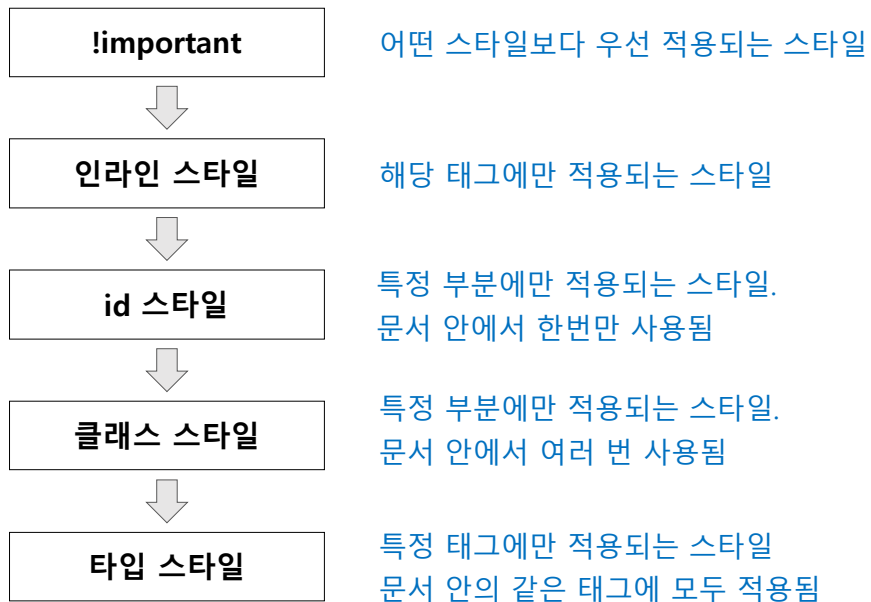
### 1. 얼마나 중요한가에 따라



### 3. 소스 순서에 따라

- 중요도와 명시도가 같다면 소스 순서에 따라 결정
- 소스에서 나중에 온 스타일이 먼저 온 스타일을 덮어씀

### 2. 얼마나 한정지를 수 있는가에 따라



# 캐스케이딩 스타일 시트

## 원칙 1: 스타일 우선 순위

Priority.html

### 이지스퍼블리싱의 미션

사람에게 구체적으로 도움을 주는 책,  
우리는 열심히 사는 사람들에게 도움이 되고 싶습니다.

우리는 책을 출간하기 전에 질문할 것입니다.

"이 책이 사람들에게 도움이 됩니까?"

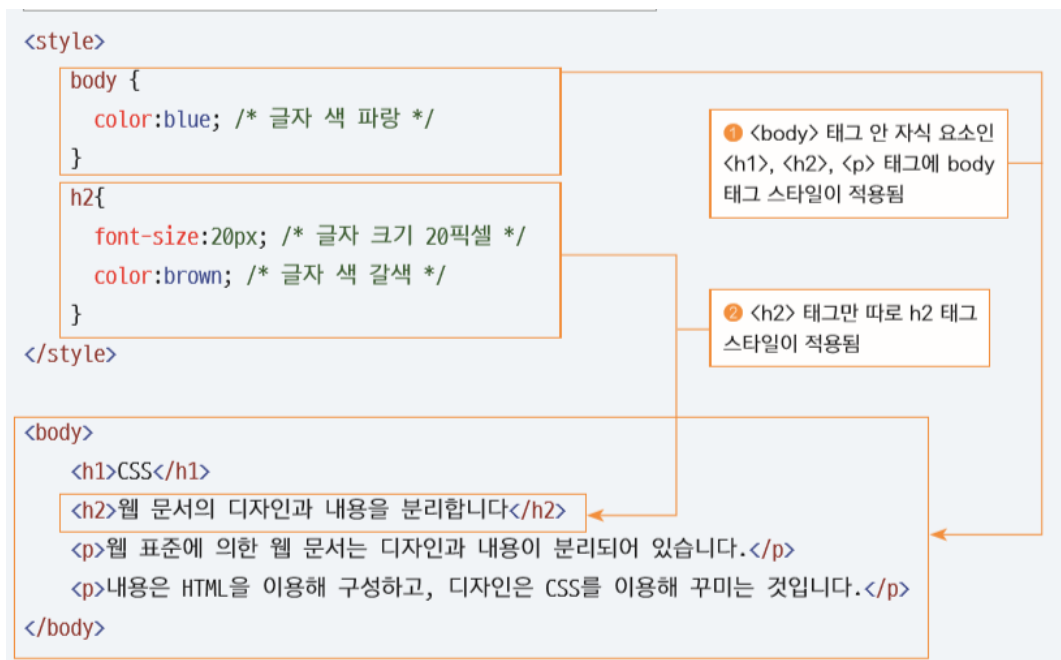


# 캐스케이딩 스타일 시트

## 원칙 2: 스타일 상속

- 자식 요소에서 별도로 스타일을 지정하지 않으면 부모 요소에 있는 스타일 속성들이 자식 요소로 전달됨.
- 상속을 이용하면 스타일 시트를 효과적으로 만들 수 있다
- 주의할 것은 스타일의 모든 속성이 부모 요소에서 자식 요소로 상속되는 것은 아니라는 점  
(예, 글자 색은 상속되지만 배경 색은 상속되지 않음)

### Inherit.html



# CSS3와 CSS 모듈

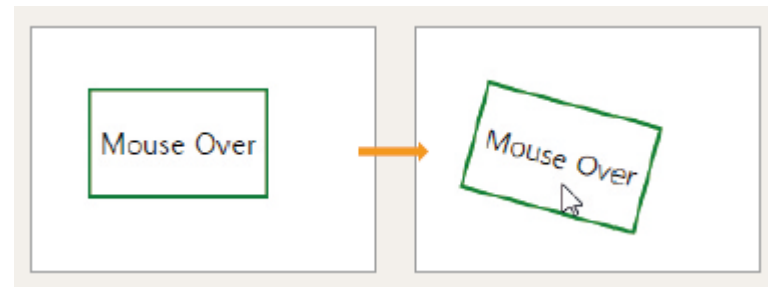
## CSS3와 브라우저 접두사

- CSS3 모듈이 계속 개발되고 있는데 표준 규약이 아닌 속성들은 브라우저에 따라 다른 방식으로 지원됨  
→속성 이름 앞에 접두사(prefix)를 붙여 브라우저별로 구분해야

크롬, 파이어폭스, 오페라, 사파리 등  
웹 표준을 지원하는 브라우저.

- 표준 규약이 완성된 속성도 옛날 버전의 **모던 브라우저** 사용자를 고려하기 위해 브라우저 접두사를 붙여 사용하기도 함

접두사	설명
-webkit-	웹킷 방식 브라우저용(사파리, 크롬 등)
-moz-	게코 방식 브라우저용(모질라, 파이어폭스 등)
-o-	오페라 브라우저
-ms-	마이크로소프트 인터넷 익스플로러



```
<style>
  .box:hover {
    -webkit-transform:rotate(15deg);
    -moz-transform:rotate(15deg);
    -o-transform:rotate(15deg);
    -ms-transform:rotate(15deg);
    transform:rotate(15deg);
  }
</style>
</head>

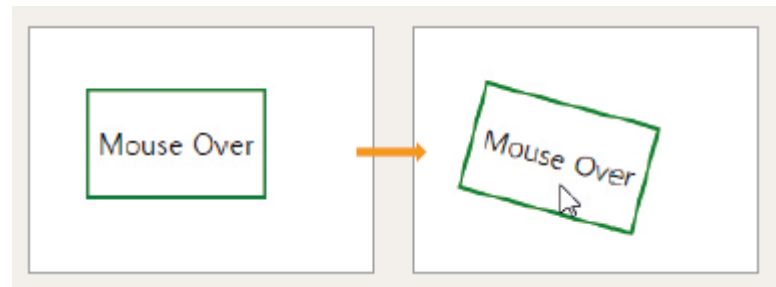
<body>
  <div class="box">Mouse Over</div>
</body>
```

# CSS3와 CSS 모듈

## 브라우저 접두사를 자동으로 붙여준다

**-prefix-free.js** : 브라우저 벤더 접두사를 자동으로 붙여줌.

1. <http://leaverou.github.io/prefixfree/> 에서 파일 다운로드
2. prefixfree.min.js 파일을 원하는 곳으로 복사하거나 옮김
3. `<script>` 태그를 이용해 웹 문서에 삽입  
`<script src="prefixfree.min.js"></script>`
4. 이제부터는 브라우저 접두사 없이 CSS3 속성 사용



```
<head>
  <script src="prefixfree.min.js"></script>
  <style>
    .box:hover {
      transform: rotate(15deg);
    }
  </style>
</head>
<body>
  <div class="box">Mouse Over</div>
</body>
```