

# HTML5 & CSS3

H t m l 5 & c s s 3



Copyright 2018.

안효인

All rights reserved.

무단 복제와 사용을 금합니다.



안효인

troment@nate.com

# Index

## CONTENTS



01 HTML5 & CSS3  
소개



03 HTML 마크업  
요소



05 CSS 이해



07 멀티미디어



02 HTML 기본



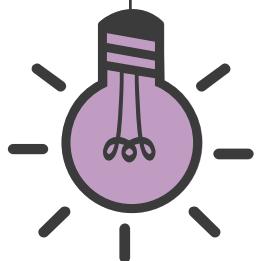
04 CSS 기본



06 HTML5와  
semantic

01

## HTML5 & CSS3 소개





# 1. HTML5 & CSS3 소개

## 1-1. HTML 개요

- ✓ HTML은 Hypertext Markup Language의 약자.
- ✓ 1990년도 이후 웹(Web, World Wide Web)에서 사용하는 문서 양식.
- ✓ 문서에 하이퍼텍스트, 표, 목록, 비디오 등을 포함할 수 있는 tag(Tag)를 사용.
- ✓ 문서를 웹브라우저에 표현할 때 tag를 사용

Tag 적용 전

```
<제목>회원정보</제목>
<이미지 이미지소스="image/mypic.png" 설명="증명사진"/>
<정의목록>
    <용어정의>이름 : </용어정의>
    <용어설명>홍길동</용어설명>
    <용어정의>이메일 : </용어정의>
    <용어설명>honggd@naver.com</용어설명>
</정의목록>
```

Markup  
적용

HTML

```
<h1>회원정보</h1>

<dl>
    <dt>이름 : </dt>
    <dd>홍길동</dd>
    <dt>이메일 : </dt>
    <dd>honggd@naver.com</dd>
</dl>
```



# 1. HTML5 & CSS3 소개

## 1-2. HTML5 개발 배경.

- ✓ 수많은 플러그인들로 인한 브라우저간의 부작용을 막기위해 개발.
- ✓ **HTML5 개발 단계**
  - 익스플로러 독점.
  - 익스플로러에 맞는 많은 플러그인 개발.
  - 거대해져가는 브라우저 및 보안 취약점 발생.
  - MS를 제외한 회사들이 HTML표준안을 W3C에 제시.
  - HTML 표준안 무시.
  - 무시당한 회사들이 Web Application 1.0 개발.
  - W3C는 XHTML2.0 개발.
  - XHTML2.0의 쇠퇴.
  - W3C에서 Web Application 1.0을 웹표준안으로 수용.
  - Web Application 1.0을 HTML5로 계명.
  - 2014년 HTML5로 정식 배포.
- ✓ **HTML5의 탄생 목적은 기존의 웹브라우저에 많은 플러그인이 설치되면서 많은 문제점이 발생했고, 이를 해결하기 위한 방안으로 W3C에서 Web Application1.0을 HTML5로 수용. 2014년에 정식으로 배포되었으며, 계속 업그레이드되고 있다.**



# 1. HTML5 & CSS3 소개

## 1-3. 웹표준이란?

- ✓ 모든 브라우저에서 웹서비스가 정상적으로 보여질 수 있도록 하는 것.
- ✓ W3C(World Wide Web Consortium) – <http://www.w3.org>
  - 월드와이드웹(worldwideweb)의 창시자인 팀 버너스 리를 중심으로 창립된 월드와이드웹 컨소시엄.
- ✓ W3C에서 HTML5를 웹 표준으로 권고하고 웹 브라우저는 이를 따름.





# 1. HTML5 & CSS3 소개

## 1-4. HTML5의 특징

- ✓ **HTML5는 지금도 개발 중에 있고, 다양한 기능이 추가됨.**
- ✓ **멀티미디어 요소 재생.**
  - 과거 브라우저는 멀티미디어를 재생하기 위해 별도의 외부 플러그인을 사용해야 했으나, HTML5에서는 멀티미디어 요소를 별도의 플러그인 없이도 재생가능.
- ✓ **서버와 통신.**
  - 서버와 클라이언트 사이에 소켓 통신이 가능.
- ✓ **Semantic tag 추가.**
  - 웹사이트를 검색엔진이 좀더 빠르게 검색할 수 있도록 하기 위해 특정tag에 의미를 부여하는 방식.
  - 예를 들어 <header>tag는 문서의 주제를 나타내는 tag로써 <header> tag가 사용된 웹 문서의 경우 검색 엔진은 웹문서의 모든 내용을 검색하는 것이 아닌 <header> tag의 내용만을 검색해서 보다 빠르게 검색을 진행할 수 있다.



# 1. HTML5 & CSS3 소개

## 1-5. HTML 문서 구조

- ✓ `<!DOCTYPE html>` tag는 현재 문서가 HTML문서임을 정의.
- ✓ 시작tag(`<tagnname>`)와 종료tag(`</tagnname>`)가 있으며, tag 사이에 문서 내용을 정의.
- ✓ 각 tag는 고유의 의미를 가지고 있으며, 웹브라우저는 이 의미에 따라 문서를 화면에 표시.

HTML Code

```
<!DOCTYPE html>                                문서type정의(DTD) : HTML문서 정의
<html>
  <head>          문서머리글(HEAD) : 제목, 검색엔진에서 사용할 키워드, 기타정보
    <meta charset="UTF-8">
    <title>document title</title>
  </head>

  <body>                                         문서본체(BODY) : 문서의 내용
    document contents
  </body>
</html>
```

Result View





# 1. HTML5 & CSS3 소개

## 1-6. Web & HTML 작동원리

- ✓ 서버는 클라이언트의 요청 내용을 분석하여 결과값을 HTML로 전송.
- ✓ 서버는 결과값을 전송한 후 클라이언트와 연결 종료.
- ✓ 클라이언트는 서버로부터 전달받은 HTML을 웹브라우저에 표시.
- ✓ 각 웹브라우저는 브라우저 엔진이 내장되어 있고, 이 엔진이 tag를 해석하여 화면에 표현.





# 1. HTML5 & CSS3 소개

## 1-7. CSS 개요

- ✓ CSS(Cascading Style Sheets)란 HTML 문서를 화면에 표시하는 방식을 정의한 언어.
- ✓ W3C에서 공인한 표준.
- ✓ 기존 웹 문서를 다양하게 설계하고, 변경 요구 대응에 따르는 어려움을 보완.
- ✓ 어떤 Style을 적용 하느냐에 따라, 하나의 구조를 전혀 다른 페이지처럼 표현.

Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

**No Stylesheet**

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:  
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:  
[No Stylesheet](#)

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet enim ad minim veniam, quis nostrud exercit tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. In vulputate vell esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accusamus et iusto odio dignissim qui blandit praesent luptatum zzini denlet augue dui dolore te feugait nulla facilisi.

Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

**Stylesheet1**

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:  
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:  
[No Stylesheet](#)

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

**Stylesheet2**

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:  
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:  
[No Stylesheet](#)

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

**Stylesheet3**

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:  
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:  
[No Stylesheet](#)

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

**Stylesheet2**

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:  
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:  
[No Stylesheet](#)

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

**Stylesheet3**

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:  
[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:  
[No Stylesheet](#)

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

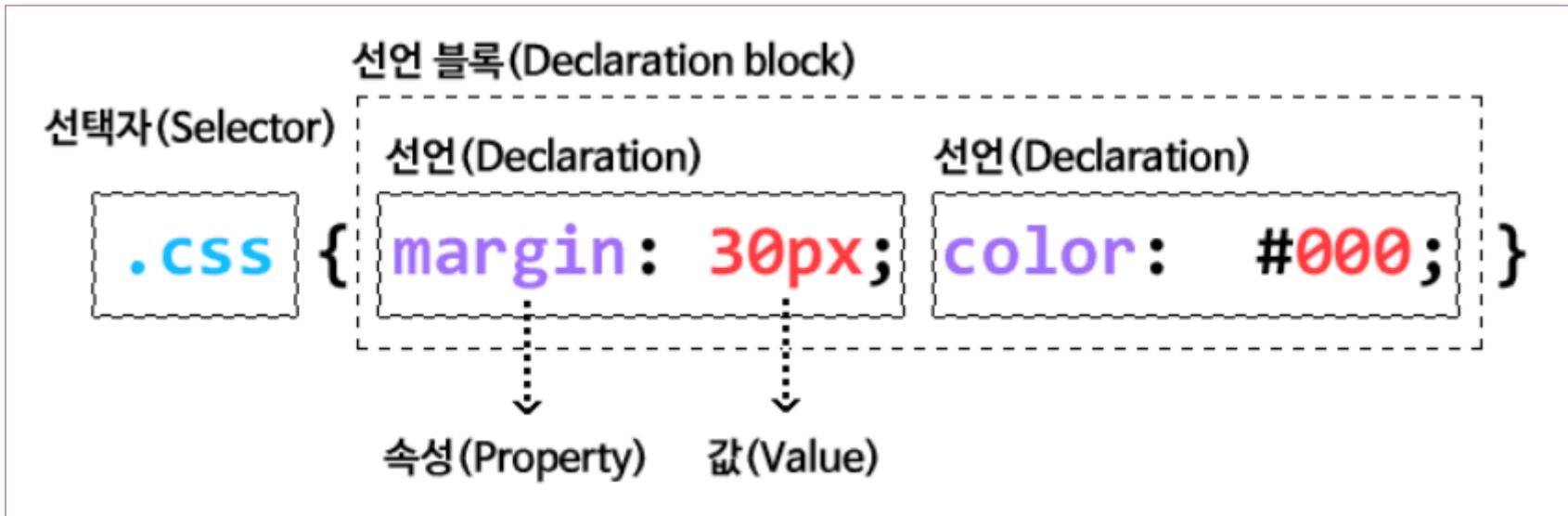
출처 :  
[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)



# 1. HTML5 & CSS3 소개

## 1-8. CSS규칙 (1/2)

- ✓ CSS 규칙은 선택자(selector)와 선언(declaration) 두 부분으로 구성.
- ✓ 선택자는 규칙이 적용되는 엘리먼트.
- ✓ 선언 부분에서는 선택자에 적용될 스타일을 작성.
- ✓ 선언은 중괄호로 감싸며, 속성(property)과 값(value)로 이루어짐.

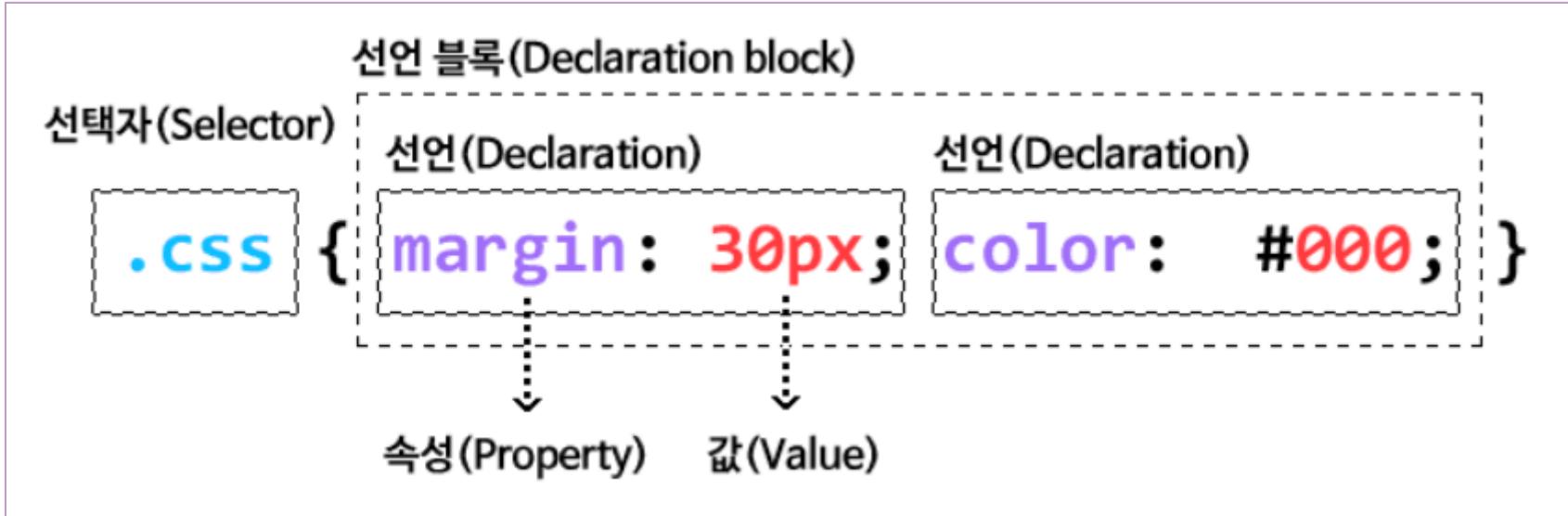




# 1. HTML5 & CSS3 소개

## 1-8. CSS규칙 (2/2)

- ✓ 속성(property)은 선택자에서 바꾸고 싶은 요소.(color, font, width, height, border, ....)
- ✓ 값(value)은 속성에 적용할 값.
- ✓ 여러 선택자에 동일한 스타일을 적용할 때, comma(,)로 구분하여 나열.(선택자 그룹화)
- ✓ 선언 안에 하나 이상의 속성을 작성할 수 있으며, 각 속성은 semi-colon(;)으로 구분.





# 1. HTML5 & CSS3 소개

## 1-9. 스타일 적용방법 (1/3) – 외부 스타일시트

- ✓ HTML문서에 스타일을 적용하는 방법은 외부 스타일시트, 내부 스타일시트, 인라인 스타일 3가지로 분류.
- ✓ 외부 스타일시트(External Style Sheet)는 \*.css파일을 <link>나 @import로 HTML문서에 연결하여 사용.
- ✓ 하나의 CSS파일만 수정하면 해당 스타일시트를 사용하는 모든 페이지에 변경내용 적용.
- ✓ 외부 스타일시트는 세가지 방법 중 가장 많이 사용.

### link

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

### @import

```
<style type="text/css">
@import url("path/filename.css"); /*@import "path/filename.css";*/
</style>
```



# 1. HTML5 & CSS3 소개

## 1-9. 스타일 적용방법 (2/3) – 내부 스타일시트

- ✓ 내부 스타일시트(Embedded Style Sheet)는 `<style>`을 이용하여 HTML페이지 내부에 CSS 적용.
- ✓ `<style>`은 `<head>`안에 작성.
- ✓ 내부 스타일 시트는 페이지마다 반복해서 작성하는 단점이 있다.
- ✓ 여러 페이지에 동일한 스타일을 적용해야 할 경우 외부 스타일시트를 사용.

```
<style type="text/css">
/* CSS 규칙 */
h1 {color: pink;}
.className {background-color: cyan; color: magenta;}
</style>
```



# 1. HTML5 & CSS3 소개

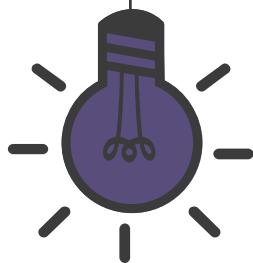
## 1-9. 스타일 적용방법 (3/3) – 인라인 스타일시트

- ✓ 인라인 스타일(Inline style)은 **style attribute**를 사용하여 개별 엘리먼트에 스타일을 적용.
- ✓ **Style 속성의 값은 CSS규칙의 선언(declaration)과 같다.**
- ✓ 하나 이상의 속성을 적용할 수 있으며, **semi-colon(;)으로 구분.**
- ✓ 세 가지 스타일을 모두 사용했을 때 가장 먼저 반영.

```
<body>
<h1>Inline Style</h1>
<p style="background-color: cyan; color: magenta;">인라인 스타일시트 적용!!!</p>
</body>
```

02

## HTML 기본





## 2. HTML 기본

### 2-1. HTML 개요

- ✓ HTML5 웹 문서를 구성하는 3가지 요소.

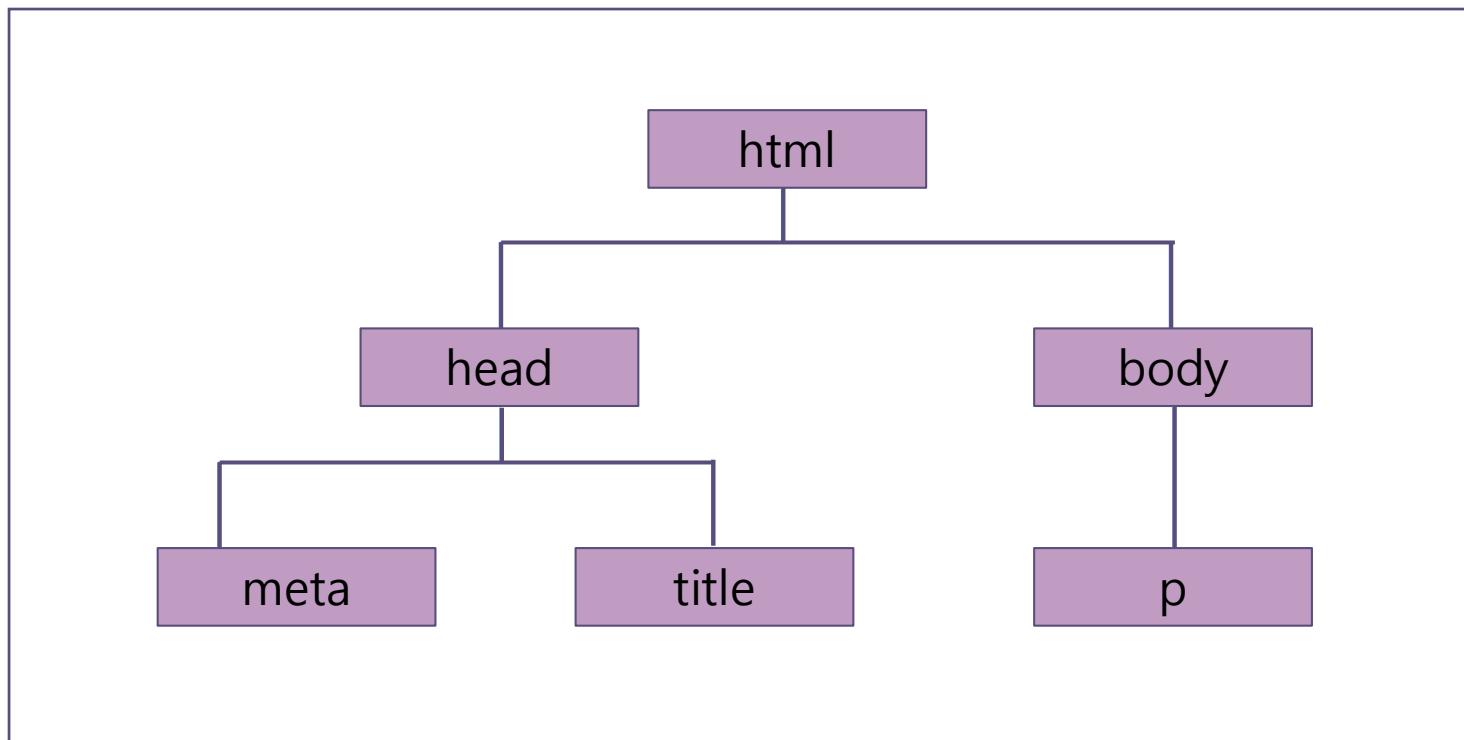




## 2. HTML 기본

### 2-1.HTML 개요

- ✓ HTML은 마크업 언어(markup language)로 웹 문서를 작성하며, tag를 사용하여 문서의 구조등을 기술하는 언어.



# 2. HTML 기본

## 2-2. tag와 속성 (1/3)

- ✓ HTML 문서는 'tag'로 만들어진다.
- ✓ HTML 문서의 전체 구성은 html, head, body tag로 구성.

The screenshot shows the W3Schools website with the URL <https://www.w3schools.com> in the address bar. The page title is "w3schools.com - THE WORLD'S LARGEST WEB DEVELOPER SITE". The main content area features a large "HTML" heading and the subtext "The language for building web pages". Below this, there are two sections: "HTML Example:" containing sample HTML code, and "CSS Example:" containing sample CSS code. At the bottom, there are "LEARN HTML" and "HTML REFERENCE" buttons.

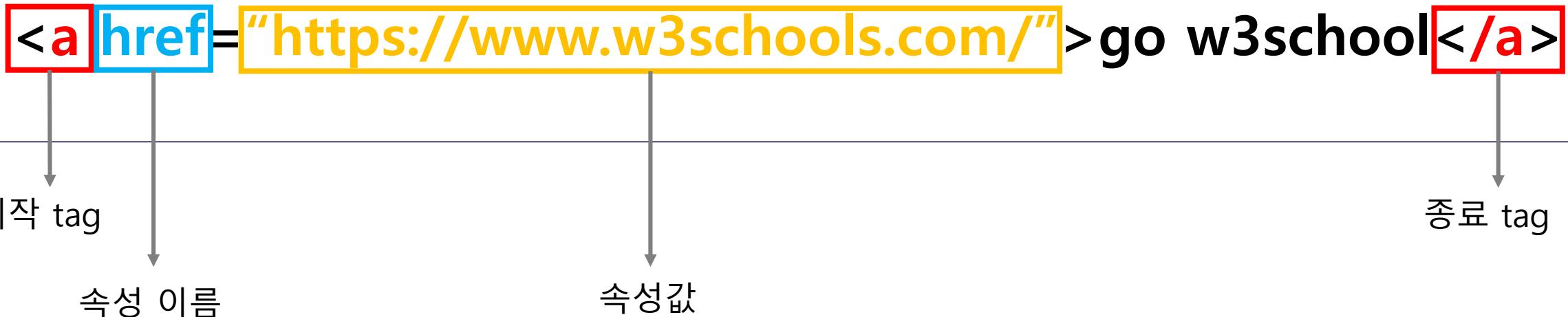
```
1 <!DOCTYPE html>
2 <html lang="en-US">
3   <head>
4     <title>W3schools Online Web Tutorials</title>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <meta name="Keywords" content="HTML,CSS,JavaScript,DOM,jQuery,PHP,SQL,XML,Bootstrap,Web,W3CSS,W3C,tutorials,programming,development,training,learning,quiz,primer,lessons,reference,examples,source code,colors,demos,tips,w3c">
8     <link rel="icon" href="/favicon.ico" type="image/x-icon">
9     <link rel="stylesheet" href="w3css/4/w3.css">
10    <script>
11      (function() {
12        var cx = '012971019331610648934:m2tou3_miyw';
13        var gcse = document.createElement('script'); gcse.type = 'text/javascript'; gcse.async = true;
14        gcse.src = 'https://www.google.com/cse/cse.js?cx=' + cx;
15        var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(gcse, s);
16      })();
17      (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
18        (i[r].q=i[r].q||[]).push(arguments),i[r].l=!new Date();a=s.createElement(o),
19        m=s.createElement('script');m.async=1;a.src=g;m.parentNode.insertBefore(a,m)
20      })(window,document,'script','https://www.google-analytics.com/analytics.js','ga');
21      ga('create', 'UA-3855518-1', 'auto');
22      ga('require', 'displayfeatures');
23      ga('send', 'pageview');
24    </script>
25    <style>
26      /* W3Schools Font Logo */
27      .w3schools-logo {
28        font-family:fontawesome;
29        text-decoration:none;
30        line-height: 1;
31        -webkit-font-smoothing: antialiased;
32        -moz-osx-font-smoothing: grayscale;
33        font-size:37px;
34        letter-spacing:3px;
35        color:#555555;
36        display:block;
37        position:relative;
38      }
39      .w3schools-logo .dotcom {color:#4CAF50;}
40      @font-face {
41        font-family:'fontawesome';
42        src:url('../lib/fonts/fontawesome.eot?14663396#iefix') format('embedded-opentype'),
43        url('../lib/fonts/fontawesome.woff?14663396') format('woff'),
44        url('../lib/fonts/fontawesome.ttf?14663396') format('truetype'),
45        url('../lib/fonts/fontawesome.svg?14663396#fontawesome') format('svg');
46        font-style:normal;
47      }
```

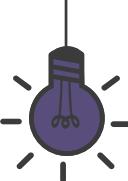


## 2. HTML 기본

### 2-2. tag와 속성 (2/3)

- ✓ tag는 시작 tag와 종료 tag로 쌍을 이루거나 시작 tag만 존재하는 tag도 있다.
- ✓ 시작 tag(<tagname>)와 종료 tag(</tagname>)는 '/'로 구분하며 중첩되지 않도록 한다.
  - 시작과 종료tag : <body> ... </body>
  - 시작 tag만 : <br/>, <hr/>, <img/>, ...
- ✓ 각각의 tag는 속성과 속성의 값이 존재한다.





## 2. HTML 기본

### 2-2. tag와 속성 (3/3)

- ✓ HTML tag에는 어느 tag에나 넣어서 사용할 수 있는 글로벌속성(global attribute)이 있다.

글로벌 속성	설명
class	tag에 적용할 스타일의 이름을 지정 ex) <div class="content">..</div>
dir	내용의 텍스트 방향을 지정. 왼쪽 >> 오른쪽(기본값, ltr), 오른쪽 >> 왼쪽(rtl). 왼쪽 또는 오른쪽 정렬과 비슷한 형태로 표현. ex)<p dir="rtl">오른쪽에서 왼쪽으로 표시됨.(오른쪽 맞춤)</p>
id	tag에 유일한 ID를 지정함. 자바스크립트에서 주로 사용. ex)<input type="text" id="userid">
style	인라인 스타일을 적용하기 위해 사용 ex)<p style="color: red; text-align: center;">빨간색 가운데</p>
title	tag에 추가 정보를 지정. tag에 마우스 포인터를 위치시킬 경우 title의 값 표시. ex)<p><abbr title="Web Application Server">WAS</abbr>는 ...</p>



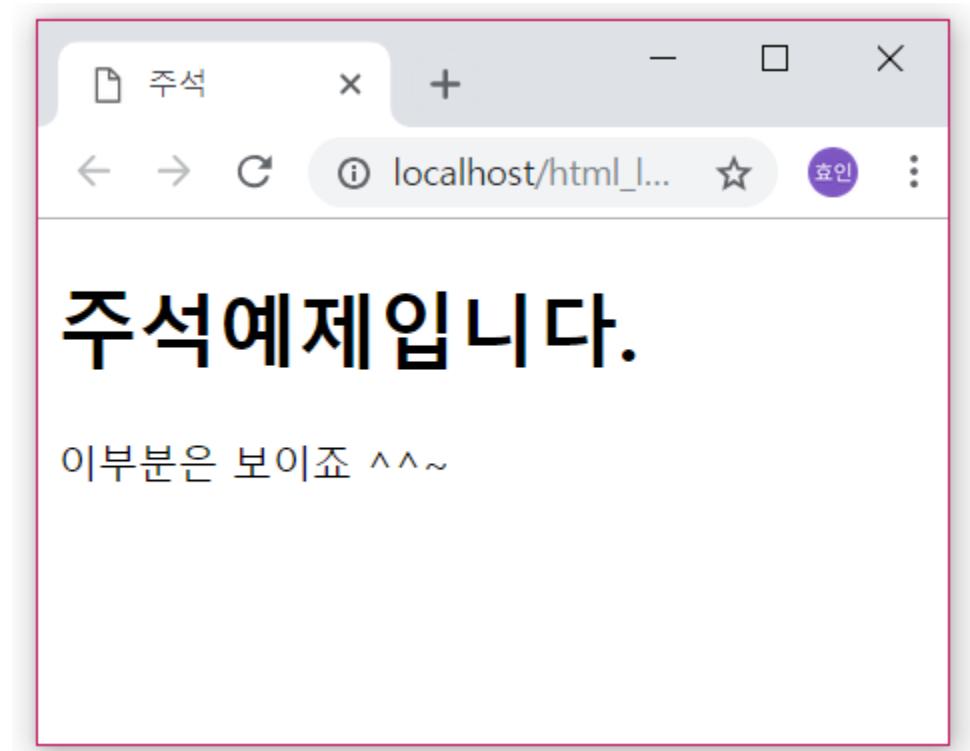
## 2. HTML 기본

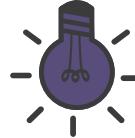
### 2-3. 주석

- ✓ 주석의 내용은 브라우저에 출력이 되지 않는다.
- ✓ HTML tag의 내용을 설명하기 위한 용도로 사용.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>주석</title>
  </head>
  <body>
    <h1>주석예제입니다.</h1>
    이부분은 보이죠 ^^~
    <!--
      주석은 내용에 포함되지 않습니다.
    -->
  </body>
</html>
```

2-1.html





## 2. HTML 기본

### 2-4. Root 요소

- ✓ <html> tag는 HTML 문서 전체를 정의.
- ✓ Head(<head>...</head>)와 Body(<body>..</body>)로 구성.

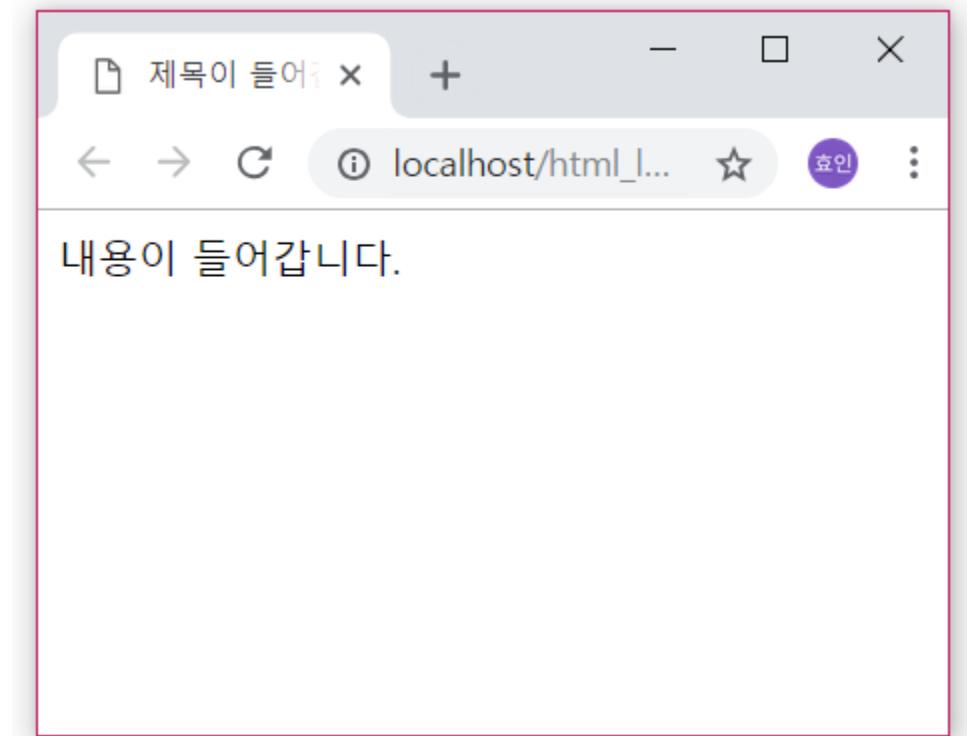
```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
        <title>제목이 들어갑니다.</title>
    </head>

    <body>
        내용이 들어갑니다.
    </body>

</html>
```

2-2.html



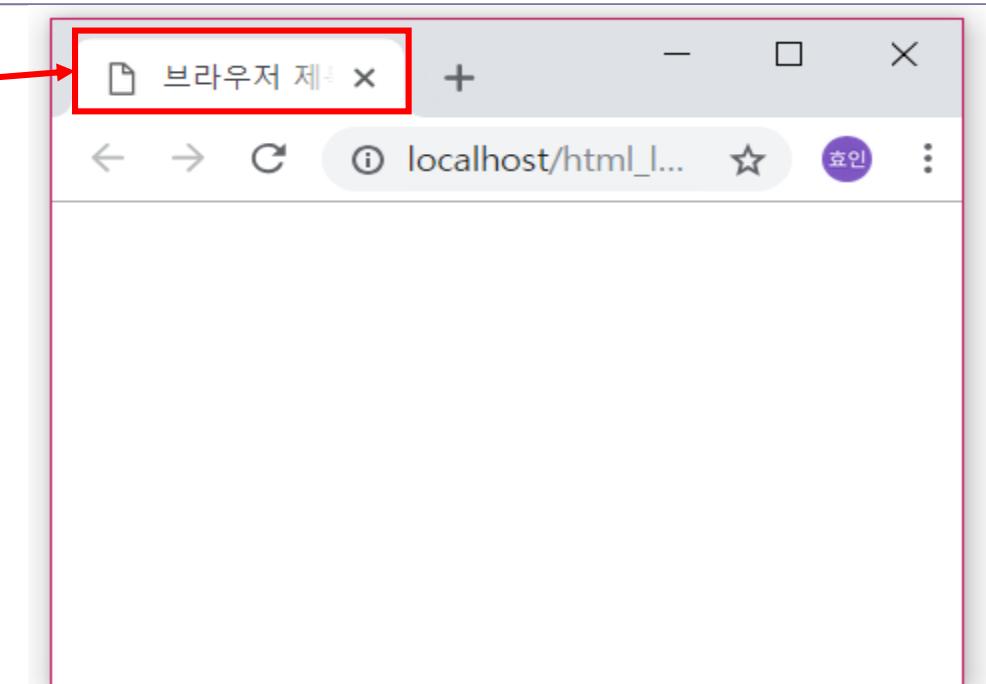
## 2. HTML 기본

### 2-5. Head 요소 (1/2) – 문서 머리글(head), 제목(title)

- ✓ <head> tag는 브라우저에게 HTML문서의 머리 부분임을 인식.
- ✓ <title>, <meta>, <style>, <script>, <link> tag를 포함 가능.
- ✓ <title> tag는 문서의 제목을 의미, 브라우저의 제목 표시줄에 tag 내용이 나타남.
- ✓ <title> tag 이외의 다른 tag로 표현한 정보는 화면에 출력 X.

```
<head>
  <meta charset="UTF-8">
  <title>브라우저 제목 표시줄에 나타남.</title>
  <meta name="author" content="troment">
  <meta name="description" content="간단한 설명.">
  <meta name="keyword" content="html5, web">
  <script type="text/javascript"></script>
  <style type="text/css"></style>
  <link rel="stylesheet" href="test.css">
</head>
<body>

</body>
```





## 2. HTML 기본

### 2-5. Head 요소 (2/2) – 메타 데이터(meta)

- ✓ 문서의 작성자, 날짜, 키워드등 브라우저의 본문에 나타나지 않는 일반 정보를 나타냄.
- ✓ name과 content 속성을 이용하여 다양한 정보를 나타냄.
- ✓ http-equiv 속성을 이용하여 인코딩 설정 및 문서 이동, 새로 고침이 가능.
- ✓ charset 속성을 이용하여 문서의 인코딩 정보를 설정.

```
<meta name="name" content="value">
```

- name 속성 : description(문서의 요약), keyword(검색어 입력, 콤마로 분리), author(제작자) 등.

```
<meta name="description" content="간단한 설명.">
```

- 페이지 설명. 검색엔진 로봇이 수집.

```
<meta name="keyword" content="html5, web">
```

- 페이지의 키워드를 ,로 구분해서 나열, 검색엔진 로봇이 수집.

```
<meta name="author" content="troment">
```

- 페이지 제작자 정보.

```
<meta http-equiv="refresh" content="30">
```

- http-equiv 속성 : refresh(문서를 자동으로 업데이트), content-type(인코딩 설정) 등.

```
<meta charset="UTF-8">
```

- 인코딩 정보를 설정.



## 2. HTML 기본

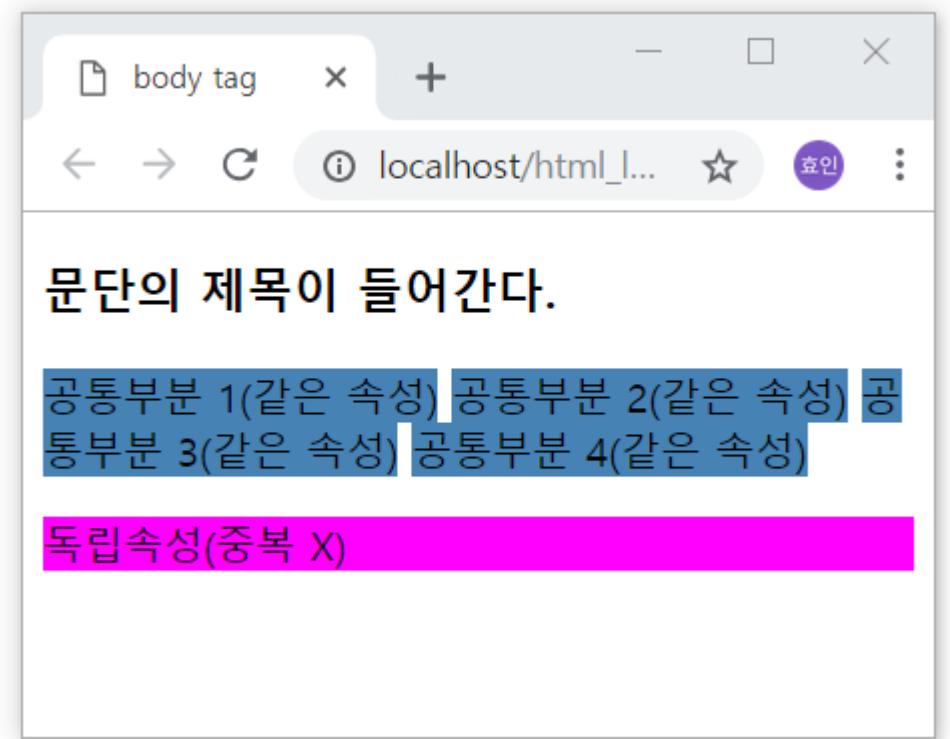
### 2-6. Body 요소 (1/2) - body

- ✓ 웹브라우저에 보여질 문서의 내용을 작성.
- ✓ <head> tag 다음에 위치하고 <head> 내부에 위치하는 tag와 <html>을 제외한 모든 tag.
- ✓ id 속성을 이용하여 문서 내에서 tag를 유일하게 식별 가능. (id속성은 중복 X)
- ✓ class 속성을 이용하여 여러 tag에 공통적인 특성(CSS)을 부여.(class속성은 중복 O)

```
<head>
<meta charset="UTF-8">
<title>body tag</title>

<style type="text/css">
.common { background-color: steelblue; color: white; }
#private { background-color: magenta; }
</style>
</head>
<body>
    <h3>문단의 제목이 들어간다.</h3>
    <span class="common">공통부분 1(같은 속성)</span>
    <span class="common">공통부분 2(같은 속성)</span>
    <span class="common">공통부분 3(같은 속성)</span>
    <span class="common">공통부분 4(같은 속성)</span>

    <p id="private">독립속성(중복 X)</p>
</body>
```





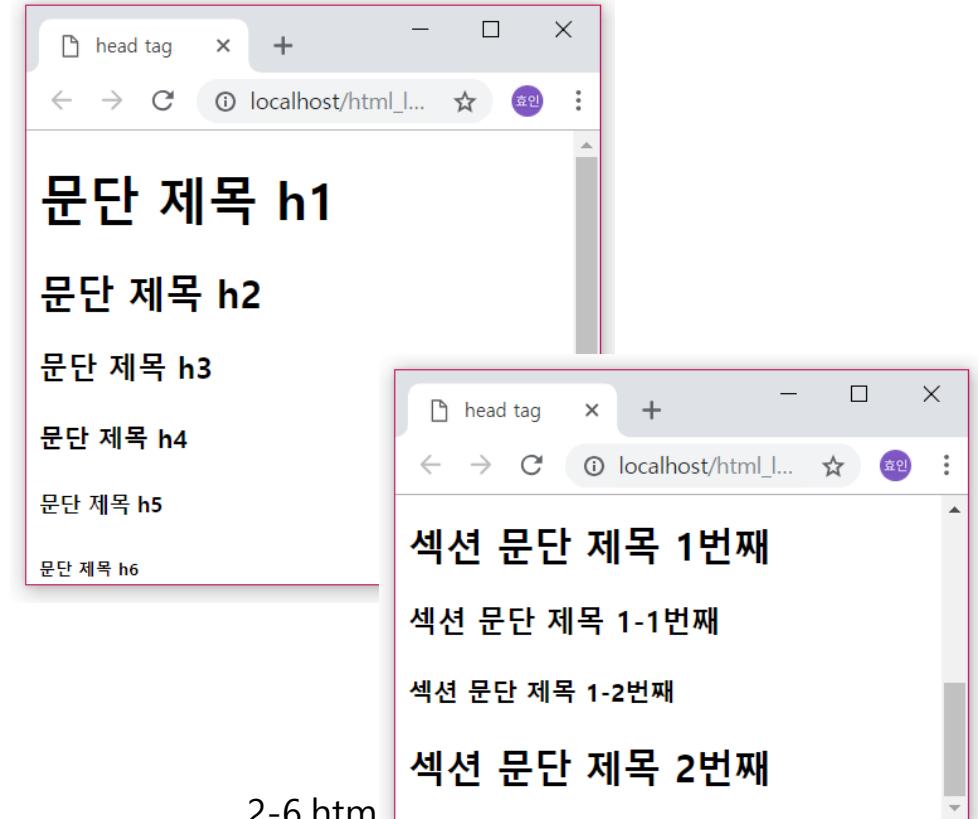
## 2. HTML 기본

### 2-6. Body 요소 (2/2) - heading

- ✓ 문단의 제목을 지정할 때 사용. <h1>부터 <h6>까지 구분. 숫자가 커질수록 글자는 작아짐.
- ✓ <section> tag를 이용하면 같은 tag를 서로 다르게 표현.
- ✓ 문서 구조를 <section> tag를 이용하여 구분하면 각 문단의 제목을 하나의 tag로 작성 가능.

```
<body>
<h1>문단 제목 h1</h1>
<h2>문단 제목 h2</h2>
<h3>문단 제목 h3</h3>
<h4>문단 제목 h4</h4>
<h5>문단 제목 h5</h5>
<h6>문단 제목 h6</h6>

<section>
  <h1>섹션 문단 제목 1번째</h1>
  <section>
    <h1>섹션 문단 제목 1-1번째</h1>
    <section>
      <h1>섹션 문단 제목 1-2번째</h1>
    </section>
  </section>
  <h1>섹션 문단 제목 2번째</h1>
</section>
</body>
```



The screenshot shows two separate browser windows side-by-side, both displaying the same HTML content. The left window displays the main document structure, and the right window displays the content within the first section tag.

**Left Window (Main Document Structure):**

- 문단 제목 h1
- 문단 제목 h2
- 문단 제목 h3
- 문단 제목 h4
- 문단 제목 h5
- 문단 제목 h6

**Right Window (Content within first section):**

- 섹션 문단 제목 1번째
- 섹션 문단 제목 1-1번째
- 섹션 문단 제목 1-2번째
- 섹션 문단 제목 2번째



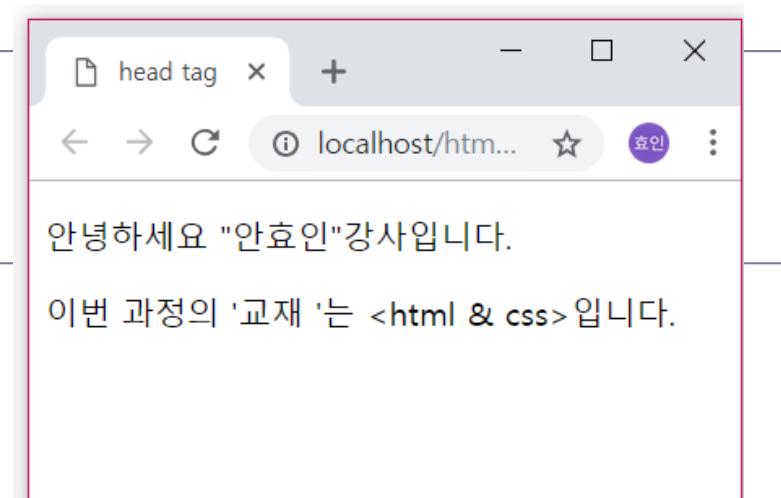
## 2. HTML 기본

### 2-7. 특수문자.

엔티티 이름	설명	화면출력
&nbsp;	Non-breaking space(공백)	
&lt;	Less than	<
&gt;	Greater than	>
&amp;	Ampersand	&
&quot;	Quotation mark	"
&copy	Copyright	©
&reg;	registered trademark	®

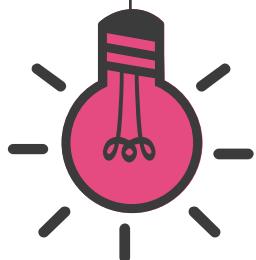
```
<body>
<p>안녕하세요 &quot;안효인&quot;강사입니다.</p>
<p>이번 과정의 &apos;교재 &apos;는 &lt;html &amp; css&gt;입니다.</p>
</body>
```

2-7.html



03

## HTML 마크업 요소





### 3. HTML 마크업 요소

#### 3-1. 포맷팅 요소 (1/2)

- ✓ 포맷팅 요소에는 화면에는 동일하게 출력되지만 각 요소가 가진 의미가 다른 것이 있다.
- ✓ 예를 들면, **<b>**와 **<strong>**은 모두 텍스트를 굵게 표현하지만, **<strong>**요소는 텍스트를 강조.

tag명	설명	tag명	설명
<abbr>	생략된 약어 표시. Title 속성을 함께 사용.	<mark>	특정 문자열을 강조. 화면에는 하이라이팅 됨.
<address>	연락처 정보 표시.	<hr>	구분선.
<blockquote>	긴 인용문구 표시, 좌우로 들여쓰기가 됨.	<b>, <strong>	굵은 글씨로 표시, 특정 문자열을 강조(<strong>).
<q>	짧은 이용문구 표시, 좌우로 따옴표가 붙음.	<i>, <em>	이탤릭(기울게) 표시, 특정 문자열을 강조(<em>)
<cite>	웹 문서나 포스트에서 참고 내용 표시.	<big>, <small>	큰 글자, 작은 글자로 표시.
<pre>	공백, 줄바꿈등 입력된 그대로 화면에 표시.	<sup>, <sub>	위 첨자, 아래 첨자로 표시.
<code>	컴퓨터 인식을 위한 소스 코드.	<s>, <u>	취소선, 밑줄



# 3. HTML 마크업 요소

## 3-1. 포맷팅 요소 (2/2)

3-0.html

pre tag를 이용하여 javascript 함수 그대로 출력

```
function test() {
    alert("함수가 그대로 출력.");
}
```

```
<h4>pre tag를 이용하여 javascript 함수 그대로 출력</h4>
<pre>
<code>
function test() {
    alert("함수가 그대로 출력.");
}
</code>
</pre>
<hr>
<blockquote>
<abbr title="Internet of Things"><mark>IoT</mark></abbr>란 인터넷을 기반으로 모든 사물을 연결하여 정보를 상호 소통하는 지능형 기술 및 서비스.

```

(출처 : <q cite="https://terms.naver.com/aliikeMeaning.nhn?query=E00273180">Naver 지식백과</q>)

```
</blockquote>
<hr>
```

<strong>의미있는 글자</strong>를 굵게 할 때는 &lt;strong&gt;, <b>단순 글자</b>를 굵게 할 때는 &lt;b&gt;를 씁니다.<br>

<em>의미있는 글자</em>를 비스듬히 할 때는 &lt;em&gt;, <i>단순 글자</i>를 비스듬히 할 때는 &lt;i&gt;를 씁니다.

IoT란 인터넷을 기반으로 모든 사물을 연결하여 정보를 상호 소통하는 지능형 기술 및 서비스.  
(출처 : "Naver 지식백과")

의미있는 글자를 굵게 할 때는 <strong>, 단순 글자를 굵게 할 때는 <b>를 씁니다.  
의미있는 글자를 비스듬히 할 때는 <em>, 단순 글자를 비스듬히 할 때는 <i>를 씁니다.



### 3. HTML 마크업 요소

#### 3-2. 목록형 요소 (1/3) - 개요

- ✓ 목록 tag는 하나 이상의 하위 tag를 포함.
- ✓ 목록 tag는 각 항목을 들여쓰기로 표현.
- ✓ 번호 또는 심볼을 이용해서 목록을 표현.

이 태그 속성

tag명	설명
<ul>	번호 없는 목록을 표시. 항목 앞에 심볼을 표시.
<ol>	번호 있는 목록을 표시. 숫자, 알파벳, 로마숫자 등으로 표시.
<li>	목록 항목 <ul>이나 <ol> tag 하위에서 사용.
<dl>	용어 정의와 설명에 대한 내용을 목록화해서 표시.
<dt>	용어 목록의 정의 부분을 나타냄.
<dd>	용어 목록의 설명 부분을 나타냄.

속성	속성값	설명
	1	숫자(기본값)
	a	영문 소문자
type	A	영문 대문자
	i	로마숫자 소문자
	I	로마숫자 대문자
start	숫자	시작 번호
reversed		역순으로 표시



### 3. HTML 마크업 요소

#### 3-2. 목록형 요소 (2/3) - 개요

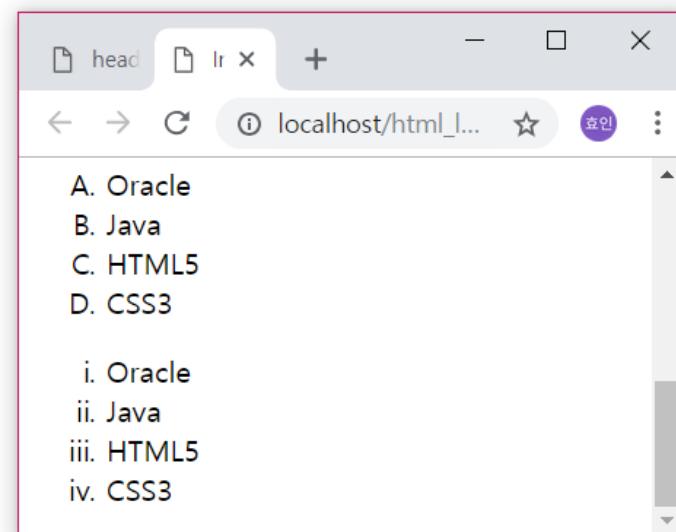
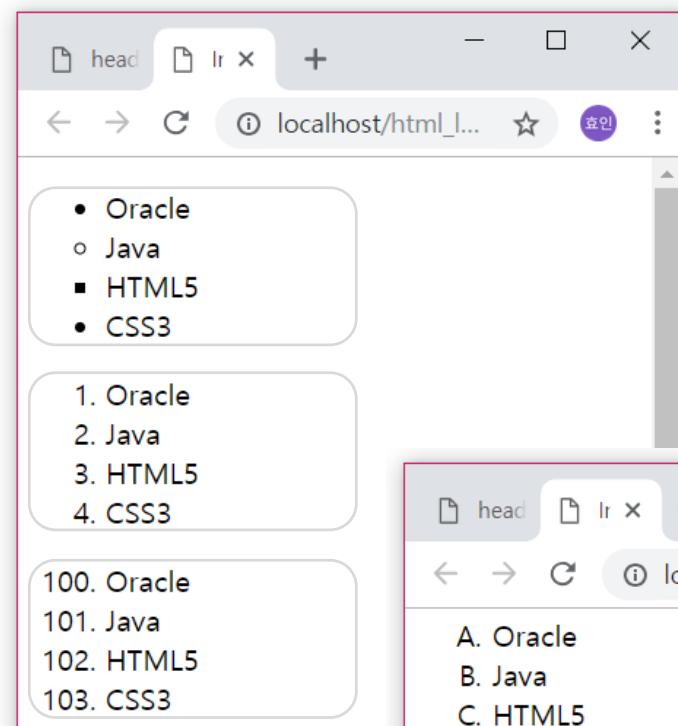
```
<ul>
    <li>Oracle</li>
    <li style="list-style-type: circle;">Java</li>
    <li style="list-style-type: square;">HTML5</li>
    <li style="list-style-type: disc;">CSS3</li>
</ul>

<ol>
    <li>Oracle</li>
    <li>Java</li>
    <li>HTML5</li>
    <li>CSS3</li>
</ol>

<ol start="100">
    <li>Oracle</li>
    <li>Java</li>
    <li>HTML5</li>
    <li>CSS3</li>
</ol>

<ol style="list-style-type: upper-alpha;">
    <li>Oracle</li>
    <li>Java</li>
    <li>HTML5</li>
    <li>CSS3</li>
</ol>

<ol style="list-style-type: lower-roman;">
    <li>Oracle</li>
    <li>Java</li>
    <li>HTML5</li>
    <li>CSS3</li>
</ol>
```





# 3. HTML 마크업 요소

## 3-2. 목록형 요소 (3/3) - 개요

```
<dl>
    <dt>HTML</dt>
        <dd>Hypertext Markup Language</dd>
    <dt>XML</dt>
        <dd>eXtensible Markup Lanuage</dd>
</dl>

<ol>
    <li>HTML & XML
        <dl>
            <dt>HTML</dt>
                <dd>Hypertext Markup Language</dd>
            <dt>XML</dt>
                <dd>eXtensible Markup Lanuage</dd>
        </dl>
    </li>
    <li>GET & POST
        <dl>
            <dt>GET</dt>
                <dd>QueryString을 통해서 data 전송. 적은양의 data전송에 적합하다.</dd>
            <dt>POST</dt>
                <dd>http 헤더에 숨겨져 서버로 전송. data 전송양의 제약이 없다.</dd>
        </dl>
    </li>
</ol>
```

The screenshot shows a browser window with the URL `localhost/html_lab/chap3/3-2.h...`. The page displays the following content:

HTML  
Hypertext Markup Language

XML  
eXtensible Markup Lanuage

1. HTML & XML

HTML  
Hypertext Markup Language

XML  
eXtensible Markup Lanuage

2. GET & POST

GET  
QueryString을 통해서 data 전송. 적은양의 data전송에 적합하다.

POST  
http 헤더에 숨겨져 서버로 전송. data 전송양의 제약이 없다.



### 3. HTML 마크업 요소

#### 3-2. 목록형 요소 - Quiz

- ✓ 순서가 없는 index(숫자, 알파벳)을 만든다.
- ✓ HTML5 소개부터 CSS이해까지의 항목을 가진 순서 목록을 만든다.
- ✓ 첫 번째 목록은 아라비아숫자로 나열하는 순서 목록.
- ✓ 두 번째 목록은 알파벳 소문자로 나열하는 순서 목록.

#### html study

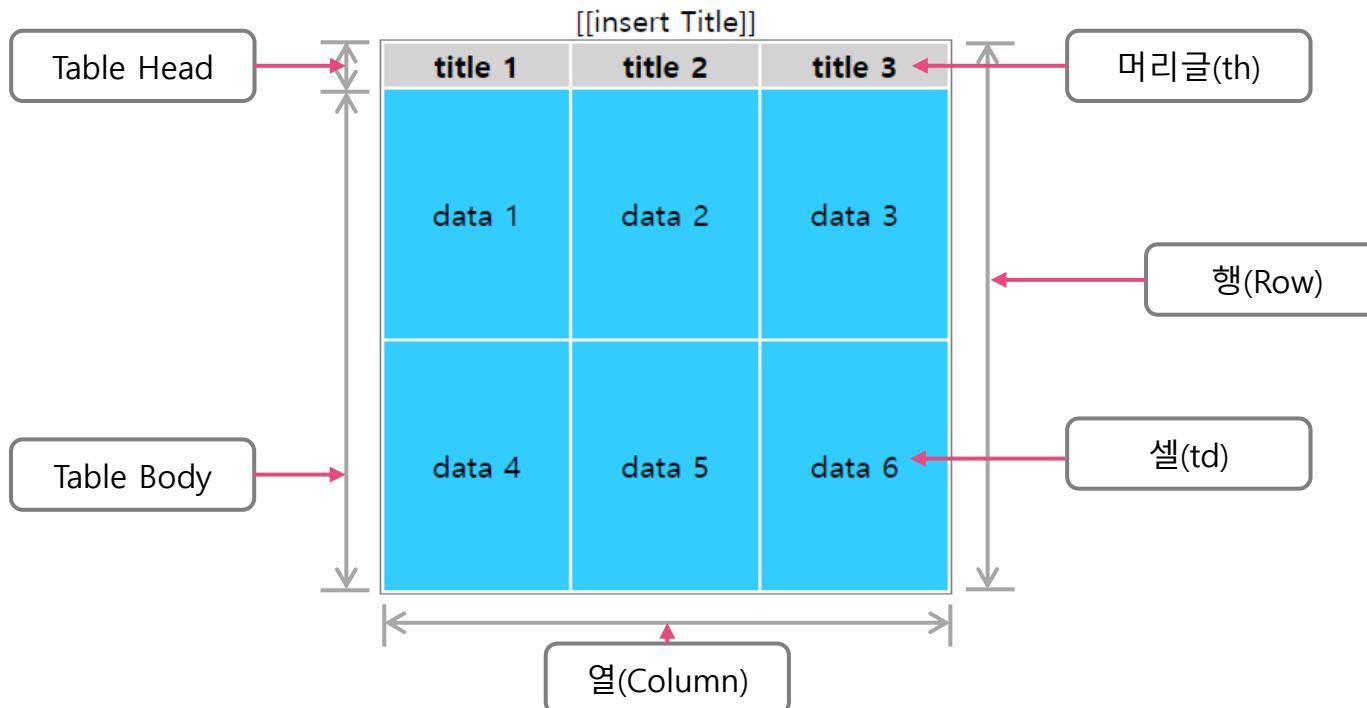
- index - 숫자
  1. HTML5와 CSS3의 소개
  2. HTML 기본
  3. HTML 마크업 요소
  4. CSS 기본
  5. CSS 이해
- index - 알파벳(소문자)
  - a. HTML5와 CSS3의 소개
  - b. HTML 기본
  - c. HTML 마크업 요소
  - d. CSS 기본
  - e. CSS 이해



### 3. HTML 마크업 요소

#### 3-3. table (1/7) – HTML table 모델

- ✓ HTML table 모델은 데이터를 행(Row)과 열(Column)의 셀(Cell)에 표시.
- ✓ 행 그룹요소인 <thead>, <tbody>, <tfoot> 요소를 사용하여 행들을 그룹화.
- ✓ <colgroup>과 <col> 요소는 열 그룹을 위한 추가적인 구조정보를 제공.
- ✓ table의 셀(Cell)은 머리글(<th>)이나 데이터(<td>)를 가질 수 있음.



```
<table>
  <caption>[[insert Title]]</caption>
  <thead>
    <tr>
      <th>title 1</th>
      <th>title 2</th>
      <th>title 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>data 1</td>
      <td>data 2</td>
      <td>data 3</td>
    </tr>
    <tr>
      <td>data 4</td>
      <td>data 5</td>
      <td>data 6</td>
    </tr>
  </tbody>
</table>
```



### 3. HTML 마크업 요소

#### 3-3. table (2/7) – table 스타일

- ✓ <table>, <tr>, <td> tag에는 table에 스타일을 적용하는 다양한 속성들이 있다.
- ✓ frame 속성은 table의 특정 선을 보여줄지를 결정하며, rules 속성은 셀과 셀사이의 줄이 나타날 것인가를 지정.
- ✓ table 정렬을 위한 align속성, background, bgcolor, border 속성 등이 있음.
- ✓ HTML5부터는 이러한 속성들을 더 이상 지원하지 않는다. 스타일은 CSS를 사용하여 적용.

```
<table cellpadding="3" cellspacing="2"
    bordercolorlight="red" bordercolordark="steelblue"
    bgcolor="cyan" border="1"
    width="300" height="150">
<tr bgcolor="orange">
    <td>1</td>
    <td>2</td>
    <td>3</td>
</tr>
<tr bgcolor="Lightpink">
    <td>가</td>
    <td>나</td>
    <td>다</td>
</tr>
</table>
```

HTML4.01까지는 table 스타일을 위한 속성이 제공되었으나 HTML5부터는 CSS를 사용하여 스타일을 적용.

1	2	3
가	나	다



### 3. HTML 마크업 요소

#### 3-3. table (3/7) – table 제목

- ✓ **<caption>** tag는 table 제목을 정의하기 위해 사용하며 **<table>** start tag 바로 뒤에 위치.
- ✓ **<caption>** tag는 table 당 하나만 사용.
- ✓ table 제목은 기본적으로 가운데 정렬. 정렬 방식 변경은 CSS를 사용.

```
<table border="1">
  <caption>강좌 리스트</caption>
  <thead>
    <tr>
      <th>No</th><th>강좌명</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td><td>HTML5 & CSS3</td>
    </tr>
    <tr>
      <td>2</td><td>Spring Framework</td>
    </tr>
  </tbody>
</table>
```

3-5.html

```
<style type="text/css">
caption {
  text-align: left;
}
</style>
```

The screenshot shows two separate browser windows side-by-side. Both windows have a title bar with 'Insert title' and a URL 'localhost/ht...'. The main content area displays a table with the following data:

No	강좌명
1	HTML5 & CSS3
2	Spring Framework

In the top window, the caption '강좌 리스트' is centered above the table. In the bottom window, the caption '강좌 리스트' is aligned to the left of the table. Both captions are highlighted with a red box.



### 3. HTML 마크업 요소

#### 3-3. table (4/7) – 행(Row) 그룹 요소

- ✓ **table 행 그룹 요소는 table의 행들을 머리글, 바닥글, 하나 이상의 본체 항목으로 그룹핑.**
- ✓ **행 그룹 요소에는 머리글(<thead>), 본체항목(<tbody>), 바닥글(<tfoot>) 요소가 있다.**
- ✓ **각 행 그룹은 최소 하나 이상의 <tr>을 가져야 함.**

```
<table border="1" width="300">
  <caption>강좌 리스트</caption>
  <thead>
    <tr>
      <th>No</th><th>강좌명</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td><td>HTML5 & CSS3</td>
    </tr>
    <tr>
      <td>2</td><td>Spring Framework</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>강좌합계</td><td>2개</td>
    </tr>
  </tfoot>
</table>
```

No	강좌명
1	HTML5 & CSS3
2	Spring Framework
강좌합계	2개

<tbody>는 여러 번 사용이 가능.  
여러행을 하나의 <tbody>로 묶어  
특정 스타일을 적용할 수 있다.



### 3. HTML 마크업 요소

#### 3-3. table (5/7) – 열(Column) 그룹 요소

- ✓ **table** 열 그룹 요소는 **table** 내에서 구조적인 분리를 가능케함.
- ✓ **<colgroup>** 요소는 명시적인 열 그룹을 만들며, **<col>** 요소는 열(column)을 묶어 그룹핑 함.
- ✓ **<colgroup>** 요소의 **span** 속성은 열 그룹에서 열의 개수를 정의.
- ✓ **<col>** 요소의 **span** 속성은 **<col>**에 의해 묶여진 열의 개수를 말함.

```
<table border="1" width="500">
  <colgroup>
    <col span="2" style="background-color: skyblue">
    <col width="200">
  </colgroup>
  <tr>
    <th>No</th><th>강좌명</th><th>가격</th>
  </tr>
  <tr>
    <td>1</td><td>HTML5 & CSS3</td><td>25000원</td>
  </tr>
  <tr>
    <td>2</td><td>Spring Framework</td><td>47000원</td>
  </tr>
</table>
```

No	강좌명	가격
1	HTML5 & CSS3	25000원
2	Spring Framework	47000원

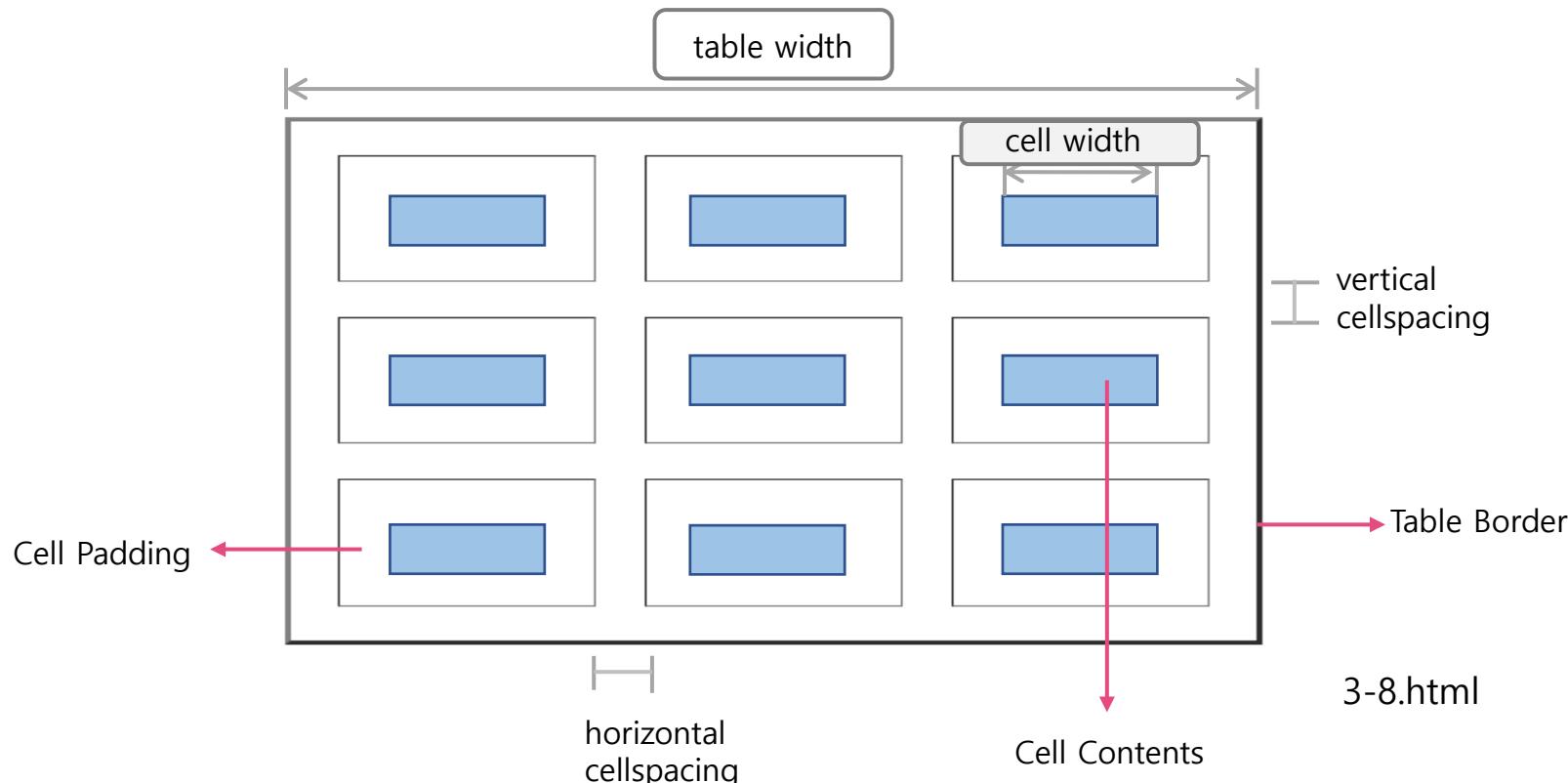
HTML 4.01까지는 열 그룹에 대한 스타일 속성이 제공되었다. align, valign, width 등과 같은 스타일 속성은 HTML5 부터 지원되지 않는다.  
스타일은 CSS 사용하여 적용.



### 3. HTML 마크업 요소

#### 3-3. table (6/7) – 테두리(border)

- ✓ **cellspacing**은 table Cell과 Cell 사이의 공간을 의미.
- ✓ **cellpadding**은 Cell 외곽과 Cell 컨텐츠 사이 공간을 의미.
- ✓ HTML5 부터는 테두리 스타일 관련 속성을 지원하지 않고, CSS를 사용.





### 3. HTML 마크업 요소

#### 3-3. table (7/7) – 셀 병합

- ✓ HTML table의 <td> 요소에는 셀(Cell)을 병합하기 위한 두개의 속성이 있음.
- ✓ colspan 속성은 두 개 이상의 열(Column)을 하나로 합치기 위해 사용.
- ✓ rowspan 속성은 두 개 이상의 행(Row)을 하나로 합치기 위해 사용.

1	2
3	4

```
<table>
<tr>
  <td>1</td>
  <td>2</td>
</tr>
<tr>
  <td>3</td>
  <td>4</td>
</tr>
</table>
```

1	2
3	4

```
<table>
<tr>
  <td rowspan="2"
    class="view">1</td>
  <td>2</td>
</tr>
<tr>
  <td>4</td>
</tr>
</table>
```

1	
3	4

```
<table>
<tr>
  <td colspan="2"
    class="view">1</td>
</tr>
<tr>
  <td>3</td>
  <td>4</td>
</tr>
</table>
```

1	2
3	4

```
<table>
<tr>
  <td>1</td>
  <td rowspan="2"
    class="view">2</td>
</tr>
<tr>
  <td>3</td>
</tr>
</table>
```



### 3. HTML 마크업 요소

#### 3-3. table - Quiz

- ✓ <caption> tag를 사용.
- ✓ Table의 width : 600, height : 250
- ✓ CSS는 사용하지 않는다.

이력서

아래 정보는 거짓이 없습니다.

사진	성명	홍 길 동	주민등록번호
			000000-0000000
생년월일 2000년 01월 01일			
주소	서울시 강남구 역삼동 77-1		
연락처	집	02-1234-5678	이메일
	핸드폰	010-9876-5432	



### 3. HTML 마크업 요소

#### 3-4. 이미지 요소 (1/2) – img

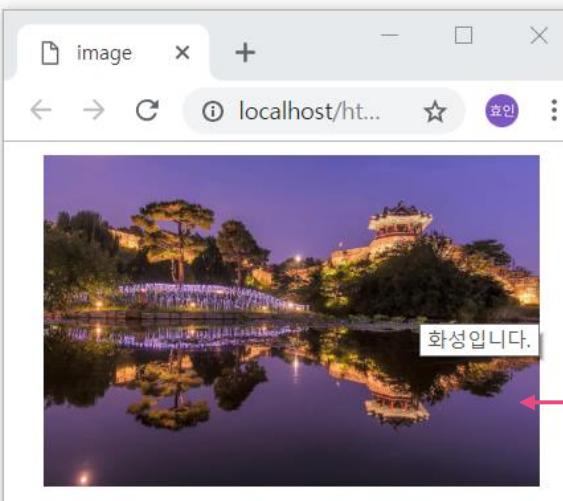
- ✓ **<img> tag**를 사용하여 이미지를 삽입하기 위해 사용.
- ✓ **src 속성**은 이미지 경로를 지정하기 위해 사용 (상대경로, URL 모두 가능).
- ✓ **height, width 속성**은 이미지 사이즈를 지정하기 위해 사용.
- ✓ **alt 속성**은 이미지를 표시할 수 없을 때 화면에 대신하여 보여질 텍스트를 지정.

```

```

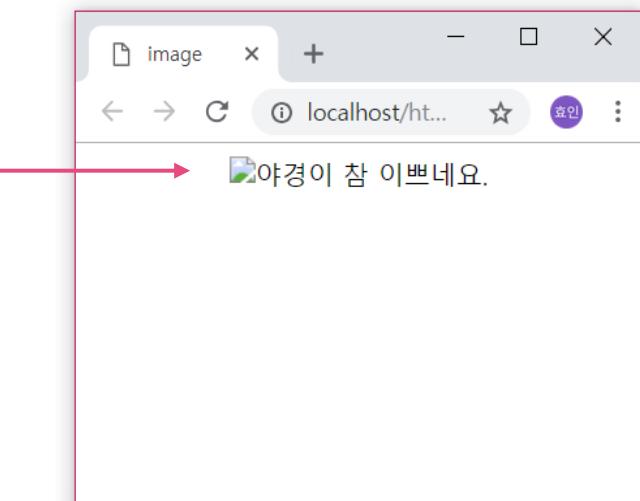
width와 height 속성값은 픽셀단위로 지정.

3-11.html



정상적으로 실행된 화면

이미지 파일이 잘못된 경우 alt의  
텍스트가 출력



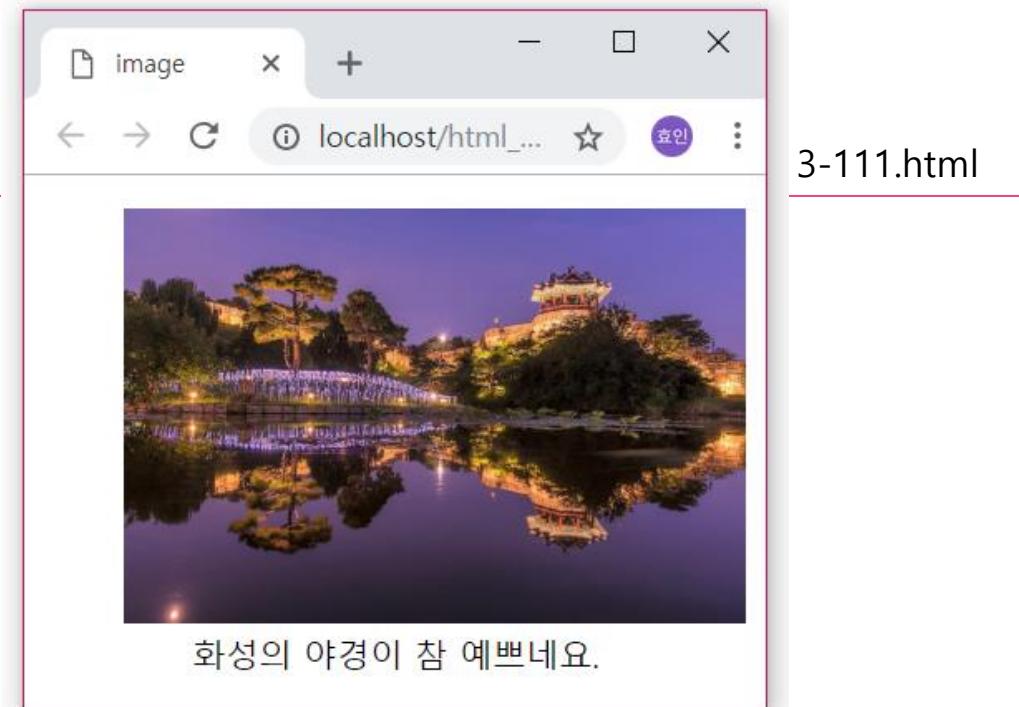


### 3. HTML 마크업 요소

#### 3-4. 이미지 요소 (2/2) – img

- ✓ <figure>는 설명 글을 붙여야 할 대상을 지정.
- ✓ <figcaption> : 설명글이 필요한 대상은 <figure> 태그로 묶고 설명 글은 <figcaption>태그로 묶는다.
- ✓ 설명글을 붙일 수 있는 대상은 이미지나 오디오, 비디오 같은 미디어파일이나 텍스트 단락이나 표.

```
<figure>
  
  <figcaption>화성의 야경이 참 예쁘네요.</figcaption>
</figure>
```





# 3. HTML 마크업 요소

## 3-5. 링크 요소 (1/5) – Anchor

- ✓ <a> tag를 사용하여, 하나의 문서에서 다른 문서로 연결하기 위해 (하이퍼링크) 사용.
- ✓ tag를 서로 중첩해서 사용할 수 없다.
- ✓ href 속성은 하이퍼링크를 클릭했을 때 이동할 문서의 URL이나 문서의 책갈피를 지정.
- ✓ target 속성은 하이퍼링크를 클릭했을 때 현재 윈도우 또는 새로운 윈도우에서 이동할지를 지정.

```
<h2>java 과정</h2>
<a href="java.html" target="_self">java fundamental</a><br>
<a href="web.html">servlet/jsp</a><br>
<a href="mybatis.html">mybatis</a><br>
<a href="spring.html" target="_self">spring/springboot</a><br>
<a href="android.html">android</a><br>
<h2>관련 사이트</h2>
<a href="http://www.naver.com" target="_self">네이버</a><br>
<a href="http://www.daum.net">다음</a><br>
<a href="http://www.nate.com">네이트</a><br>
<a href="http://www.google.com" target="_blank">구글</a><br>
```

Target 속성값 중 \_self는 현재 윈도우에서 다른 문서로 이동하고, \_blank는 새로운 윈도우 또는 새로운 탭에서 다른 문서로 이동 (브라우저 설정에 따른).

3-10.html

### java 과정

[java fundamental](#)  
[servlet/jsp](#) 현재창  
[mybatis](#)  
[spring/springboot](#)  
[android](#)

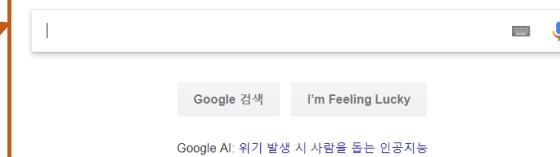
### 관련 사이트

[네이버](#)  
[다음](#)  
[네이트](#)  
[구글](#)

### Java Fundamental

1. java의 이해.
2. java 기본 문법.
3. Object Oriented programming(OOP)
4. java collection framework
5. input/output (IO)
6. Network
7. JDBC

Google





### 3. HTML 마크업 요소

#### 3-5. 링크 요소 (2/5) – target

##### 속성 값

##### 설명

**\_blank**

링크 내용이 새 창이나 새 탭에서 열린다.

**\_self**

target 속성의 기본 값으로 링크가 있는 화면에서 열린다.

**\_parent**

프레임을 사용했을 때 링크 내용을 부모 프레임에 표시.

**\_top**

프레임을 사용했을 때 프레임에서 벗어나 링크 내용을 전체 화면으로 표시.



### 3. HTML 마크업 요소

#### 3-5. 링크 요소 (3/5) – #anchor

- ✓ 같은 페이지 안에서 특정 요소를 클릭 시 그 위치로 한 번에 이동시킨다.
- ✓ `<tag id="anchor name"> text or image </tag>`
- ✓ `<a href="#anchor name">text or image</a>`

```
<ul id="menu"> ←  
  <li><a href="#content1">내용1</a></li>  
  <li><a href="#content2">내용2</a></li>  
  <li><a href="#content3">내용3</a></li>  
</ul>  
  
<h3 id="content1">내용 1</h3>  
<p>내용이 상당히 길다.....</p>  
<p><a href="#menu">메뉴로이동</a></p> →  
  
<h3 id="content2">내용 2</h3>  
<p>내용이 상당히 길다.....</p>  
<p><a href="#menu">메뉴로이동</a></p> →  
  
<h3 id="content3">내용 3</h3>  
<p>내용이 상당히 길다.....</p>  
<p><a href="#menu">메뉴로이동</a></p> →
```



### 3. HTML 마크업 요소

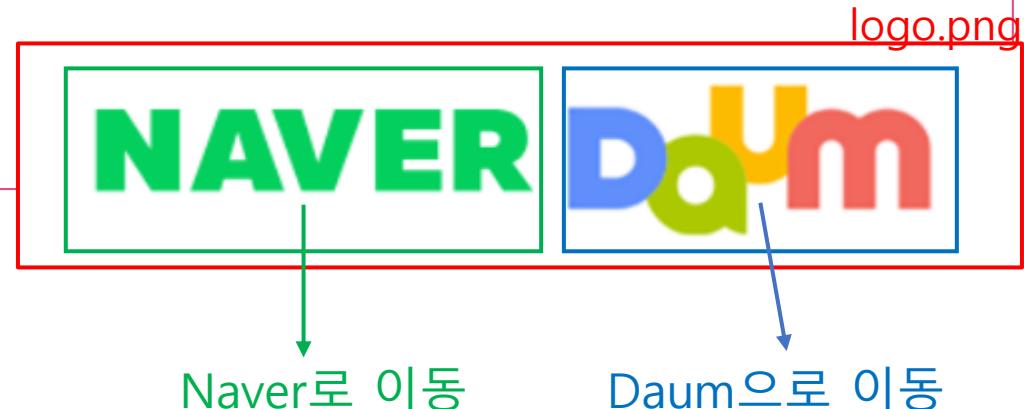
#### 3-5. 링크 요소 (4/5) – map

- ✓ 하나의 이미지에 여러 개의 link (Click 위치에 따라 서로 다른 link).
- ✓ <map name="map name"> <area> <area> ... </map>
- ✓ 이미지에 영역을 표시 할 때 <area> 사용.
- ✓ area의 속성은 href(링크 경로), target(링크 표시 대상), shape(링크로 사용할 영역의 형태) 등.
- ✓ shape의 값으로는 default, rect, circle, poly가 있다.

```

<map name="Logo">
    <area shape="rect" coords="5,5,185,80"
          href="http://www.naver.com" target="_blank">
    <area shape="rect" coords="190,5,345,80"
          href="http://www.daum.net" target="_blank">
</map>
```

3-112.html





### 3. HTML 마크업 요소

#### 3-5. 링크 요소 (5/5) – link

- ✓ link tag를 사용하여 문서와 외부 자원을 연결하기 위해 사용.
- ✓ <head> 위치에 정의하며 여러 자원을 연결할 수 있다.
- ✓ rel 속성은 현재 문서와 연결된 문서 사이의 연관관계를 지정하기 위해 사용.
- ✓ href 속성은 연결된 문서의 위치를 지정하기 위해 사용.

```
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="css/main.css">
  <link rel="stylesheet" href="css/menu.css">
</head>
```

주로 스타일시트(CSS)를 연결하기 위한 용도로 사용.



### 3. HTML 마크업 요소

#### 3-6. 프레임 요소 – iframe

- ✓ 화면의 일부분에 다른 문서를 포함.
- ✓ src 속성은 포함시킬 외부 문서의 경로를 지정(상대경로, URL 모두 가능).
- ✓ height, width 속성은 프레임 사이즈를 지정.
- ✓ name 속성은 프레임의 이름을 지정.

```
<h3>iframe 요소</h3>
<a href="java.html" target="javacontent">
    java fundamental</a>
<a href="web.html" target="javacontent">
    servlet/jsp</a>
<a href="mybatis.html" target="javacontent">
    mybatis</a>
<a href="spring.html" target="javacontent">
    spring/springboot</a>
<a href="android.html" target="javacontent">
    android</a>
<br><br>
<iframe src="java.html" name="javacontent"
        width="500" height="300"></iframe>
```

3-12.html

iframe

localhost/html\_lab/cha...

### iframe 요소

java fundamental servlet/jsp mybatis spring/springboot android

#### Java Fundamental

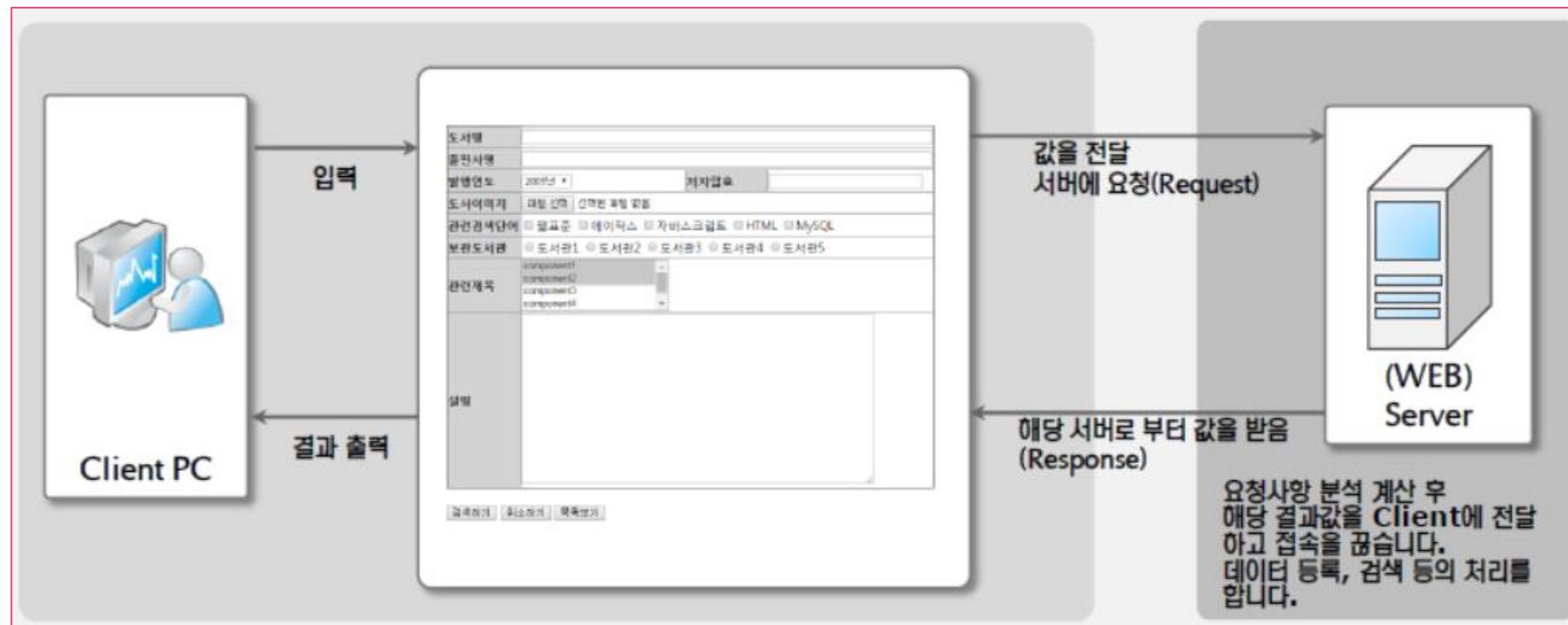
1. java의 이해.
2. java 기본 문법.
3. Object Oriented programming(OOP)
4. java collection framework
5. input/output (IO)
6. Network
7. JDBC



### 3. HTML 마크업 요소

#### 3-7. form control 요소 (1/5) – 개요(1/2)

- ✓ 사용자로부터 데이터를 입력 받아 서버에서 처리하기 위한 용도로 사용.
- ✓ 사용자의 요청에 따라 서버는 HTML form을 전달(회원가입양식, 검색 양식 등).
- ✓ 사용자는 HTML form에 적절한 데이터를 입력한 후 서버로 전송. 이를 submit이라 함.
- ✓ 서버는 사용자의 요청을 분석한 후 데이터를 등록하거나, 원하는 데이터를 조회하여 결과를 다시 반환.





### 3. HTML 마크업 요소

#### 3-7. form control 요소 (2/5) – 개요(2/2)

- ✓ 사용자가 입력하기 위한 **control** 요소들은 모두 **<form>** tag 하위에 위치해야 서버로 전송됨.
- ✓ 각 **control** 요소마다 텍스트 입력, 버튼 클릭 등 다양한 형식으로 입력을 받는다.

tag명	설 명
<b>&lt;form&gt;</b>	사용자에게 입력 받을 항목을 정의. form tag 내부에 여러 개의 control 요소를 포함.
<b>&lt;input&gt;</b>	텍스트 box, 체크 box, 라디오 버튼 등 사용자가 데이터를 입력할 수 있도록 함.
<b>&lt;textarea&gt;</b>	여러 줄의 문자를 입력할 수 있도록 함.
<b>&lt;button&gt;</b>	버튼을 표시.
<b>&lt;select&gt;</b>	select box(드롭다운, 콤보box)를 표시.
<b>&lt;optgroup&gt;</b>	select box의 각 항목들을 그룹화 함.
<b>&lt;option&gt;</b>	select box의 각 항목들을 정의 함.
<b>&lt;label&gt;</b>	마우스를 이용하여 <b>&lt;input&gt;</b> 항목을 선택 시 편리함을 제공, for 속성을 이용하여 다른 control 요소와 텍스트를 연결시켜서 더 편리하게 선택할 수 있도록 함.
<b>&lt;fieldset&gt;</b>	입력 항목들을 그룹화 함
<b>&lt;legend&gt;</b>	<b>&lt;fieldset&gt;</b> 의 제목을 지정 함.



### 3. HTML 마크업 요소

#### 3-7. form control 요소 (3/5) – form

- ✓ 사용자가 입력한 자료들을 어떤 방식으로 서버로 전달할 것인지 결정.
- ✓ 서버에서 어떤 프로그램을 이용해 처리할 것인지 결정.
- ✓ <form [속성="속성값"]> form control </form>

속성	설명
	사용자가 입력한 내용을 서버 쪽 프로그램으로 어떻게 넘겨줄지 지정.
method	<b>GET</b> : 주소 표시줄에 사용자가 입력한 내용이 표시. 256 ~ 4096bytes의 데이터만 서버로 전송. <b>POST</b> : 사용자의 입력을 표준 입력으로 넘겨주기 때문에 입력 내용의 길이에 제한이 없다. 사용자가 입력한 내용이 표시되지 않는다.
name	form의 이름을 지정. 한 문서 안에 여러 개의 <form> 태그가 있을 경우, 구분자로 사용.
action	<form> 태그 안의 내용들을 처리해 줄 서버상의 프로그램 지정.(URL)
target	<action> 태그에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에 열도록 지정.
autocomplete	자동완성 기능. 기본값 on



### 3. HTML 마크업 요소

#### 3-7. form control 요소 (4/5) – label

- ✓ **form control**에 레이블(텍스트)을 연결.
- ✓ 사용법
  - <label [속성="속성값"]> 레이블 <input ...> </label>
  - <label for="id이름"> <input id="id이름" [속성="속성값"]> </label>

```
<h2>form control - label</h2>
<form method="post" action="login.jsp">
<ul type="none">
    <li>
        <label for="userid">아이디 : </label>
        <input type="text" id="userid" name="userid">
    </li>
    <li>
        <label for="pass">비밀번호 : </label>
        <input type="text" id="pass" name="pass">
    </li>
    <li><input type="submit" value="로그인"></li>
</ul>
</form>
```

The screenshot shows a web browser window with the title 'form control - label'. The page content is a login form. It has two text input fields: one for '아이디' (User ID) and one for '비밀번호' (Password). Below the password field is a submit button labeled '로그인' (Login).



### 3. HTML 마크업 요소

#### 3-7. form control 요소 (5/5) – fieldset, legend

- ✓ **form** 요소를 그룹으로 묶음.

```
<fieldset>
    <legend>필수 입력</legend>
    <ul type="none">
        <li>
            <label for="userid">아이디 : </label><input type="text" id="us
        </li>
        <li>
            <label for="pass">비밀번호 : </label><input type="text" id="pa:
        </li>
    </ul>
</fieldset>
<fieldset>
    <legend>선택 입력</legend>
    <ul type="none">
        <li>
            <label for="phone">전화번호 : </label><input type="text" id="phone" name="phone">
        </li>
        <li>
            <label for="address">주소 : </label><input type="text" id="address" name="address">
        </li>
    </ul>
</fieldset>
```

필수 입력

아이디 :

비밀번호 :

선택 입력

전화번호 :

주소 :



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (1/17) – 개요

- ✓ <input> tag는 type 속성에 따라 여러 가지 형태로 화면에 표시.
- ✓ <input type="유형" [속성="속성값"]>
- ✓ id 속성은 여러 번 사용된 폼 요소를 구분하기 위해 사용.
- ✓ id 속성 값은 최소한 한 개 이상의 문자여야 하며 공백은 허용 X.
- ✓ 같은 html document에서 id는 하나의 값만 가능하고, name은 중복이 허용된다.

```
<h2>form control - input</h2>
아이디 : <input type="text" id="id" name="userid"><br>
비밀번호 : <input type="password" id="pwd" name="pass"><br>
전화번호 : <input type="text" id="tel" name="phone" size="15"
            maxlength="11"><br>
<input type="submit" value="전송">
<input type="reset" value="초기화">
```

3-131.html

form control - input

아이디 :

비밀번호 :

전화번호 :

전송    초기화



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (2/17) – type 속성

##### type

##### 설명

text 한 줄의 텍스트를 입력할 수 있는 텍스트 상자.

password 비밀번호를 입력할 수 있는 필드.(text가 '\*'로 표시)

search 검색 상자.

tel 전화번호를 입력할 수 있는 필드.

url URL 주소를 입력할 수 있는 필드.

email 메일 주소를 입력할 수 있는 필드.

datetime 국제 표준시(UTC)로 설정된 날짜와 시간(년, 월, 일, 시, 분, 초, 분할 초).

datetime-local 사용자 지역을 기준으로 날짜와 시간(년, 월, 일, 시, 분, 초, 분할 초).

date 사용자 지역을 기준으로 날짜(년, 월, 일).

month 사용자 지역을 기준으로 날짜(년, 월).

week 사용자 지역을 기준으로 날짜(년, 주).

time 사용자 지역을 기준으로 시간(시, 분, 초, 분할 초).



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (3/17) – type 속성

##### type

##### 설명

number 숫자를 조절할 수 있는 화살표.

range 숫자를 조절할 수 있는 슬라이드 막대.

color 색상 표.

checkbox 주어진 항목에서 2개 이상 선택 가능한 체크박스(다중선택).

radio 주어진 항목에서 1개만 선택할 수 있는 라디오 버튼(단일선택).

file 파일을 첨부할 수 있는 버튼.

submit 서버 전송 버튼.

image submit + image.

reset 리셋 버튼.

button 기능이 없는 버튼.

hidden 사용자에게는 보이지 않지만 서버로 넘겨지는 값을 설정.



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (4/17) – textfield, password

##### ✓ **textfield**

- 한 줄의 일반 텍스트를 입력 받는 필드.
- `<input type="text" [속성="속성값"]>`

##### ✓ **password field**

- 비밀번호 입력란.
- 내용이 표시되지 않고 '\*' 또는 '●'으로 표시.
- `<input type="password" [속성="속성값"]>`

textfield 속성.

속성	설명
name	textfield를 구별할 수 있도록 이름을 정함.
size	textfield의 길이를 지정. 화면에 몇 글자가 보이도록 할 것인지를 지정. (글자수 제한 X)
value	textfield가 화면에 보일 때 textfield 부분에 표시 될 내용.
maxlength	textfield에 입력할 수 있는 최대 문자수를 지정.



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (5/17) – textfield, password

```
<h2>form control - input</h2>
<form method="post" action="login.jsp">
  <fieldset>
    <label for="userid">아이디 : </label>
    <input type="text" id="userid" name="userid" size="10"
           maxlength="16" value="아이디입력">
    <label for="pass">비밀번호 : </label>
    <input type="text" id="pass" name="pass" size="10"
           maxlength="16" placeholder="비밀번호입력">
    <input type="submit" value="로그인">
  </fieldset>
</form>
```

3-141.html

form control

localhost/html\_1...

아이디 : 아이디입력      비밀번호 : 비밀번호입력      로그인



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (6/17) – search, url, email, tel

- ✓ **type="search"** : 검색 상자 (박스 오른쪽에 x가 있어 텍스트를 쉽게 지울 수 있다.)
  - <input type="search" [속성="속성값"]>
- ✓ **type="url"** : 영문자와 마침표(.), 슬래시(/)로만 이루어진 텍스트. http://로 시작하는 사이트 주소 입력.
  - <input type="url" [속성="속성값"]>
- ✓ **type="email"** : email 형식 체크.
  - <input type="email" [속성="속성값"]>
- ✓ **type="tel"** : 전화번호 입력.
  - <input type="tel" [속성="속성값"]>

# 3. HTML 마크업 요소

## 3-8. 사용자 입력을 위한 input (7/17) – search, url, email, tel

```
<li>
    <label for="name">이름 : </label>
    <input type="search" id="name" name="name">
</li>
<li>
    <label for="mail">이메일 : </label>
    <input type="email" id="mail" name="mail">
</li>
<li>
    <label for="phone">연락처 : </label>
    <input type="tel" id="phone" name="phone">
</li>
<li>
    <label for="homepage">홈페이지 : </label>
    <input type="url" id="homepage" name="homepage">
</li>
<li><input type="submit" value="전송"></li>
```

3-142.html

form control - label

이름 : ghdrifehd

이메일 :

연락처 :

홈페이지 :

전송

form control -

이름 : 흥길동

이메일 : a

연락처 :

홈페 ! 이메일 주소에 '@'를 포함해 주세요. 'a'에 '@'가 없습니다.

전송

URL을 입력하세요.



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (8/17) – number, range

- ✓ **number, range** 필드에서 사용할 수 있는 속성.

속성	설명
min	필드에 입력할 수 있는 최소값을 지정. type="range"일 때 기본 최소값은 0.
max	필드에 입력할 수 있는 최대값을 지정. type="range"일 때 기본 최대값은 100.
step	짝수나 홀수 등 특정 숫자로 제한하려 할 대 숫자 간격을 지정. 기본값은 1.
value	페이지 로딩시 필드에 표시할 초기값.



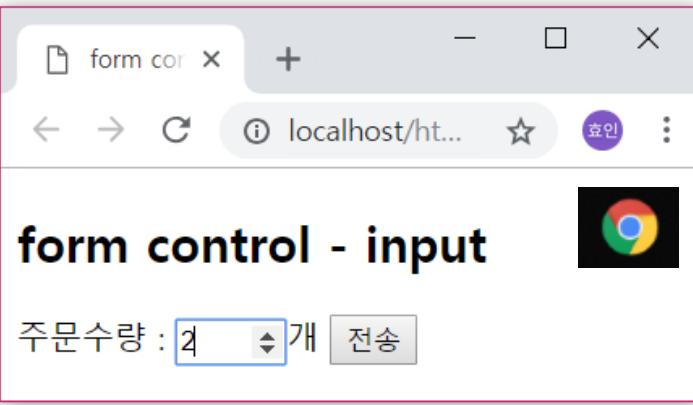
### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (9/17) – number

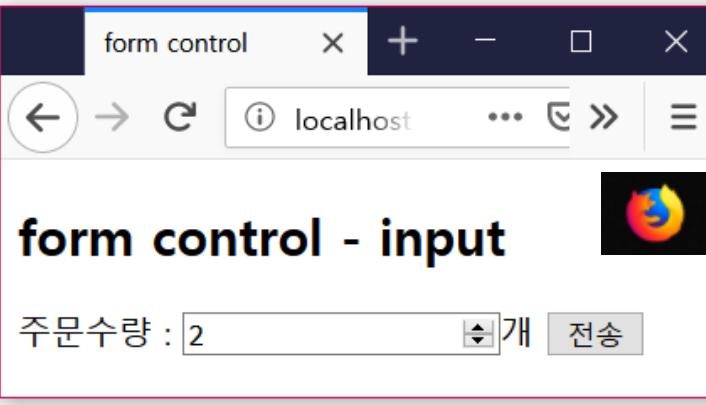
- ✓ 사용자가 입력한 내용을 숫자로 인식.
  - <input type="number" [속성="속성값"]>
- ✓ 직접 숫자를 입력하거나 브라우저에 따라 스피너박스가 표시.

```
<h2>form control - input</h2>
<form>
  <label for="pnum">주문수량 :
  <input type="number" id="pnum" name="pnum"
         min="1" max="5" value="2">개</label>
  <input type="submit" value="전송">
</form>
```

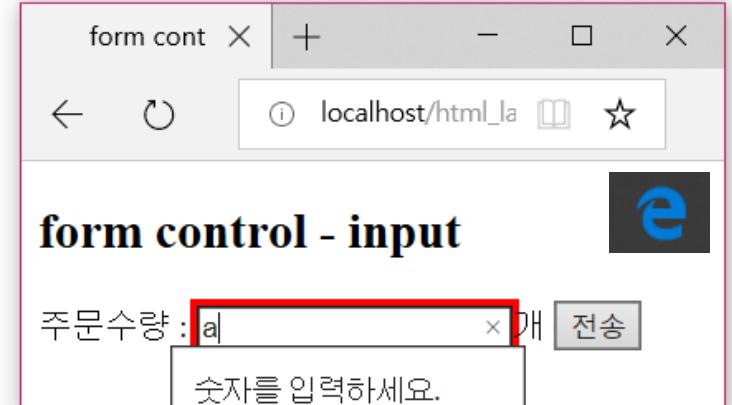
3-143.html



A screenshot of a browser window titled "form control - input". The address bar shows "localhost/html\_1a". The page content displays the text "주문수량 :  개" followed by a "전송" button. A small Chrome logo is visible in the top right corner.



A screenshot of a browser window titled "form control - input". The address bar shows "localhost/html\_1a". The page content displays the text "주문수량 :  개" followed by a "전송" button. A small Firefox logo is visible in the top right corner.



A screenshot of a browser window titled "form control - input". The address bar shows "localhost/html\_1a". The page content displays the text "주문수량 :  개" followed by a "전송" button. Below the input field, there is a red-bordered box containing the text "숫자를 입력하세요." (Please enter a number). A small Edge logo is visible in the top right corner.



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (10/17) – range

- ✓ 슬라이드 막대를 움직여 숫자 값을 입력.
  - <input type="range" [속성="속성값"]>

```
<h2>form control - input</h2>
<form>
  <label for="pnum">주문수량 :
  <input type="range" id="pnum" name="pnum"
         min="1" max="5" value="2">개</label>
  <input type="submit" value="전송">
</form>
```

3-144.html

A screenshot of a browser window titled "form control - input". The address bar shows "localhost/html\_1a". The page content includes the text "주문수량 :  개" and a "전송" button.

A screenshot of a browser window titled "form control - input". The address bar shows "localhost/html\_1a". The page content includes the text "주문수량 :  개" and a "전송" button. A blue slider handle is positioned at the value 2.

A screenshot of a browser window titled "form control - input". The address bar shows "localhost/html\_1a". The page content includes the text "주문수량 :  개" and a "전송" button. A blue slider handle is positioned at the value 2.



### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (11/17) – checkbox, radio

- ✓ **checked** 속성은 여러 개의 항목 중 선택된 항목을 표시.
- ✓ **radio button**은 name 속성의 값이 모두 일치해야 함.(name 속성이 같은 항목들 중 단일선택)
- ✓ **check box**는 name 속성의 값과는 상관없이 다중 선택이 가능.
- ✓ **<input> tag의 type 속성을 checkbox, radio로 지정.**
  - <input type="checkbox" [속성="속성값"]>
  - <input type="radio" [속성="속성값"]>

```
<h2>form control - checkbox</h2>
<h3>좋아하는 과일은?</h3>
<input type="checkbox" name="fruit" value="사과"> 사과
<input type="checkbox" name="fruit" value="수박" checked="checked"> 수박
<input type="checkbox" name="fruit" value="딸기"> 딸기
<input type="checkbox" name="fruit" value="포도"> 포도<br>
<h2>form control - radio</h2>
<h3>성별은?(label X)</h3>
<input type="radio" name="gender" value="m"> 남
<input type="radio" name="gender" value="f" checked="checked"> 여
<h3>성별은?(label O)</h3>
<label><input type="radio" name="gender" value="m"> 남</label>
<label><input type="radio" name="gender" value="f" checked="checked"> 여</label>
```

3-16.html

The screenshot shows a browser window with two sections. The top section, titled 'form control - checkbox', contains the question '좋아하는 과일은?' and four checkboxes for fruit: 사과, 수박 (checked), 딸기, and 포도. The bottom section, titled 'form control - radio', contains the question '성별은?' with two radio buttons: 남 (unchecked) and 여 (checked). Red annotations highlight the checked state of the '수박' checkbox and the checked state of the '여' radio button.

form control - checkbox

좋아하는 과일은?

사과  수박  딸기  포도

form control - radio

성별은?(label X) **checkbox부분**  
을 선택

남  여

성별은?(label O) **checkbox 또는**  
**label부분을 선택**

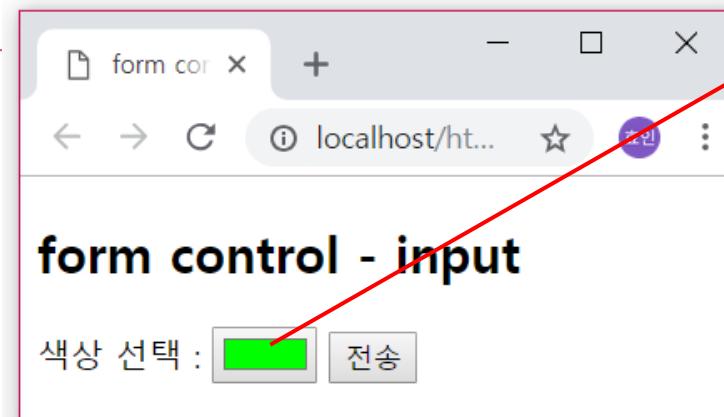
남  여

# 3. HTML 마크업 요소

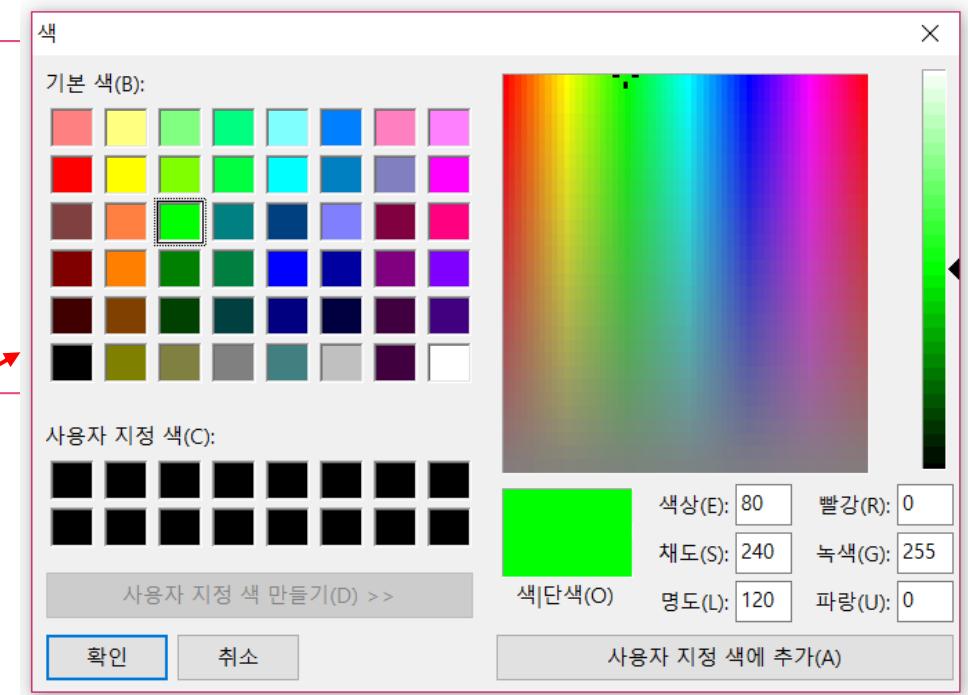
## 3-8. 사용자 입력을 위한 input (12/17) – color

- ✓ 사용자가 색상을 선택할 수 있는 색상표.
  - <input type="color" [속성="속성값"]>
- ✓ 지원 브라우저 : 파이어폭스, 크롬, 오페라와 안드로이드 브라우저.
- ✓ 그 외 브라우저는 텍스트 필드로 표시.

```
<h2>form control - input</h2>
<form>
  <label for="pnum">색상 선택 :
  <input type="color" id="pcolor" name="pcolor"
         value="#00ff00"></label>
  <input type="submit" value="전송">
</form>
```



3-145.html





### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (13/17) – 날짜, 시간

- ✓ 날짜, 시간에서 공통으로 사용할 수 있는 속성.
- ✓ 날짜 : <input type="date | month | week" [속성="속성값"]>
- ✓ 시간 : <input type="time | datetime | datetime-local" [속성="속성값"]>

속성	설명
min	날짜나 시간의 최소값을 지정.
max	날짜나 시간의 최대값을 지정.
step	스핀 박스의 화살표를 누를 때마다 날짜나 시간을 얼마나 조절할지 지정.
value	화면에 표시할 초기값 지정. type="time"일 경우, 시간은 00:00부터 23:59까지 입력하고 type="datetime"이나 type="datetime-local"일 경우, 날짜 다음에 키워드 T를 쓰고 24시간제로 시간을 지정. 예를 들어 오후 7시일 경우 T19:00.



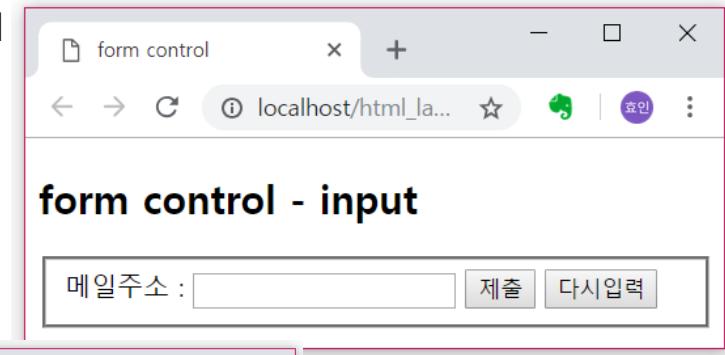
### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (14/17) – button

- ✓ <input type="submit | reset | button" [value="버튼라벨"] [속성="속성값"]>.
- ✓ reset button은 <input> 요소에 입력한 모든 정보를 초기화.
- ✓ submit button은 사용자가 입력한 form의 정보를 서버로 전송.
- ✓ button은 submit이나 reset과 같이 자체 기능은 없고, 스크립트 함수와 연결해 사용.

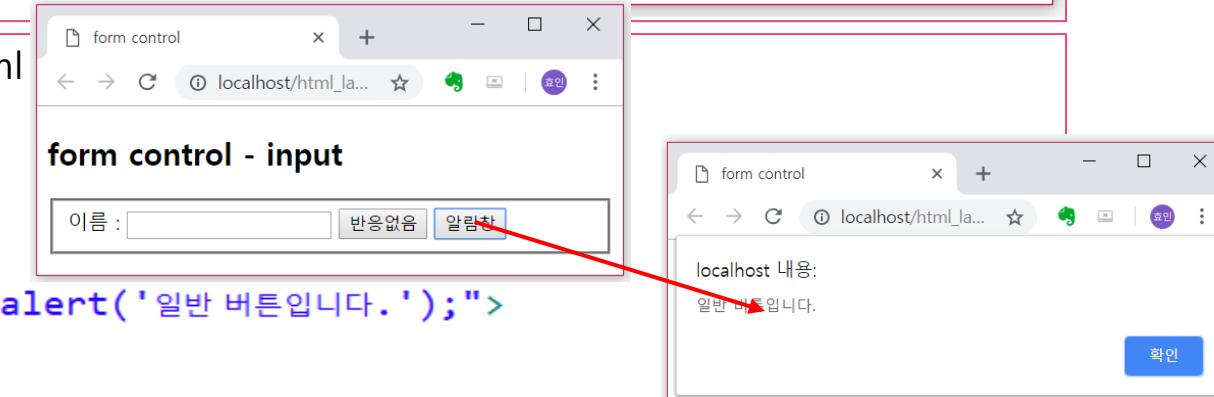
```
<h2>form control - input</h2>
<form method="post" action="mail.jsp">
  <fieldset>
    <label for="mail">메일주소 : </label>
    <input type="email" id="mail" name="mail">
    <input type="submit" value="제출">
    <input type="reset" value="다시입력">
  </fieldset>
</form>
```

3-146.html



```
<h2>form control - input</h2>
<form method="post" action="test.jsp">
  <fieldset>
    <label for="name">이름 : </label>
    <input type="text" id="name" name="name">
    <input type="button" value="반응없음">
    <input type="button" value="알림창" onclick="alert('일반 버튼입니다.');" >
  </fieldset>
</form>
```

3-147.html





### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (15/17) – image button

- ✓ image + submit.
- ✓ <input type="image" src="img path" alt="대체텍스트" [속성="속성값"]>

```
<h2>form control - input</h2>
<form method="post" action="Login.jsp">
  <fieldset>
    <label for="userid">아이디 : </label>
    <input type="text" id="userid" name="userid" size="10"
           maxlength="16" value="아이디입력">
    <label for="pass">비밀번호 : </label>
    <input type="text" id="pass" name="pass" size="10"
           maxlength="16" placeholder="비밀번호입력">
    <input type="image" src="../img/Login.png" alt="로그인">
  </fieldset>
</form>
```

3-148.html

form control - input

아이디 : 아이디입력 비밀번호 : 비밀번호입력 로그인

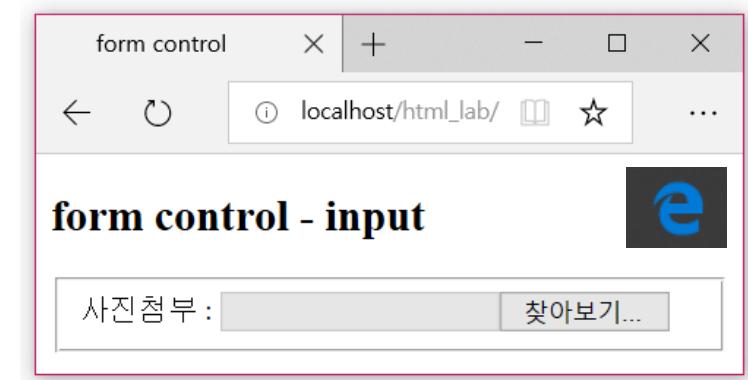
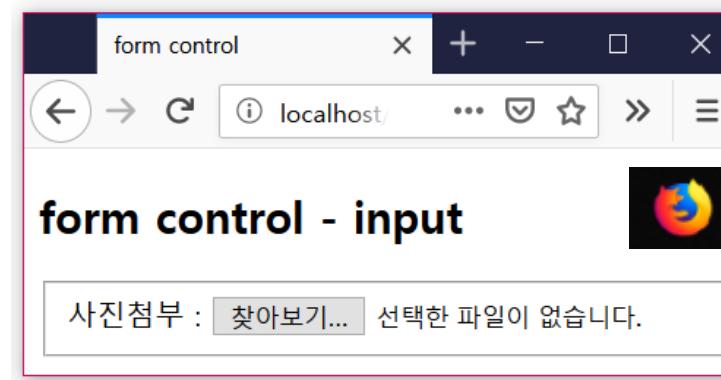
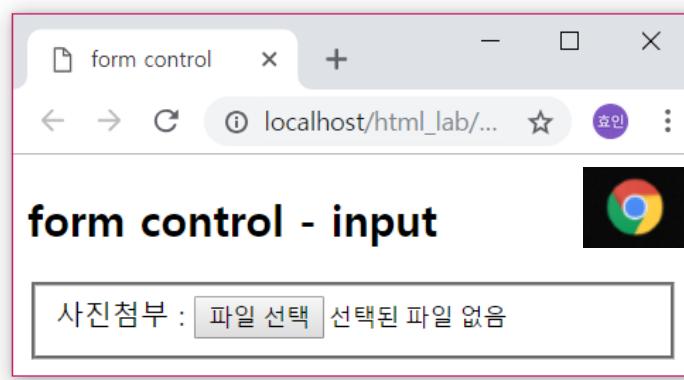
# 3. HTML 마크업 요소

## 3-8. 사용자 입력을 위한 input (16/17) – file

- ✓ **form**에 파일 첨부.

```
<h2>form control - input</h2>
<form method="post" action="register.jsp" enctype="multipart/form-data">
  <fieldset>
    <label for="picture">사진첨부 : </label>
    <input type="file" id="picture" name="picture">
  </fieldset>
</form>
```

3-149.html





### 3. HTML 마크업 요소

#### 3-8. 사용자 입력을 위한 input (17/17) – 속성

속성	설명
autofocus	페이지를 불러 오자마자 품의 요소 중에서 원하는 요소에 마우스 커서를 표시. html5이전에는 자바스크립트로 구현.
placeholder	텍스트를 입력할 때 도움이 되도록 입력란에 적당한 힌트 내용을 표시. 클릭 시 자동으로 내용이 사라짐.
readonly	입력란에 텍스트를 사용자가 직접 입력하지 못하게 읽기 전용으로 지정. readonly, readonly="readonly", readonly="true"로 표현.
required	form에 data를 입력한 후 submit 클릭 시 data를 서버로 전송하기 전 필수 입력 항목을 체크. required, required="required"로 표현.
min, max, step	min, max는 해당 필드의 최대, 최소값지정. step은 일정 간격 지정. type이 date, datetime, datetime-local, month, week, time, number, range에서 사용.
size, minlength, maxlength	minlength, maxlength는 텍스트 입력시 최대, 최소길이 지정. size는 화면에 보여지는 글자의 길이 지정.
height, width	type="image"일 때 이미지의 너비와 높이를 지정.
list	<datalist>에 미리 정의해 놓은 옵션 값을 <input> 안에 나열해 보여줌.
multiple	type="email"이나 type="file"일 때 두 개 이상의 값을 입력. <input> 태그 안에 속성 이름만 표시하면 됨.



### 3. HTML 마크업 요소

#### 3-9. 여러 data 나열 (1/6) – dropdown

- ✓ <select> tag는 select box(dropdown)를 표시.
- ✓ <option> tag는 select box에 포함될 항목들을 정의.
- ✓ selected 속성은 여러 개의 항목 중 선택된 항목을 표시.
- ✓ value 속성은 각 항목 값을 지정하기 위해 사용.

```
<h2>form control - select</h2>
<select name="area">
    <option value="010">핸드폰
    <option value="02" selected="selected">서울
    <option value="031">경기도
    <option value="041">충청도
    <option value="051">경상도
    <option value="061">전라도
</select>
```

The image displays two screenshots of a web browser window titled "form control - select". Both screenshots show a dropdown menu with the following options: 핸드폰, 서울, 경기도, 충청도, 경상도, and 전라도. In the top screenshot, the option "서울" is selected. In the bottom screenshot, the option "서울" is highlighted with a blue selection bar.

3-14.html



### 3. HTML 마크업 요소

#### 3-9. 여러 data 나열 (2/6) – dropdown

✓ 사용자 입력이 아닌 여러 옵션 중에서 선택하는 목록.

✓ <select 속성="속성값">

```
<option value="값" [속성="속성값"]>내용1</option>
```

```
<option value="값" [속성="속성값"]>내용2</option>
```

.....

```
</select>
```

<select> 속성

##### 속성

##### 설명

size      화면에 표시될 dropdown 메뉴의 항목 개수 지정.

multiple      브라우저 화면에 여러 개의 옵션이 함께 표시되면서 ctrl 키를 누른 상태로 여러 항목을 선택(다중선택).

<option> 속성

##### 속성

##### 설명

value      옵션을 선택했을 때 서버로 넘겨질 값을 지정.

selected      화면에 표시될 때 기본으로 선택되어 있는 옵션을 지정.



### 3. HTML 마크업 요소

#### 3-9. 여러 data 나열 (3/6) – dropdown

```
<label for="chanel">선호채널</label>
<select id="chanel" name="chanel">
  <option value="mbc">MBC
  <option value="kbs" selected="selected">KBS
  <option value="sbs">SBS
  <option value="jtbc">JTBC
  <option value="tvn">TVN
  <option value="ytn">YTN
</select>
```

3-1411.html

A screenshot of a web browser window titled "form control". The page content is "form control - select". It contains a form with two fields: "설문조사" and "선호채널". The "선호채널" field has a dropdown menu open, showing options: MBC, KBS, SBS, JTBC, and TVN. The option "KBS" is highlighted with a blue selection bar.

```
<label for="chanel">선호채널</label>
<select id="chanel" name="chanel" size="5" multiple="multiple">
  <option value="mbc">MBC
  <option value="kbs" selected="selected">KBS
  <option value="sbs">SBS
  <option value="jtbc">JTBC
  <option value="tvn">TVN
  <option value="ytn">YTN
</select>
```

3-1412.html

A screenshot of a web browser window titled "form control". The page content is "form control - select". It contains a form with two fields: "설문조사" and "선호채널". The "선호채널" field has a dropdown menu open, showing options: MBC, KBS, SBS, JTBC, and TVN. Multiple options are selected: MBC, KBS, SBS, and TVN. The option "JTBC" is highlighted with a blue selection bar.



### 3. HTML 마크업 요소

#### 3-9. 여러 data 나열 (4/6) – dropdown

- ✓ <optgroup> : dropdown 목록에서 여러 항목들을 몇 가지 그룹으로 묶을 경우.
- ✓ label 속성을 이용하여 그룹의 제목을 지정.

```
<label for="chanel">선후채널</label>
<select id="chanel" name="chanel">
    <optgroup label="지상파">
        <option value="mbc">MBC
        <option value="kbs" selected="selected">KBS
        <option value="sbs">SBS
    </optgroup>
    <optgroup label="종합편성채널">
        <option value="jtbc">JTBC
        <option value="tvn">TVN
        <option value="ytn">YTN
    </optgroup>
</select>
```

3-1413.html

The screenshot shows a web browser window titled 'form control'. The URL bar displays 'localhost/html\_lab/c...'. The main content area has a heading 'form control - select'. Below it, there is a form with two fields: '설문조사' and '선후채널'. The '선후채널' field is a dropdown menu containing the following options:

- 이름 [input field]
- 선후채널 [dropdown menu]
  - 지상파
  - MBC
  - KBS
  - SBS
  - 종합편성채널
  - JTBC
  - TVN
  - YTN



### 3. HTML 마크업 요소

#### 3-9. 여러 data 나열 (5/6) – datalist

- ✓ <input>과 함께 사용하여 텍스트필드에 직접 값을 입력하는 것이 아니라 datalist의 선택 값이 텍스트 필드에 입력됨.

- ✓ <input type="text" list="datalist id">  
<datalist id="datalist id">  
    <option>.....</option>  
    <option>.....</option>  
    .....  
  </datalist>

<option> 속성

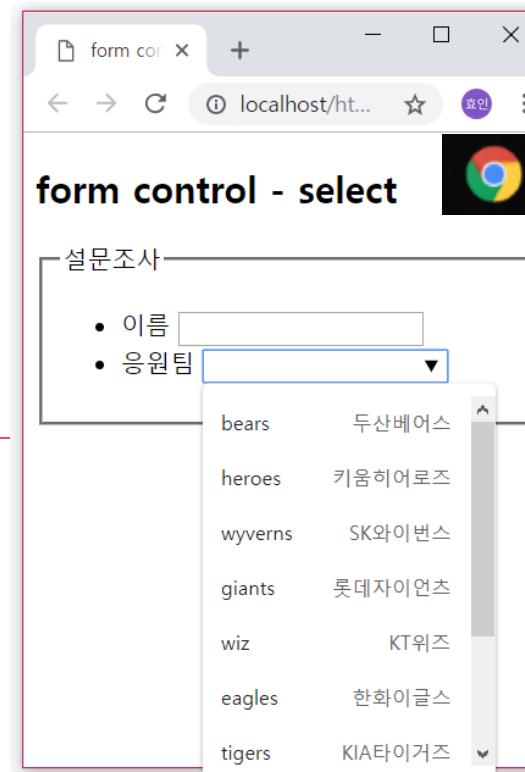
속성	설명
value	사용자가 레이블을 선택했을 때 서버로 넘겨질 값 지정.
label	사용자를 위해 브라우저에 표시할 레이블. 따로 지정하지 않을 경우, value값을 레이블로 사용.

# 3. HTML 마크업 요소

## 3-9. 여러 data 나열 (6/6) – datalist

```
<label for="bbteam">응원팀</label>
<input type="text" id="bbteam" list="teamList">
<datalist id="teamList">
    <option value="bears" label="두산베어스"></option>
    <option value="heroes" label="키움히어로즈"></option>
    <option value="wyverns" label="SK와이번스"></option>
    <option value="giants" label="롯데자이언츠"></option>
    <option value="wiz" label="KT위즈"></option>
    <option value="eagles" label="한화이글스"></option>
    <option value="tigers" label="KIA타이거즈"></option>
    <option value="Lions" label="삼성라이온즈"></option>
    <option value="twins" label="LG트윈스"></option>
    <option value="dinos" label="NC다이노스"></option>
</datalist>
```

3-1414.html



Three screenshots of different browsers showing the same dropdown menu for the "응원팀" input field. Each browser has a different title bar and icon. The dropdown menu lists the same ten options: bears (두산베어스), heroes (키움히어로즈), wyverns (SK와이번스), giants (롯데자이언츠), wiz (KT위즈), eagles (한화이글스), tigers (KIA타이거즈), Lions (삼성라이온즈), twins (LG트윈스), and dinos (NC다이노스). The menu is displayed over a form with fields for "이름" and "응원팀".



### 3. HTML 마크업 요소

#### 3-10. 여러 줄 입력 – textarea

- ✓ <textarea> tag는 여러 줄을 입력할 수 있는 box를 표시.
- ✓ cols와 rows 속성은 text box의 크기를 지정.
- ✓ <textarea></textarea> tag 사이의 문자열은 text box에 표시.
- ✓ disabled 속성은 화면에 표시는 하되 데이터를 수정할 수 없도록 비활성화 상태로 표시.

```
<h2>form control - textarea</h2>
<textarea rows="10" cols="30">
이 영역은 text area에 보입니다.
</textarea>
<br>
<h2>form control - textarea disabled</h2>
<textarea rows="10" cols="30" disabled="disabled">
이 영역은 text area에 보입니다.
보이긴 하지만 비활성화 되어있습니다.
</textarea>
```

3-15.html

The image displays two side-by-side browser windows. Both windows have a title bar labeled 'form control - text'. The left window shows a standard text area containing the text '이 영역은 text area에 보입니다.'. The right window shows a text area with the same text, but it is visually different, appearing grayed out or less prominent, and a tooltip at the bottom of the window states '보이긴 하지만 비활성화 되어있습니다.' This visual difference serves as a visual cue for the 'disabled' state of the input field.



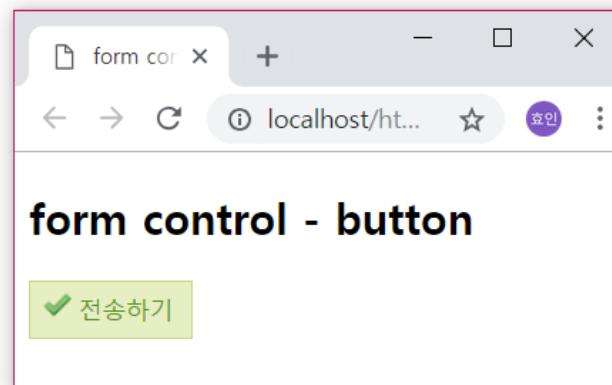
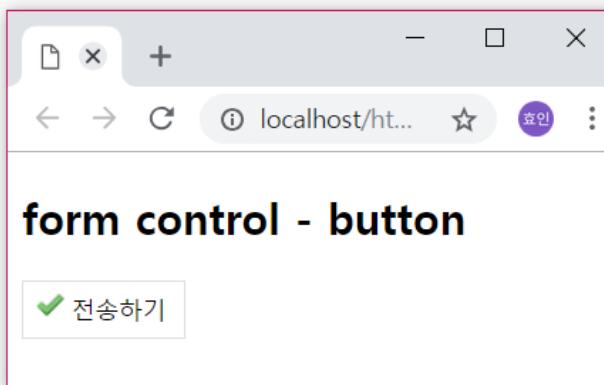
### 3. HTML 마크업 요소

#### 3-11. 기타 form control (1/4) – button

- ✓ <button> 태그의 type 속성은 버튼이 활성화 되었을 때 어떤 동작을 할지 지정. 기본은 submit.
- ✓ <button [type="submit | reset | button"]>내용</button>
- ✓ input과의 차이점은 <button> 태그는 contents를 포함할 수 있기 때문에 아이콘을 추가할 수 있고, CSS를 이용하여 원하는 형태로 꾸밀 수 있다.

```
<h2>form control - button</h2>
<form>
  <button type="submit" class="subm">
     전송하기
  </button>
</form>
```

3-151.html



CSS

```
.subm{ /* 버튼 스타일 */
  display:block; /* 블록 레벨 요소 */
  background-color:#fff; /* 배경색 */
  border:1px solid #dedede; /* 테두리 */
  cursor:pointer; /* 마우스 포인터 */
  padding:5px 10px 6px 7px; /* 패딩 */
}
.subm img{ /* 버튼 내 원쪽 이미지 */
  border:0; /* 테두리 없음 */
  padding:0; /* 패딩 없음 */
  width:16px; /* 가로 크기 */
  height:16px; /* 세로 크기 */
}
.subm:hover{ /* 버튼 위로 마우스 포인터 올렸을 때 스타일 */
  background-color:#e6efc2; /* 배경색 */
  border:1px solid #c6d880; /* 테두리 */
  color:#529215; /* 글자색 */
}
```



### 3. HTML 마크업 요소

#### 3-11. 기타 form control (2/4) – progress

- ✓ 작업의 진행 상태를 표시.
- ✓ 작업의 시작을 0, 최종 완료를 최대값으로 설정.
- ✓ <progress value="값" [max="값"]></progress>

##### 속성

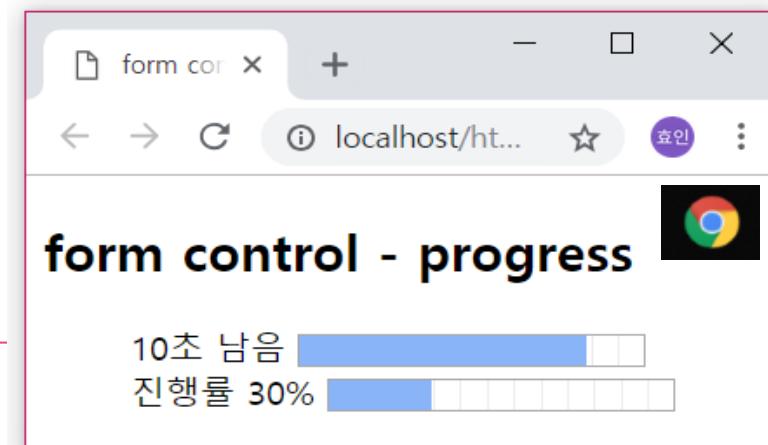
##### 설명

value      작업 진행 상태를 나타내며 부동 소수점으로 표현. 0보다 크거나 같고 max 값보다는 작거나 같아야 한다. 만약 max 값이 지정되지 않았다면 이 값은 1.0 보다 작아야 한다.

max      작업이 완료되려면 얼마나 많은 작업을 해야 하는지 부동 소수점으로 표현. 0보다 커야한다.

```
<li>
  <label>10초 남음</label>
  <progress value="50" max="60"></progress>
</li>
<li>
  <label>진행률 30%</label>
  <progress value="30" max="100"></progress>
</li>
```

3-150.html





### 3. HTML 마크업 요소

#### 3-11. 기타 form control (3/4) – meter

- ✓ 값이 차지하는 크기 표시.
- ✓ <progress>와 결과 화면은 비슷하지만 차이점은 <progress>는 작업의 진행 상황을 나타내는 반면, <meter>는 전체 크기 중에서 얼마나 차지하는지를 표현.
- ✓ <meter value="값" [속성="속성값"]></meter>

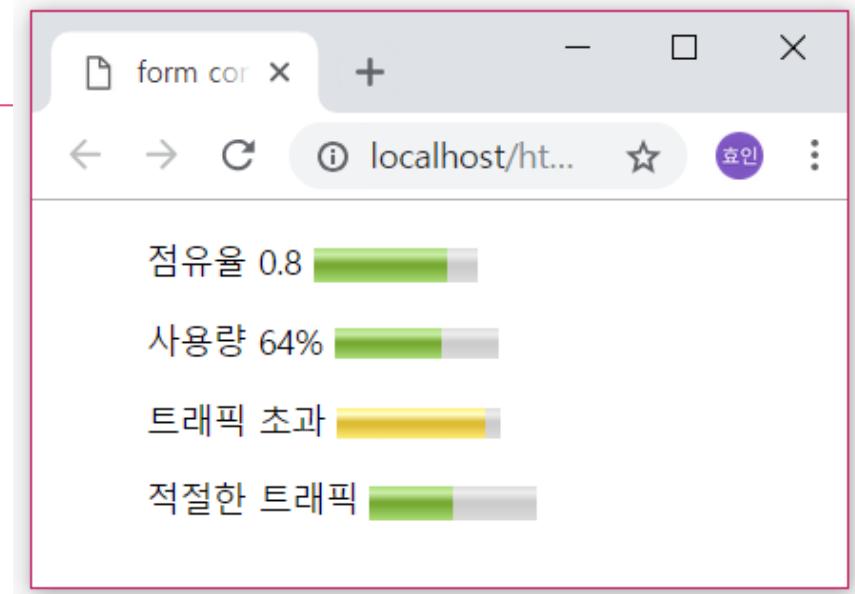
속성	설명
min, max	범위의 최소값과 최대값을 지정. 값을 정하지 않으면 0과 1로 지정.
value	범위 내에서 차지하는 값.
low	"이 정도면 낮다."라고 할 정도의 값을 지정.
high	"이 정도면 높다."라고 할 정도의 값을 지정.
optimum	"이 정도면 적당하다."라고 할 정도의 범위를 지정. Optimum값이 high값보다 크다면 value값이 클수록 좋고 optimum 값이 low값보다 작다면 값이 작을 수록 좋다.



### 3. HTML 마크업 요소

#### 3-11. 기타 form control (4/4) – meter

```
<ul>
  <li>
    <label>점유율 0.8 </label>
    <!-- 전체 크기 1을 기준으로 0.8만큼 차지합니다 -->
    <meter value="0.8"></meter>
  </li>
  <li>
    <label>사용량 64%</label>
    <!-- 전체 100 중에서 64만큼 차지합니다 -->
    <meter min="0" max="100" value="64"></meter>
  </li>
  <li>
    <label>트래픽 초과</label>
    <!-- 전체 크기는 1024~10240까지인데 높다고 설정한 8192보다 현재 값 9216이 더 큽니다 -->
    <meter min="1024" max="10240" low="2048" high="8192" value="9216"></meter>
  </li>
  <li>
    <label>적절한 트래픽</label>
    <!-- 전체 1 중에서 현재 0.5를 차지하고 있으며 적정도를 0.8로 설정했습니다 -->
    <meter value="0.5" optimum="0.8"></meter>
  </li>
</ul>
```





### 3. HTML 마크업 요소

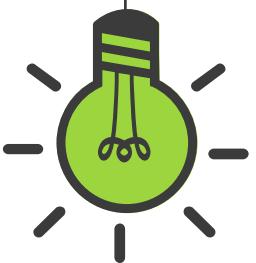
#### 3-8. 사용자 입력을 위한 input (/) -

- ✓ **form control**을 조합하여 완성.

The screenshot shows a web browser window with the title bar 'form control'. The address bar displays 'localhost/html\_lab/chap3/3-17.html'. The main content area contains a form titled 'form control test - 회원가입'. The form consists of several input fields and controls:

- 이름: An input field for entering a name.
- 아이디: An input field for entering an ID, with a '중복확인' button to its right.
- 비밀번호: An input field for entering a password.
- 비밀번호확인: An input field for confirming the password.
- 성별: A gender selection section with radio buttons for '남' (Male) and '여' (Female), where '여' is selected.
- 좋아하는 과일: A section for selecting favorite fruits, featuring checkboxes for '사과', '수박' (Watermelon), '딸기' (Strawberry), and '포도' (Grape). '수박' is checked.
- 이메일: An input field for email, with a dropdown menu showing '@ 네이버' and a suffix input field.
- 연락처: An input field for phone number, with a dropdown menu showing '02' and a suffix input field.
- Buttons at the bottom: '회원가입' (Join Member) and '초기화' (Reset).

# 04 CSS 기본





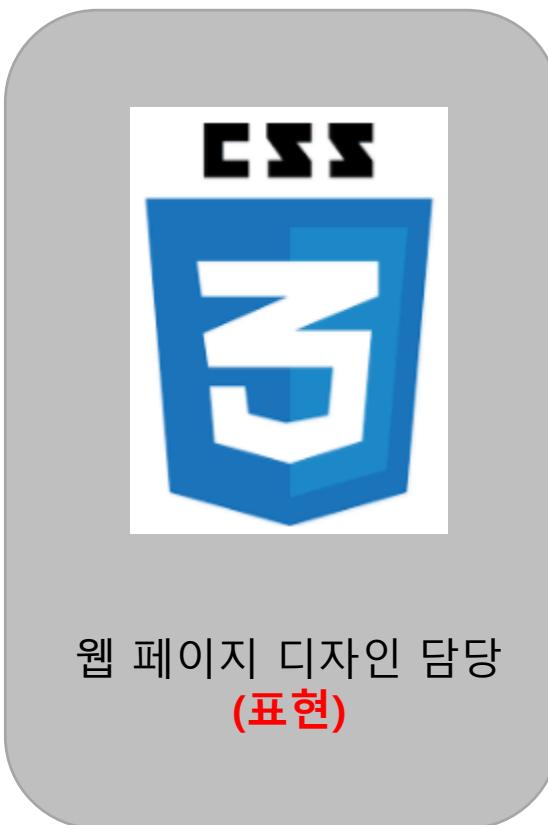
## 4. CSS 기본

### 4-1. CSS?

- ✓ HTML5 웹 문서를 구성하는 3가지 요소.



웹 페이지 문서 담당  
(구조)



웹 페이지 디자인 담당  
(표현)



웹 페이지 이벤트 담당  
(동작)



## 4. CSS 기본

### 4-1. CSS?

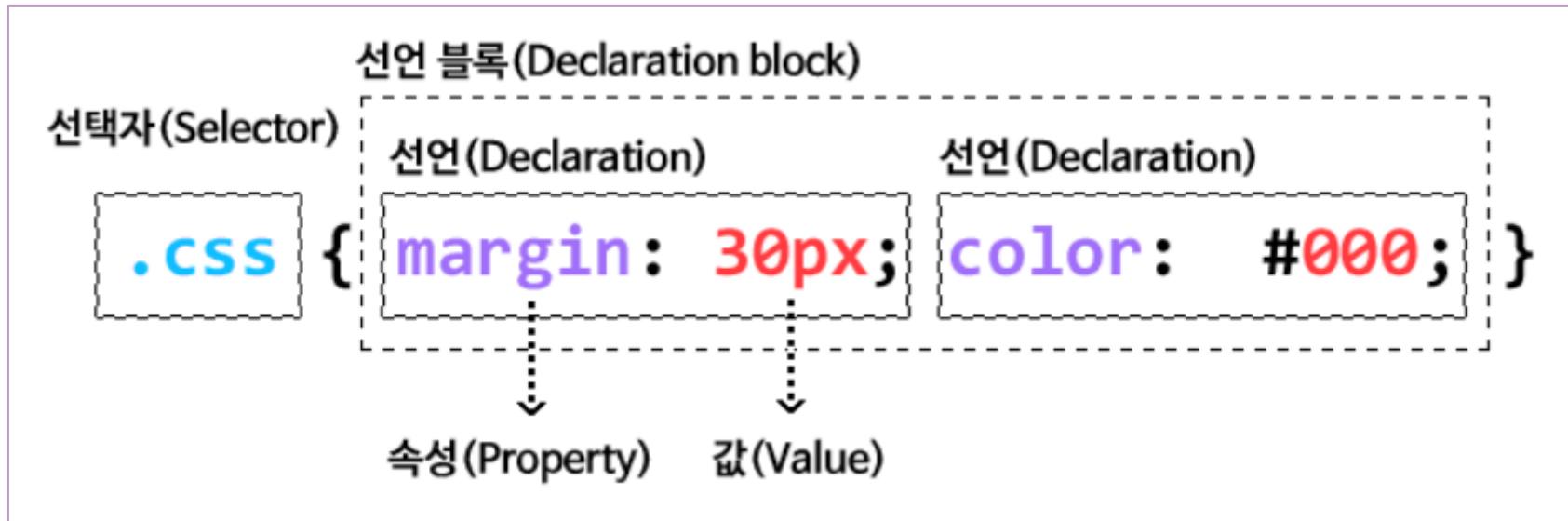
- ✓ Cascading Style Sheets의 약자.
- ✓ 스타일 사용이유.
  - 웹 문서의 내용과 상관없이 디자인만 바꿀 수 있다.
    - [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
  - 다양한 기기에 맞게 탄력적으로 바꿔는 문서를 만들 수 있다.
- ✓ 웹 페이지를 표현하기 위한 스타일 규칙을 모아 놓은 문서.
- ✓ 웹 브라우저 별 CSS3 지원.
  - 테스트 사이트 <http://css3test.com>



## 4. CSS 기본

### 4-1. CSS?

- ✓ CSS 규칙은 선택자(selector)와 선언(declaration) 두 부분으로 구성.
- ✓ 선택자는 규칙이 적용되는 엘리먼트.
- ✓ 선언 부분에서는 선택자에 적용될 스타일을 작성.
- ✓ 선언은 중괄호로 감싸며, 속성(property)과 값(value)로 이루어짐.

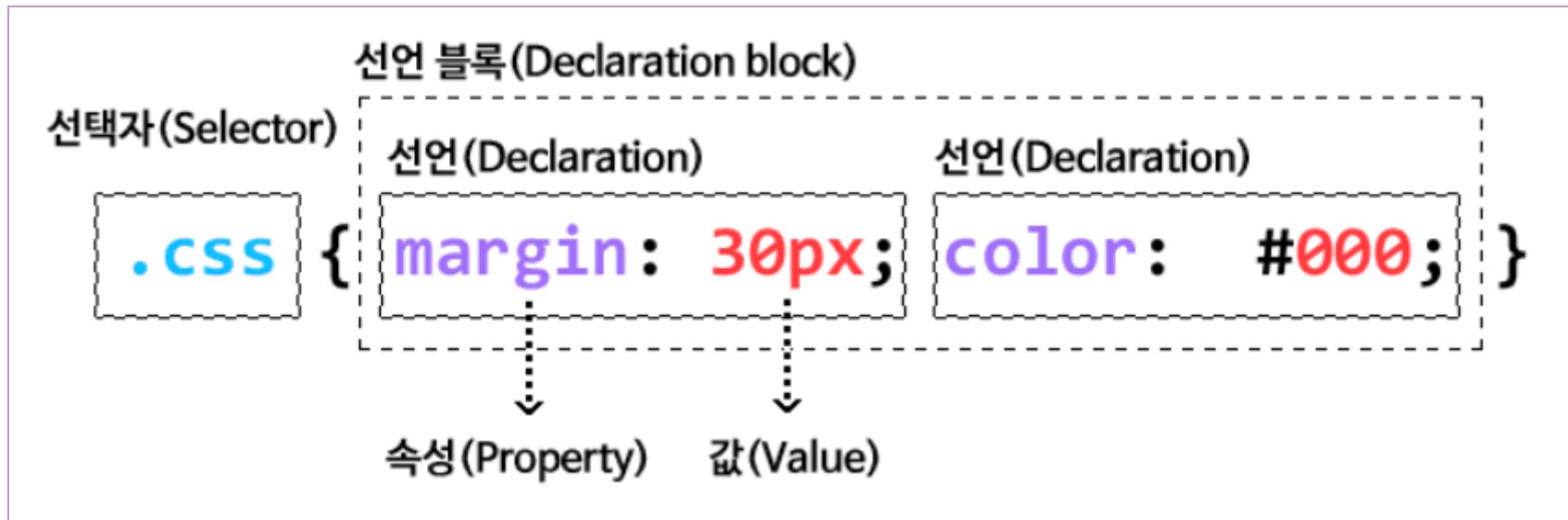




## 4. CSS 기본

### 4-1. CSS?

- ✓ 속성(property)은 선택자에서 바꾸고 싶은 요소.(color, font, width, height, border, ....)
- ✓ 값(value)은 속성에 적용할 값.
- ✓ 여러 선택자에 동일한 스타일을 적용할 때, comma(,)로 구분하여 나열.(선택자 그룹화)
- ✓ 선언 안에 하나 이상의 속성을 작성할 수 있으며, 각 속성은 semi-colon(;)으로 구분.
- ✓ 주석 : /\* 내용 \*/



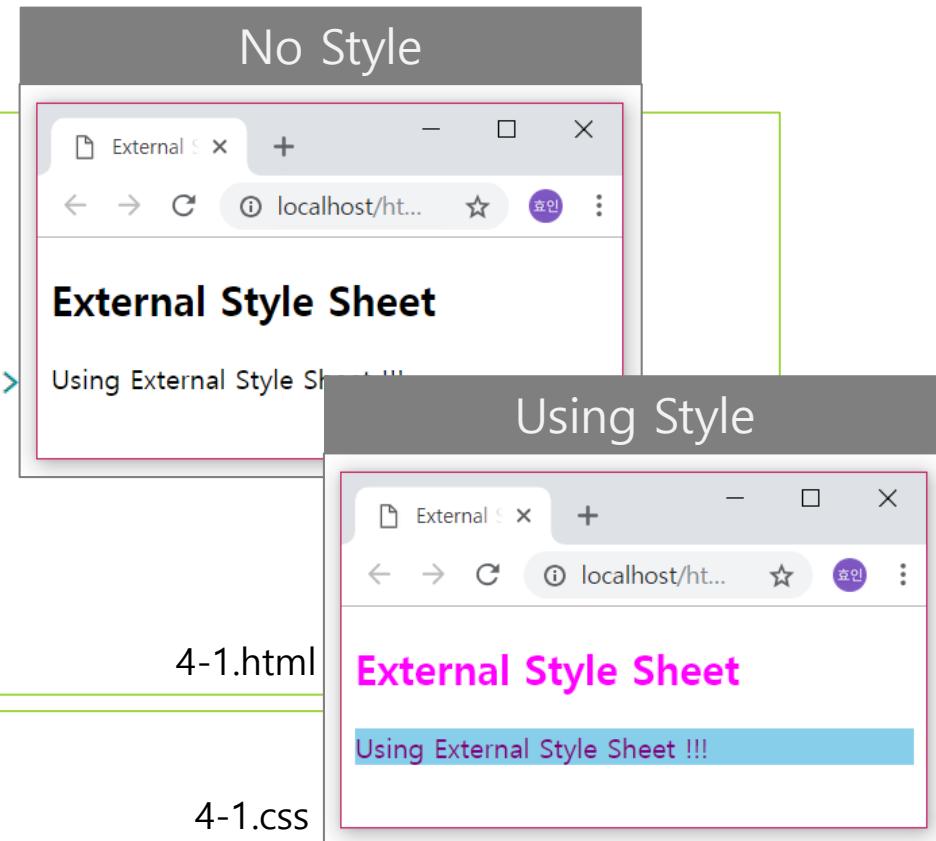
# 4. CSS 기본

## 4-2. 외부 스타일 시트 적용 (1/2)

- ✓ <link>를 사용하여 외부 스타일 시트를 적용.
- ✓ <link>는 <head>안에 작성하며 rel, type, href 3가지 속성을 주로 사용.
- ✓ rel은 HTML 페이지와 링크된 파일간의 관계를 의미.
- ✓ href는 CSS file 경로를 의미함.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>External Style Sheet</title>
    <link rel="stylesheet" type="text/css" href="../css/4-1.css">
</head>
<body>
    <h2>External Style Sheet</h2>
    <p>Using External Style Sheet !!!</p>
</body>
</html>
```

```
h2 {color: magenta;}
p {background-color: skyblue; color: purple;}
```



4-1.html

4-1.css

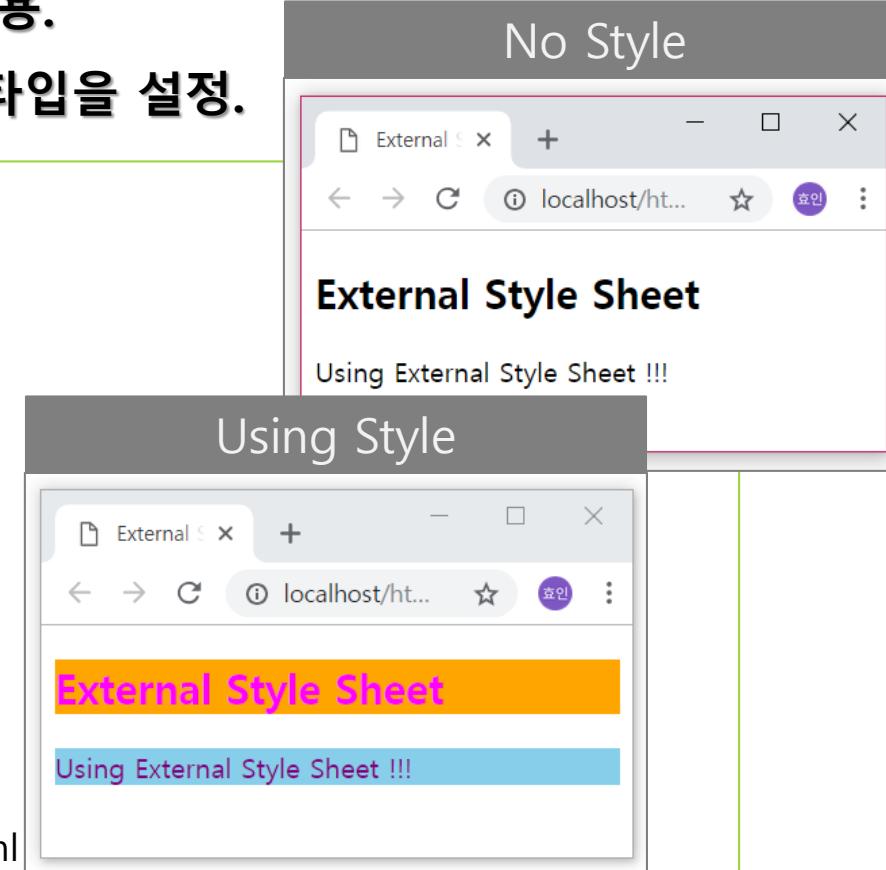


# 4. CSS 기본

## 4-2. 외부 스타일 시트 적용 (2/2)

- ✓ **@import**를 사용하여 외부 스타일 시트를 적용.
- ✓ **@import**는 스타일 시트 중 최상단에 위치해야 함.
- ✓ **@import url("file path");** 또는 **@import "file path";** 형태로 사용.
- ✓ <link>와 달리 <style>의 media 속성을 통해 보여지는 미디어 타입을 설정.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>External Style Sheet</title>
    <style type="text/css" media="screen">
        @import url("../css/4-1.css");
        /* @import "../css/4-1.css"; */
        h2 {background-color: orange;}
    </style>
</head>
<body>
    <h2>External Style Sheet</h2>
    <p>Using External Style Sheet !!!</p>
</body>
</html>
```



The image shows two browser windows side-by-side. The top window is titled 'No Style' and displays the text 'Using External Style Sheet !!!' in a plain black font. The bottom window is titled 'Using Style' and displays the same text, but the entire page has a yellow background color, indicating that the external style sheet containing the 'background-color: orange;' rule was successfully applied.

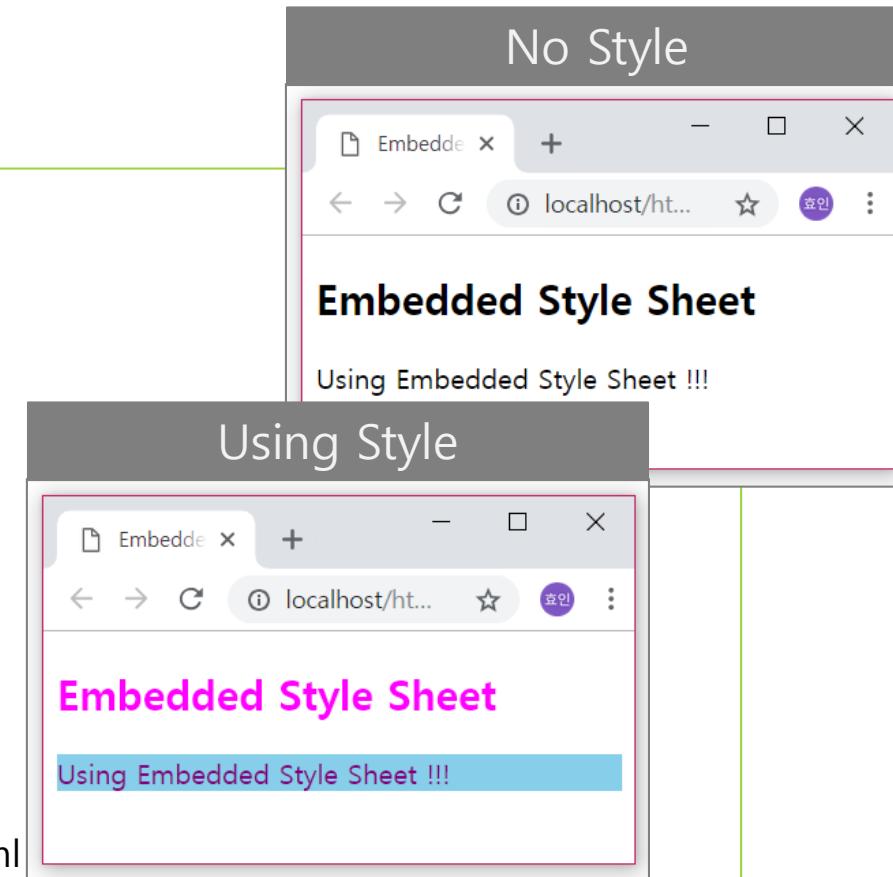


# 4. CSS 기본

## 4-3. 내부 스타일 시트 적용

- ✓ <style>을 사용하여 내부 스타일 시트를 적용.
- ✓ <style> tag 내부에 CSS 규칙을 작성.
- ✓ 외부 스타일 시트보다 우선 적용.
- ✓ <head> tag 내부에 작성.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Embedded Style Sheet</title>
    <style type="text/css">
        h2 {color: magenta;}
        p {background-color: skyblue; color: purple;}
    </style>
</head>
<body>
    <h2>Embedded Style Sheet</h2>
    <p>Using Embedded Style Sheet !!!</p>
</body>
</html>
```



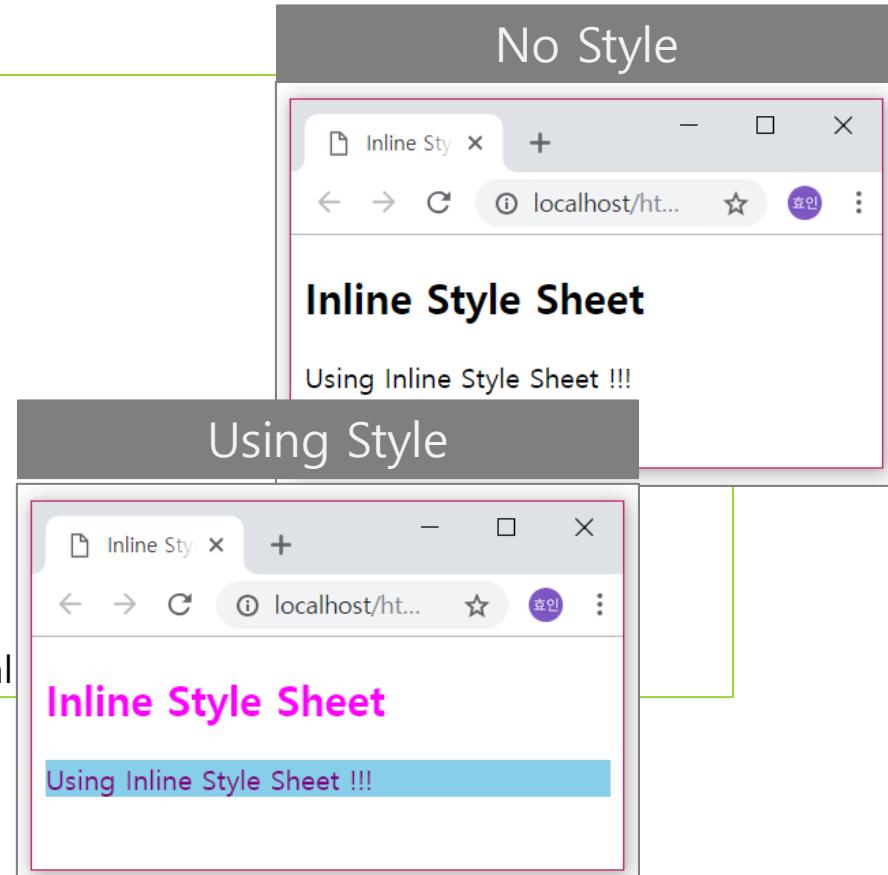
# 4. CSS 기본

## 4-4. 인라인 스타일 적용

- ✓ 개별 element마다 스타일을 지정하므로 유지보수에 용이하지 않다.
- ✓ 스타일 적용 우선순위는 “[인라인 스타일](#) > [내부 스타일 시트](#) > [외부 스타일](#)” 시트 순.
- ✓ **Style** 속성을 사용하고 속성값으로 CSS규칙을 작성.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Inline Style Sheet</title>
</head>
<body>
    <h2 style="color: magenta;">Inline Style Sheet</h2>
    <p style="background-color: skyblue; color: purple;">
        Using Inline Style Sheet !!!
    </p>
</body>
</html>
```

4-4.html





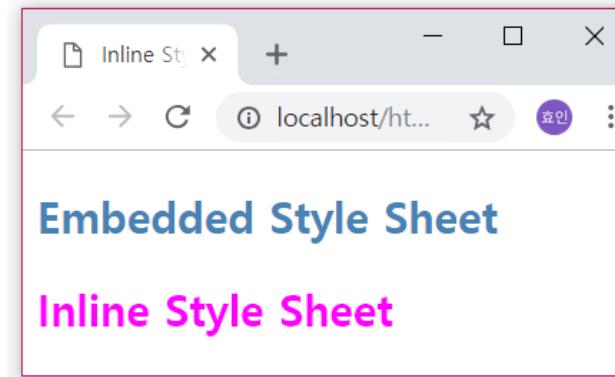
## 4. CSS 기본

### 4-5. 스타일 우선순위

- ✓ 스타일 적용 우선순위는 “**인라인 스타일 > 내부 스타일 시트 > 외부 스타일 시트**” 시트 순.

```
<style type="text/css">
  h2 {color: steelblue;}
</style>
</head>
<body>
  <h2>Embedded Style Sheet</h2>
  <h2 style="color: magenta;">Inline Style Sheet</h2>
</body>
```

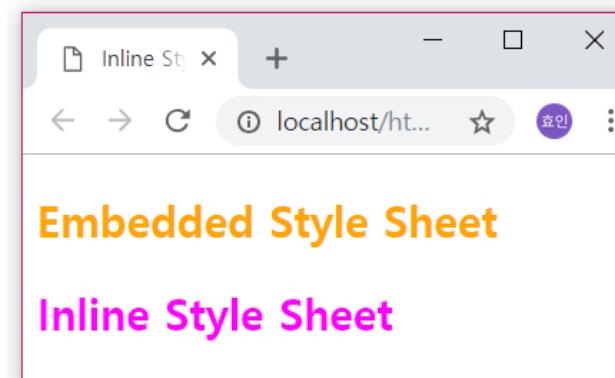
4-5.html



```
<link rel="stylesheet" type="text/css" href="../css/4-1.css">
<style type="text/css">
  h2.em {color: orange;}
</style>
</head>
<body>
  <h2 class="em">Embedded Style Sheet</h2>
  <h2>External Style Sheet</h2>
</body>
```

```
h2 {color: magenta;}
h2.em {color: steelblue;}
```

4-6.html





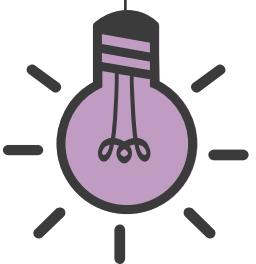
## 4. CSS 기본

### 4-6. 요약

- ✓ <link rel="stylesheet" href="../css/style.css">를 사용하여 외부 스타일시트를 적용.
- ✓ 내부 스타일 시트는 <head> tag 내부에 <style></style> tag를 정의하고 CSS를 정의.
- ✓ 내부 스타일시트는 외부 스타일시트보다 우선 적용.
- ✓ 인라인 스타일을 이용해서 각 element마다 스타일을 적용할 수 있는 방법도 있지만 유지보수에 용이하지 않다.



# 05 CSS 이해





# 5. CSS 이해

## 5-1. 선택자의 이해 (1/15) - 개요

- ✓ HTML 문서에서 CSS 규칙 적용 타겟이 되는 다양한 종류의 CSS 선택자(selector)가 존재.
- ✓ 일반 선택자는 전체 선택자, 타입 선택자, 클래스 선택자, ID 선택자로 분류.
- ✓ 복합 선택자는 자식 선택자, 하위 선택자, 인접 형제 선택자, 일반 형제 선택자로 분류.
- ✓ 그 외에도 가상 클래스 선택자, 가상 엘리먼트 선택자, 속성 선택자가 존재.

선택자	의미	사용법	CSS
전체 선택자	HTML 문서 내 모든 element 선택	* { }	2
타입 선택자	매칭되는 element 선택	h1, h2, h3 { }	1
클래스 선택자	class 속성 값과 매칭되는 element 선택	.className { }	1
ID 선택자	id 속성 값과 매칭되는 element 선택	#idName { }	1
하위 선택자	하위 element 선택	E1 E2 { }	1
자식 선택자	직속 하위 element 선택	E1 > E2 { }	2
인접 형제 선택자	인접 형제(sibling) 관계인 element 선택	E1 + E2 { }	2
일반 형제 선택자	형제(sibling) 관계인 element 선택	E1 ~ E2 { }	3

복합 선택자

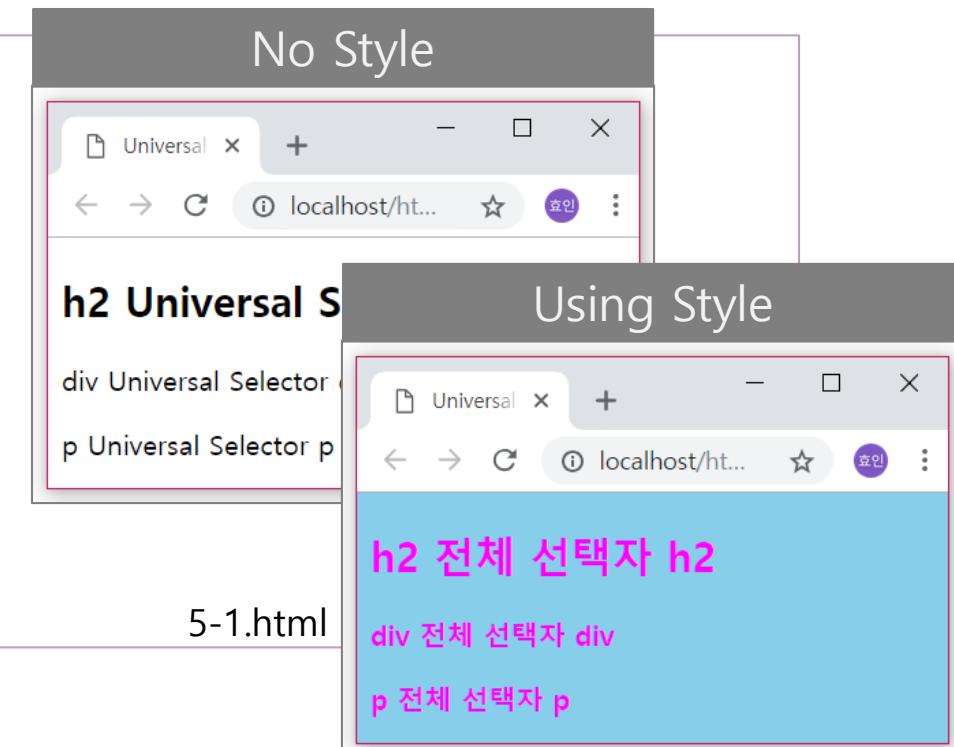


# 5. CSS 이해

## 5-1. 선택자의 이해 (2/15) – 일반 선택자 요소 (1/4)

- ✓ 전체 선택자 (Universal Selector) 사용법은 “`* { }`” 이다.
- ✓ HTML 문서 내 모든 `element`를 선택.
- ✓ 잘 사용되지 않으며 우선 순위가 가장 낮다.
- ✓ 일반 선택자의 우선 순위는 **전체 선택자 < 타입 선택자 < 클래스 선택자 < ID 선택자** 이다.

```
<style type="text/css">
* {
    background: skyblue;
    color: magenta;
    font-weight: bold;
}
</style>
</head>
<body>
    <h2>h2 Universal Selector h2</h2>
    <div>div Universal Selector div</div>
    <p>p Universal Selector p</p>
</body>
```



5-1.html



# 5. CSS 이해

## 5-1. 선택자의 이해 (3/15) – 일반 선택자 요소 (2/4)

- ✓ 타입 선택자(Type Selector) 사용법은 “`elementType { }`” 이다.
- ✓ 태그명을 이용해서 스타일을 적용할 태그를 선택.
- ✓ 1개 이상의 HTML 엘리먼트를 사용할 수 있다.
- ✓ 여러 엘리먼트를 선택할 때에는 콤마(,)로 구분.

```
<style type="text/css">
div, p {
    padding: 10px; margin: 10px; font-weight: bold;
}
div {
    background: skyblue; color: magenta;
}
p {
    background: Lightgray; color: orange;
}
</style>
</head>
<body>
    <h2>h2 Type Selector h2</h2>
    <div>div Type Selector div</div>
    <p>p Type Selector p</p>
</body>
```

The screenshot shows three browser windows side-by-side, each displaying a different example of a type selector:

- No Style**: Shows the initial state of the page with no styles applied.
- Using Style**: Shows the page after applying the CSS rules from the code above. The `h2`, `div`, and `p` elements are styled according to the defined rules.
- h2 Type Selector h2**: A zoomed-in view of the first `h2` element, showing its bold font weight and black color.
- div Type Selector div**: A zoomed-in view of the first `div` element, showing its blue background and magenta color.
- p Type Selector p**: A zoomed-in view of the first `p` element, showing its gray background and orange color.



# 5. CSS 이해

## 5-1. 선택자의 이해 (4/15) – 일반 선택자 요소 (3/4)

- ✓ 클래스 선택자(Class Selector) 사용법은 “`.className { }`” 이다.
- ✓ 클래스 명은 공백 없이 대소문자 또는 Hypen(-), UnderScore(\_)로 시작.(기호나 숫자 시작 X.)
- ✓ HTML 문서에서 동일한 클래스 명을 중복해서 사용 가능.
- ✓ `class` 속성 값에 하나 이상의 클래스를 적용 가능.

```
<style type="text/css">
  .target1 {
    background: skyblue; color: magenta;
  }
  .target2 {
    background: Lightgray; color: orange;
  }
  p.target1 {
    font-weight: bold; font-size: 20px;
  }
</style>
</head>
<body>
  <h2>h2 Class Selector h2</h2>
  <div class="target1">div Class Selector div</div>
  <p class="target2">p Class Selector p</p>
  <p class="target1">p Class Selector p</p>
</body>
```

No Style

h2 Class Selector h2

div Class Selector div

p Class Selector p

p Class Selector p

Using Style

h2 Class Selector h2

div Class Selector div

p Class Selector p

p Class Selector p

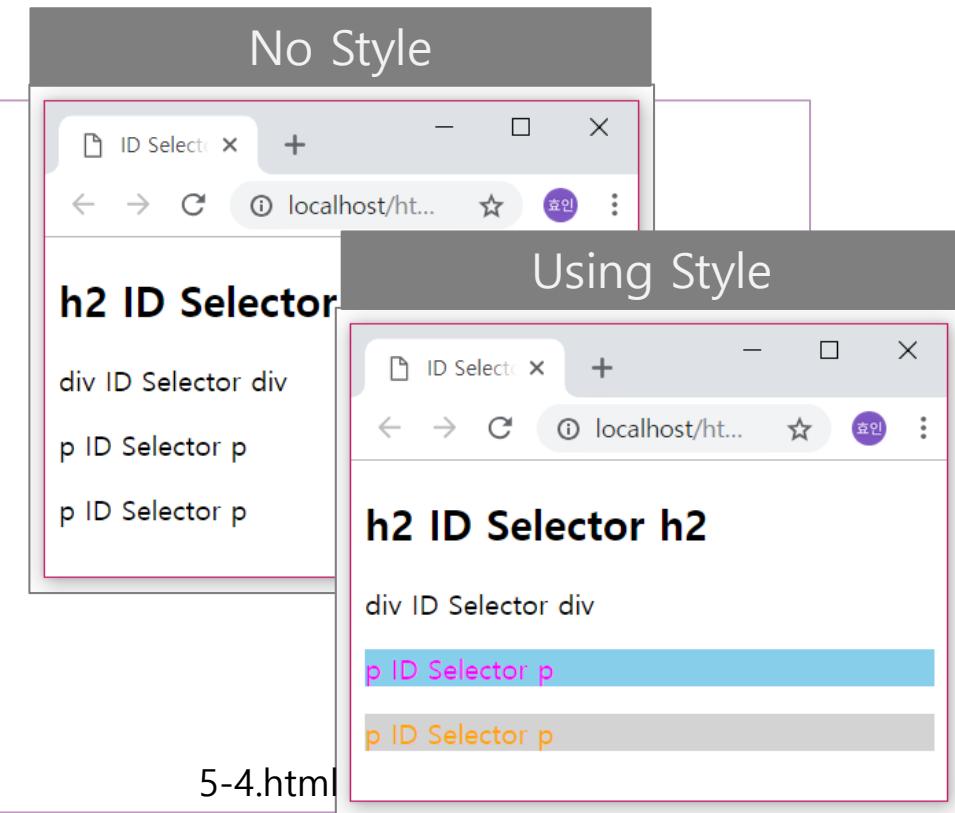


# 5. CSS 이해

## 5-1. 선택자의 이해 (5/15) – 일반 선택자 요소 (4/4)

- ✓ ID 선택자(ID Selector) 사용법은 “`#IDName { }`” 이다.
- ✓ HTML 문서에서 동일한 ID를 중복 사용할 수 없다.(Class와 달리 ID는 유일해야 함.)
- ✓ id 속성 값엔 1개의 id만 사용 가능.
- ✓ 일반 선택자 중 가장 우선순위가 높다.

```
<style type="text/css">
#target1 {
    background: skyblue; color: magenta;
}
#target2 {
    background: lightgray; color: orange;
}
</style>
</head>
<body>
    <h2>h2 ID Selector h2</h2>
    <div>div ID Selector div</div>
    <p id="target1">p ID Selector p</p>
    <p id="target2">p ID Selector p</p>
</body>
```

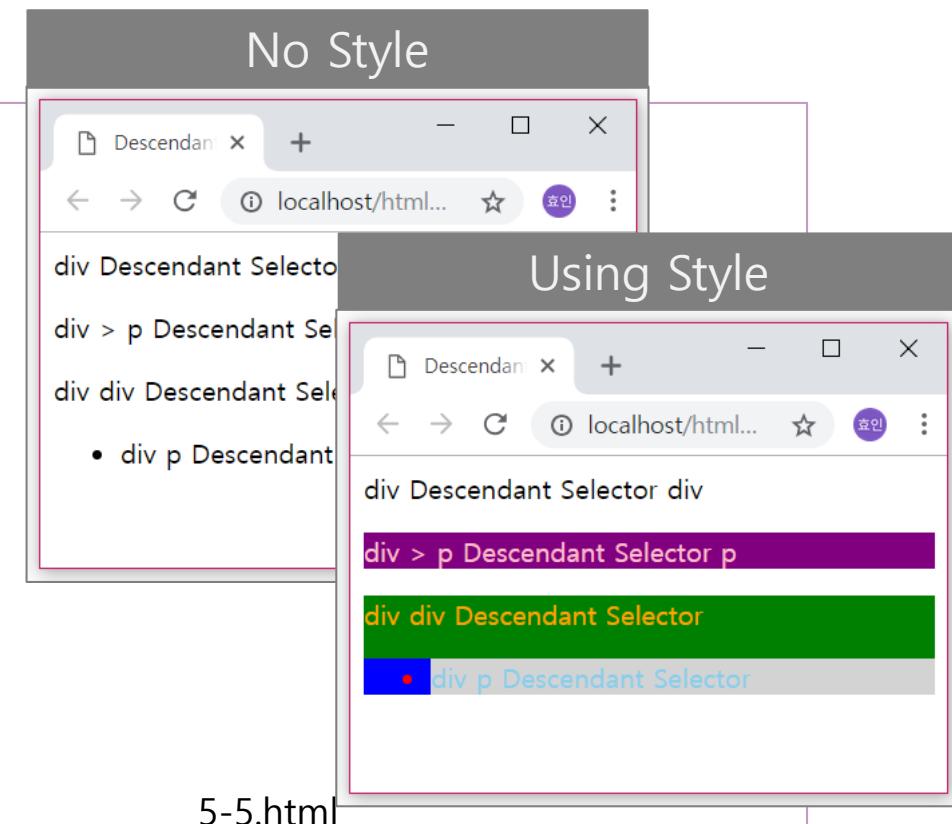


# 5. CSS 이해

## 5-1. 선택자의 이해 (6/15) – 복합 선택자 요소 (1/2)

- ✓ 하위 선택자(Descendant Selector) 사용법은 “**element element { }**” 이다.
- ✓ 하위 선택자는 1단계 하위 요소(child)와 2단계 이상 하위요소(descendant)에 모두 적용.
- ✓ 자식 선택자(Child Selector) 사용법은 “**element > element { }**” 이다.
- ✓ 자식 선택자는 1단계 하위 요소(child)에만 적용.

```
<style type="text/css">
div div { background: blue; color: red; }
div p { background: lightgray; color: skyblue; }
div > div { background: green; color: orange; }
div > p { background: purple; color: pink; }
</style>
</head>
<body>
<div>div Descendant Selector div
    <p>div > p Descendant Selector p</p>
    <div>div div Descendant Selector
        <span><div>
            <ul>
                <li><p>div p Descendant Selector</p></li>
            </ul>
        </div></span>
    </div>
</div>
</body>
```

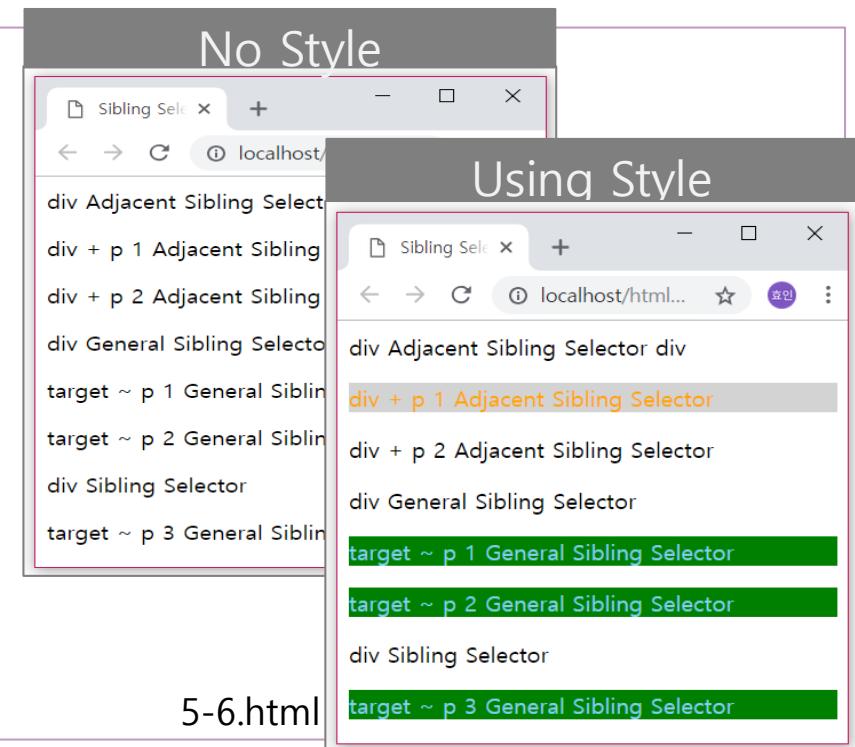


# 5. CSS 이해

## 5-1. 선택자의 이해 (7/15) – 복합 선택자 요소 (2/2)

- ✓ 인접 형제 선택자(Adjacent Sibling Selector) 사용법은 “`element + element { }`” 이다.
- ✓ 형제(sibling) 관계인 엘리먼트가 여러 개 존재할 경우 첫 번째 엘리먼트만 선택.
- ✓ 일반 형제 선택자(General Sibling Selector) 사용법은 “`element ~ element { }`” 이다.
- ✓ 형제(sibling) 관계인 엘리먼트가 여러 개 존재할 경우 모든 엘리먼트를 선택.

```
<style type="text/css">
div + p { background: lightgray; color: orange; }
div.target ~ p { background: green; color: skyblue; }
</style>
</head>
<body>
    <div>div Adjacent Sibling Selector div</div>
    <p>div + p 1 Adjacent Sibling Selector</p>
    <p>div + p 2 Adjacent Sibling Selector</p>
    <div class="target">div General Sibling Selector</div>
    <p>target ~ p 1 General Sibling Selector</p>
    <p>target ~ p 2 General Sibling Selector</p>
    <div>div Sibling Selector</div>
    <p>target ~ p 3 General Sibling Selector</p>
</body>
```





## 5. CSS 이해

### 5-1. 선택자의 이해 (8/15) – 가상 클래스 선택자 요소 (1/3)

- ✓ 가상 클래스 선택자(Pseudo-Classes Selector)는 User Agent가 제공하는 가상의 클래스를 지정.
- ✓ 사용법은 “가상 클래스 { }”이다.

선택자	의미	CSS
:link	방문하지 않은 링크를 선택.	1
:visited	방문한 링크를 선택.	1
:hover	지정된 요소에 마우스가 올라간 경우 선택.	1
:active	지정된 요소가 활성화 된 경우 선택.	1
:focus	지정된 요소가 포커스를 가질 경우 선택.	2
:first-child	지정된 요소 중 부모의 첫 번째 자식 선택.	2
:last-child	지정된 요소 중 부모의 마지막 자식 선택.	3
:nth-child(n)	지정된 요소 중 n번째 자식 선택. (n = 0, 1, ...)	3
:enabled	지정된 요소 요소가 enabled인 경우 선택.	3
:disabled	지정된 요소 요소가 disabled인 경우 선택.	3
:checked	지정된 요소 요소가 checked인 경우 선택.	3



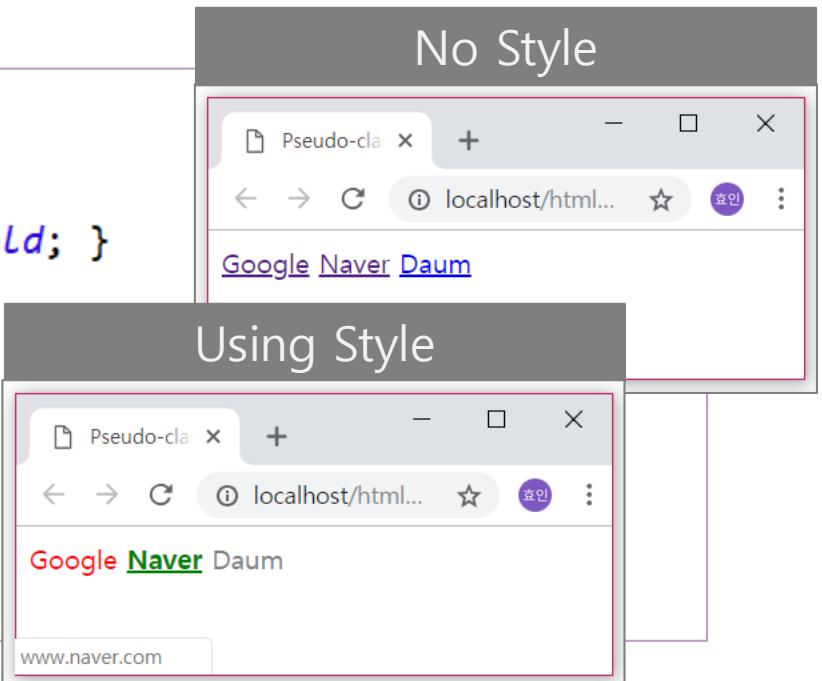
# 5. CSS 이해

## 5-1. 선택자의 이해 (9/15) – 가상 클래스 선택자 요소 (2/3)

- ✓ `:link`는 방문하지 않은 링크를 선택하고, `:visited`는 방문한 링크를 선택.
- ✓ `:hover`는 요소에 마우스가 올라간 경우 선택하고, `:active`는 요소가 활성화된 경우 선택.
- ✓ `:focus`는 요소가 포커스를 가질 경우 선택.
- ✓ `:link, :visited, :hover, :active`는 IE 6.0에서 지원하지 않고, `:focus`는 IE 6.0, 7.0에서 지원하지 않는다.

```
<style type="text/css">
a:link { color: gray; text-decoration: none; }
a:visited { color: red; text-decoration: none; }
a:hover { color: green; text-decoration: underline; font-weight: bold; }
a:active { color: blue; text-decoration: none; }
</style>
</head>
<body>
    <a href="http://www.google.com">Google</a>
    <a href="http://www.naver.com">Naver</a>
    <a href="http://www.daum.net">Daum</a>
</body>
```

5-7.html



The image shows a browser window with two tabs: "No Style" and "Using Style".  
In the "No Style" tab, the links are displayed as plain text: "Google Naver Daum".  
In the "Using Style" tab, the links are styled according to the CSS rules defined in the code. The "Naver" link is colored green and has a underline, while the other links are in their original colors (gray, red, blue).  
A red box highlights the "Using Style" tab and its content.



# 5. CSS 이해

## 5-1. 선택자의 이해 (10/15) – 가상 클래스 선택자 요소 (3/3)

- ✓ `:nth-child(n)`는 지정된 요소 부모의 n번째 자식들에 적용.
- ✓ 팔호에 상수 뿐만 아니라 수열( $2n + 1$ )도 가능.
- ✓ n은 0부터 시작하며, 자식 순번은 1부터 시작.
- ✓ 첫 번째 요소와 마지막 요소에 대한 스타일 지정은 `:first-child`와 `:last-child` 선택자를 사용.

```
<style type="text/css">
li:nth-child(5) { color: magenta; }
li:nth-child(3n+1) { color: skyblue; }
li:last-child { color: orange; }
</style>
</head>
<body>
  <ul>
    <li>첫 번째 li</li><li>두 번째 li</li>
    <li>세 번째 li</li><li>네 번째 li</li>
    <li>다섯 번째 li</li><li>여섯 번째 li</li>
    <li>일곱 번째 li</li><li>여덟 번째 li</li>
  </ul>
</body>
```

No Style

- 첫 번째 li
- 두 번째 li
- 세 번째 li
- 네 번째 li
- 다섯 번째 li
- 여섯 번째 li
- 일곱 번째 li
- 여덟 번째 li

Using Style

- 첫 번째 li
- 두 번째 li
- 세 번째 li
- 네 번째 li
- **다섯 번째 li**
- 여섯 번째 li
- 일곱 번째 li
- **여덟 번째 li**

5-8.html



# 5. CSS 이해

## 5-1. 선택자의 이해 (11/15) – 가상 엘리먼트 선택자 요소 (1/2)

- ✓ 가상 엘리먼트 선택자(Psuedo-Element Selector)는 보이지 않는 가상의 엘리먼트를 선택.
- ✓ 사용법은 “`::가상 엘리먼트 { }`” 이다.
- ✓ 가상 엘리먼트 표기법으로 CSS1과 CSS2에서 **single colon(:)**을 사용했음.
- ✓ CSS3에선 가상 클래스와 가상 엘리먼트를 구별하기 위해 **double colon(::)**으로 대체.

선택자	의미	CSS
<code>::after</code>	지정된 요소 뒤에 content 추가.	2
<code>::before</code>	지정된 요소 앞에 content 추가.	2
<code>::first-letter</code>	지정된 요소의 첫 번째 문자 선택.	1
<code>::first-line</code>	지정된 요소의 첫 번째 라인 선택.	1
<code>::selection</code>	사용자에 의해 선택된 요소의 위치 선택.	3

# 5. CSS 이해

## 5-1. 선택자의 이해 (12/15) – 가상 엘리먼트 선택자 요소 (2/2)

- ✓ ::after, ::before는 IE9.0부터 지원.(8.0은 single colon CSS2 문법만 지원)
- ✓ ::first-letter, ::first-line은 IE9.0부터 지원.(5.5~8.0은 single colon CSS2 문법만 지원)
- ✓ ::selection은 IE9.0부터 지원.
- ✓ 결과 화면에서 마우스로 텍스트를 드래그하면 ::selection 스타일이 적용.

```
<style type="text/css">
p.intro:first-letter { font-size: 200%; }
p.intro:first-line { font-weight: bold; }
p:last-child::after {
    content: "(source: www.w3schools.com)";
    color: orange; }
::selection { color: steelblue; }
</style>
</head>
<body>
    <h1>CSS Tutorial</h1>
    <h3>CSS is a language that describes the style of an HTML document.
    <p class="intro">CSS describes how HTML elements should be displayed.
    <p>This tutorial will teach you CSS from basic to advanced.
</body>

</html>
```

No Style

Using Style

CSS Tutorial

CSS is a language that describes the style of an HTML document.

CSS describes how HTML elements should be displayed.

This tutorial will teach you CSS from basic to advanced.(source: www.w3schools.com)



# 5. CSS 이해

## 5-1. 선택자의 이해 (13/15) – 속성 선택자 요소 (1/2)

- ✓ 특정한 속성을 가지거나 속성 값을 갖는 엘리먼트를 선택.
- ✓ Existence([]), Equality([=]), Space([~=]), Prefix([^=]), Substring([\*=]) 등이 있다.
- ✓ 속성 선택자를 사용하기 위해서는 HTML문서를 작성할 때에 name, title등의 속성값을 규칙적으로 정의.
- ✓ 화면에 같은 분류의 많은 항목들을 일괄적으로 선택할 때 유용. (예: 특정이름을 갖는 체크박스)

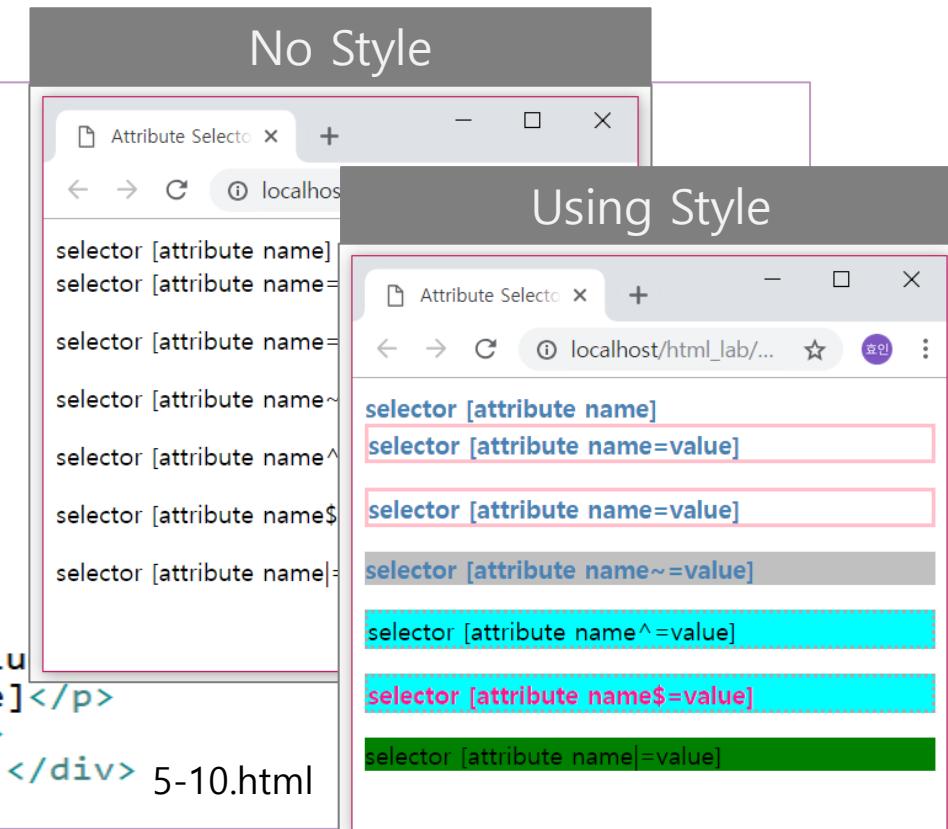
선택자	의미	CSS
[A]	A 속성이 포함된 엘리먼트 선택.	2
[A=V]	A 속성 값이 V와 정확히 일치하는 엘리먼트 선택.	2
[A~=V]	A 속성 값이 V단어(word)를 포함하는 엘리먼트 선택.	2
[A^=V]	A 속성 값이 V로 시작하는 엘리먼트 선택.	3
[A*=V]	A 속성 값이 V를 포함하는 엘리먼트 선택.	3
[A\$=V]	A 속성 값이 V로 끝나는 엘리먼트 선택.	3
[A =V]	A 속성 값이 정확히 V이거나, V-으로 시작하는 엘리먼트 선택.	2

# 5. CSS 이해

## 5-1. 선택자의 이해 (14/15) – 속성 선택자 요소 (2/2)

- ✓ [A~=V]는 속성 값이 V단어(space로 구분)를 포함하는 엘리먼트 선택.
- ✓ [A\*=V]는 속성 값이 V를 포함하는 엘리먼트를 선택.(V는 단어가 아니어도 됨)
- ✓ [A|=V]는 속성 값이 V와 정확히 일치하거나, V 뒤에 Hyphen(-)이 붙은 엘리먼트를 선택.
- ✓ [A\$=V]는 속성 값이 V로 끝나는 엘리먼트를 선택.

```
<style type="text/css">
[title] { color: steelblue; font-weight: bold; }
[title="two"] { border: 2px solid pink; }
p[title="tow"] { color: orange; }
p[title~="second"] { background: silver; }
p[class^="second"] { background: cyan; }
p[class$="wrap"] { color: deeppink; font-weight: bold; }
[class*="three"] { background: green; }
[class|= "second"] { border: 2px dotted darkgray; }
</style>
</head>
<body>
    <div title="one">selector [attribute name]</div>
    <div title="two">selector [attribute name=value]</div>
    <p title="two">selector [attribute name=value]</p>
    <p title="first second third">selector [attribute name~=value]</p>
    <p class="second-container">selector [attribute name^=value]</p>
    <p class="second-wrap">selector [attribute name$=value]</p>
    <div class="one-two-three">selector [attribute name|=value]</div>
</body>
```



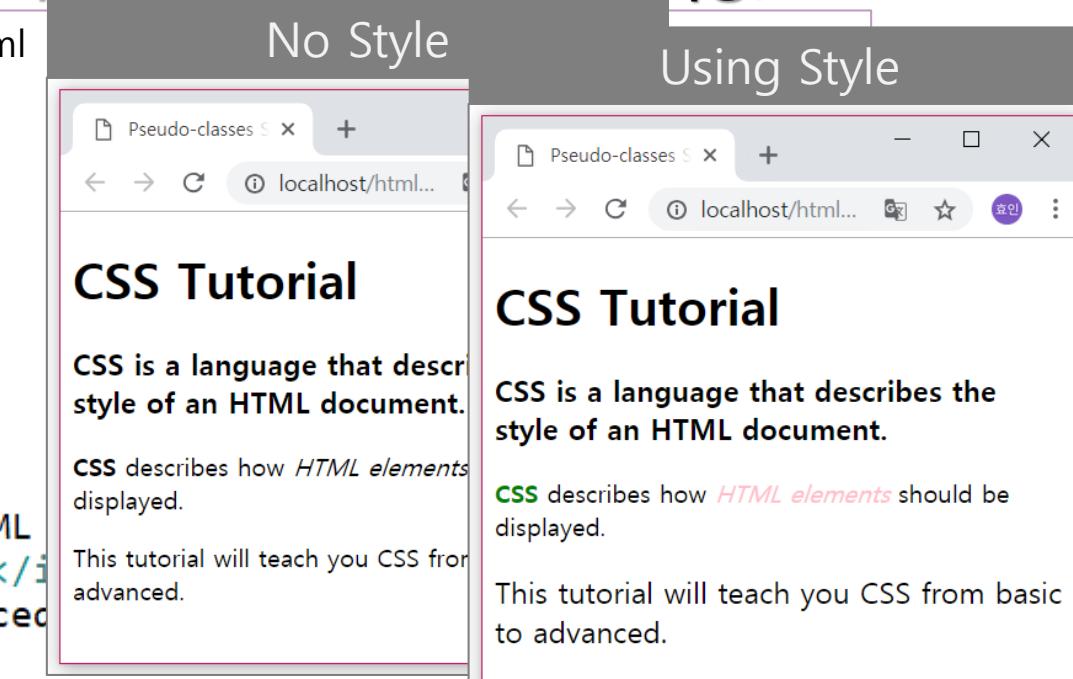
# 5. CSS 이해

## 5-1. 선택자의 이해 (15/15) – CSS 규칙 적용 우선순위

- ✓ 같은 엘리먼트에 두 개 이상의 CSS 규칙이 적용된 경우 마지막 규칙, 구체적인 규칙, !important가 우선 적용.
- ✓ CSS 규칙들 중 하단에 작성한 규칙이 마지막 규칙이다.
- ✓ P {} 보단 p b {}가 더 구체적이므로 p {}가 아닌 p b {}가 적용됨.
- ✓ 속성 값 뒤에 !important를 작성하면, 같은 엘리먼트에 대해 보다 우선적으로 스타일 적용.

```
<style type="text/css">
i { color: orange; }
i { color: pink; }
p b { color: green !important; }
p b { color: gray; }
p#intro { font-size: 100% }
p { font-size: 120%; }
</style>
</head>
<body>
    <h1>CSS Tutorial</h1>
    <h3>CSS is a language that describes the style of an HTML</h3>
    <p id="intro"><b>CSS</b> describes how <i>HTML elements</i> are displayed.</p>
    <p>This tutorial will teach you CSS from basic to advanced</p>
</body>
```

5-11.html





# 5. CSS 이해

## 5-2. Font 속성 (1/6) – 개요

- ✓ <font> tag 관련 속성은 CSS property로 대체 가능하므로 추천하지 않는 기능.
- ✓ CSS Font 관련 속성은 text의 글꼴, 굵기, 크기, 스타일 등을 지정.
- ✓ font-family, font-size, font-style, font-variant, font-weight, font로 구성.
- ✓ font는 하나의 선언에서 여러 font 관련 속성을 지정.

속성	의미	CSS
font-family	글꼴 지정 (font name).	1
font-size	글자 크기 지정.	1
font-style	글자 스타일 지정.	1
font-variant	소문자를 작은 대문자(small-caps)로 변형.	1
font-weight	글자 굵기 지정.	1
font	font에 관한 속성을 한번에 지정하는 단축형(short hand) 속성.	1

# 5. CSS 이해

## 5-2. Font 속성 (2/6) – font-family

- ✓ **font-family** 속성 사용법은 “E { font-family: 글꼴이름, 글꼴이름, ... }” 이다.
- ✓ CSS Parser는 앞의 글꼴부터 읽으며 글꼴이 사용자 PC에 없을 경우 다음 글꼴을 적용.
- ✓ generic font명을 뒤에 작성하는 것이 일반적이다.
- ✓ **font name**에 white-space가 포함된 경우 quotation으로 감싸야 한다.

```
<style type="text/css">
#serif { font-family: serif; }
#sans-serif { font-family: sans-serif; }
/*white-space가 들어간 폰트명은 quotation으로 감싼다.*/
#monospace { font-family: monospace, "Times New Roman"; }
</style>
</head>
<body>
    <h2>font-family</h2>
    <p id="serif"><b>serif</b>는 글자의 획 끝을 serif로 장식한 서체</p>
    <p id="sans-serif"><b>sans-serif</b>는 serif가 없는 글자 끝이 곧은 서체</p>
    <p id="monospace"><b>monospace</b>는 글자의 너비가 모두 동일한 서체</p>
</body>
```

The image shows two browser windows side-by-side, illustrating the effect of the CSS code on the left.

**No Style** window:

font-family

serif는 글자의 획 끝을 serif로 장식한 서체  
sans-serif는 serif가 없는 글자 끝이 곧은 서체  
monospace는 글자의 너비가 모두 동일한 서체

**Using Style** window:

font-family

serif는 글자의 획 끝을 serif로 장식한 서체  
sans-serif는 serif가 없는 글자 끝이 곧은 서체  
monospace는 글자의 너비가 모두 동일한 서체

The 'No Style' window shows the original definitions of the font families. The 'Using Style' window shows how they are applied to the corresponding HTML elements in the code, resulting in different visual styles.

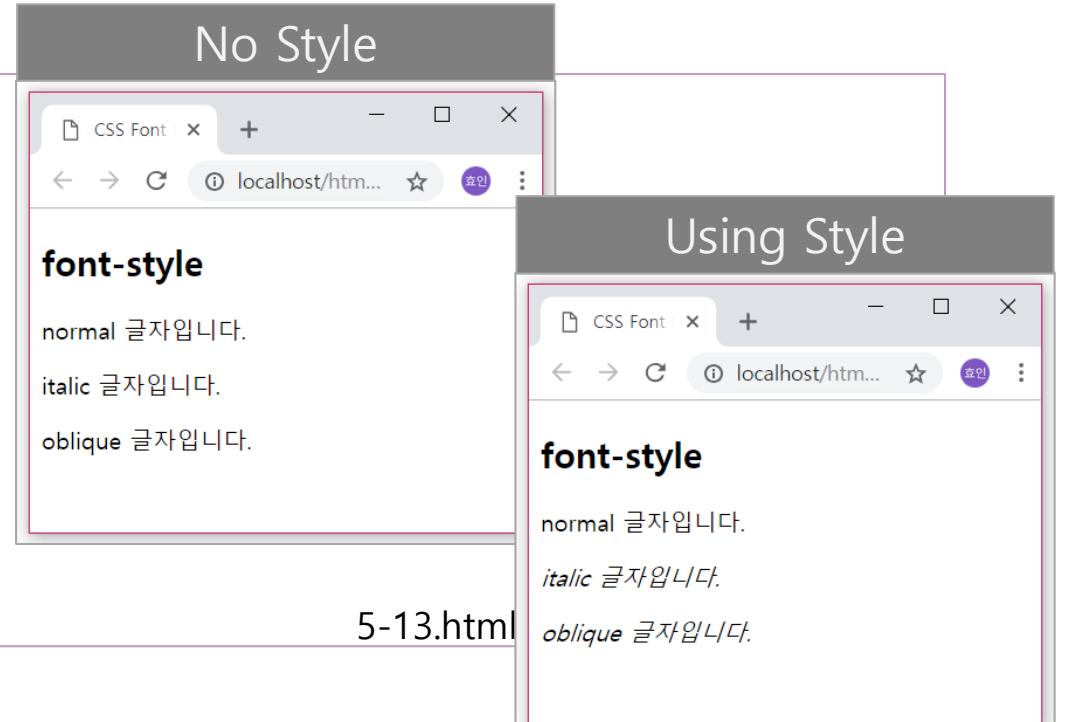


# 5. CSS 이해

## 5-2. Font 속성 (3/6) – font-style

- ✓ 글자 스타일을 지정하기 위해 사용하며 사용법은 “E { font-style: normal | italic | oblique }” 이다.
- ✓ 속성 값은 **normal**, **italic**, **oblique**가 있으며 기본값은 **normal**이다.
- ✓ **italic** 속성은 글자를 이탤릭체로 표시, **oblique** 속성은 글자를 기울임꼴로 표시하기 위해 사용.
- ✓ **italic**은 디자이너에 의해 디자인된 특수한 글꼴이고 **oblique**는 일반 글자를 8~12deg 정도 기울여서 표시.

```
<style type="text/css">
#normal { font-style: normal; }
#italic { font-style: italic; }
#oblique { font-style: oblique; }
</style>
</head>
<body>
    <h2>font-style</h2>
    <p id="normal">normal 글자입니다.</p>
    <p id="italic">italic 글자입니다.</p>
    <p id="oblique">oblique 글자입니다.</p>
</body>
```



The image shows two browser windows side-by-side. Both windows have a title bar labeled "CSS Font" and a URL bar showing "localhost/htm...".

**No Style** window content:

- font-style**
- normal 글자입니다.
- italic 글자입니다.
- oblique 글자입니다.

**Using Style** window content:

- font-style**
- normal 글자입니다.
- italic 글자입니다.
- oblique 글자입니다.

Both windows show identical text content and styling, demonstrating that the CSS styles are being applied correctly.

# 5. CSS 이해

## 5-2. Font 속성 (4/6) – font-variant

- ✓ **font-variant** 속성 사용법은 “`E { font-variant: normal | small-caps; }`” 이다.
- ✓ 초기값은 **normal**이며, 한글은 변화 없다.
- ✓ **font-weight** 속성 사용법은 “`E { font-weight: normal | bold | bolder | lighter; }`” 이다.
- ✓ 초기값은 **normal**이며, 100~900까지 숫자 값으로 사용가능.(400 : **normal**, 700 : **bold**)

```
<style type="text/css">
#variant { font-variant: small-caps; }
#weight1 { font-weight: bold; }
#weight2 { font-weight: bolder; }
#weight3 { font-weight: 500; }
#weight4 { font-weight: lighter; }
</style>
</head>
<body>
    <h2>font-variant, font-weight</h2>
    <p id="variant">font-variant 속성 적용.</p>
    <p id="weight1">font-weight h2: bold 속성 적용.</p>
    <p id="weight2">font-weight : bolder 속성 적용.</p>
    <p id="weight3">font-weight : 500 속성 적용.</p>
    <p id="weight4">font-weight : lighter 속성 적용.</p>
</body>
```

The image shows two side-by-side browser windows. The left window is titled 'No Style' and the right is titled 'Using Style'. Both windows have a header bar with back/forward buttons, a URL field showing 'localhost/htm...', and a status bar with Korean text. The main content area of both windows displays the same HTML code from the previous screenshot. In the 'No Style' window, the text is in a standard sans-serif font. In the 'Using Style' window, the text is in a serif font. The 'Using Style' window also has a red border around its content area, while the 'No Style' window does not.

No Style

Using Style

font-variant, font-weight

font-variant 속성 적용.

font-weight h2: bold 속성 적용.

font-weight : bolder 속성 적용.

font-weight : 500 속성 적용.

font-weight : lighter 속성 적용.

font-variant, font-weight

FONT-VARIANT 속성 적용.

font-weight h2: bold 속성 적용.

font-weight : bolder 속성 적용.

font-weight : 500 속성 적용.

font-weight : lighter 속성 적용.



# 5. CSS 이해

## 5-2. Font 속성 (5/6) – font-size

- ✓ **font-size** 속성 사용법은 “E { font-size: 속성 값 }” 이다.
- ✓ 절대 사이즈 속성 값은 xx-small, x-small, small, medium, large, x-large, xx-large가 있다.
- ✓ 상대 사이즈 속성 값은 larger, maller가 있다.
- ✓ 그 외 px, cm, %(부모 엘리먼트와의 비율) 단위도 사용 가능.

```
<style type="text/css">
#h1 { font-size: 150%; }
#h2 { font-size: 100%; }
p { font-size: 75%; }
#p10pt { font-size: 10pt; }
#p10px { font-size: 10px; }
</style>
</head>
<body>
    <h1>font-size : h1</h1>
    <h1 id="h1">font-size : h1 150%</h1>
    <h2>font-szie : h2</h2>
    basic font
    <h2 id="h2">font-szie : h2 100%</h2>
    <p>font-size : p 75%</p>
    <p id="p10pt">font-size : p 10pt</p>
    <p id="p10px">font-size : p 10px</p>
</body>
```

5-15.html

The image shows two side-by-side browser windows. Both windows have a title bar with a close button and are displaying the same HTML content. The left window is titled "No Style" and contains the raw HTML code. The right window is titled "Using Style" and shows the rendered content with applied CSS styles. In both windows, the text is displayed in a monospace font. The "Using Style" window shows the following rendered text:

Element	Style	Rendered Text
h1	font-size: 150%	font-size : h1 150%
h2	font-size: 100%	font-szie : h2 100%
p	font-size: 75%	font-size : p 75%
p	font-size: 10pt	font-size : p 10pt
p	font-size: 10px	font-size : p 10px

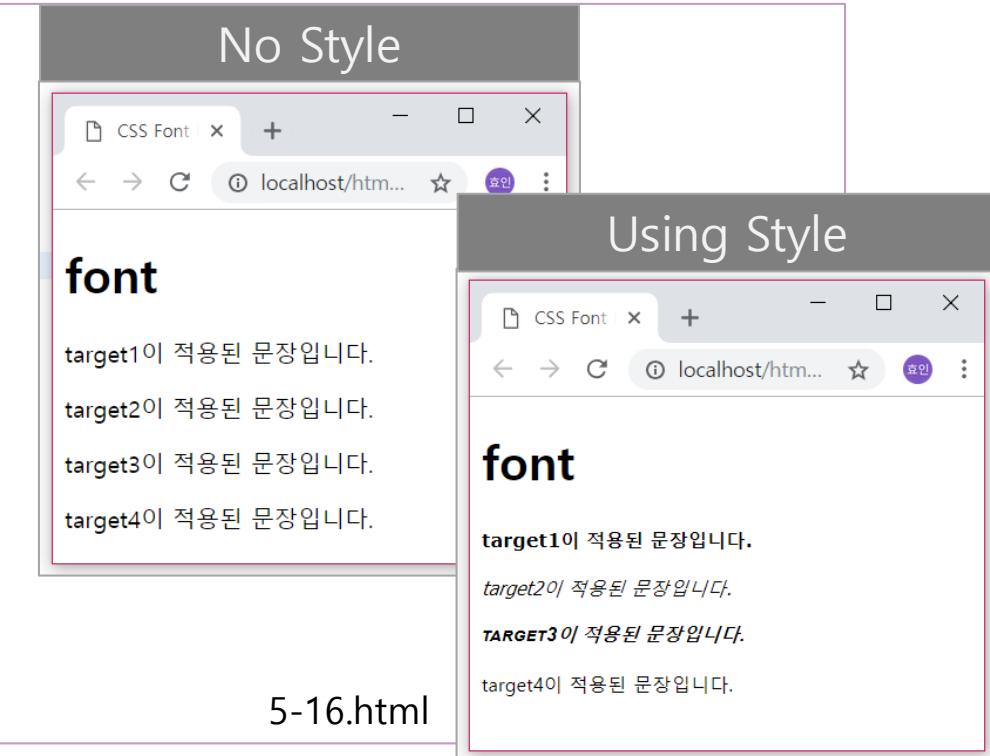


# 5. CSS 이해

## 5-2. Font 속성 (6/6) - font

- ✓ **font** 속성은 **font-style**, **font-variant**, **font-weight**, **font-size**등을 한번에 지정 가능한 단축형.
- ✓ 속성 값은 “**font-style** **font-variant** **font-weight** **font-size/line-height** **font-family**”순으로 작성.
- ✓ **font-size**와 **font-family**는 필수 값이며, 생략 시 기본 값이 적용.
- ✓ 여러 속성을 정의할 때 순서에 맞지 않게 정의하면 일부만 적용되거나 전체가 무시 될 수도 있다.

```
<style type="text/css">
/* 글씨 크기가 132px이고 행간격이 1.7em인 bold체 Verdana 글꼴 */
#target1 { font: bold 13px/1.7em Verdana; }
/* 글씨 크기가 13px인 italic체 Tahoma 글꼴 */
#target2 { font: italic 13px Tahoma; }
/* Full option */
#target3 { font: italic small-caps bold 13px/1.7em Arial; }
/* bold가 순서에 맞지 않아서 이후의 sans-serif 글꼴 무시 */
#target4 { font: 13px/1.5em bold sans-serif; }
</style>
</head>
<body>
    <h1>font</h1>
    <p id="target1">target1이 적용된 문장입니다.</p>
    <p id="target2">target2이 적용된 문장입니다.</p>
    <p id="target3">target3이 적용된 문장입니다.</p>
    <p id="target4">target4이 적용된 문장입니다.</p>
</body>
```



The image shows two side-by-side browser windows. The left window, titled 'No Style', displays the text 'font' in a standard sans-serif font. The right window, titled 'Using Style', displays the same text in a bold, italicized, small-caps font. This visual comparison illustrates how the CSS rules defined in the code affect the presentation of the text.

No Style

Using Style

font

target1이 적용된 문장입니다.

target2이 적용된 문장입니다.

target3이 적용된 문장입니다.

target4이 적용된 문장입니다.

font

target1이 적용된 문장입니다.

target2이 적용된 문장입니다.

TARGET3이 적용된 문장입니다.

target4이 적용된 문장입니다.



## 5. CSS 이해

### 5-3. Text 속성 (1/8) - 개요

- ✓ CSS Text 관련 속성은 글자, 공간, 단어, 문단들이 보여지는 속성을 정의.
- ✓ **text-align, text-decoration, text-indent, text-transform, white-space**등으로 구성.
- ✓ 들여쓰기를 위해 &nbsp;문자를 사용하는 것이 아니라 **text-indent** 속성을 사용하여 들여쓰기를 적용.

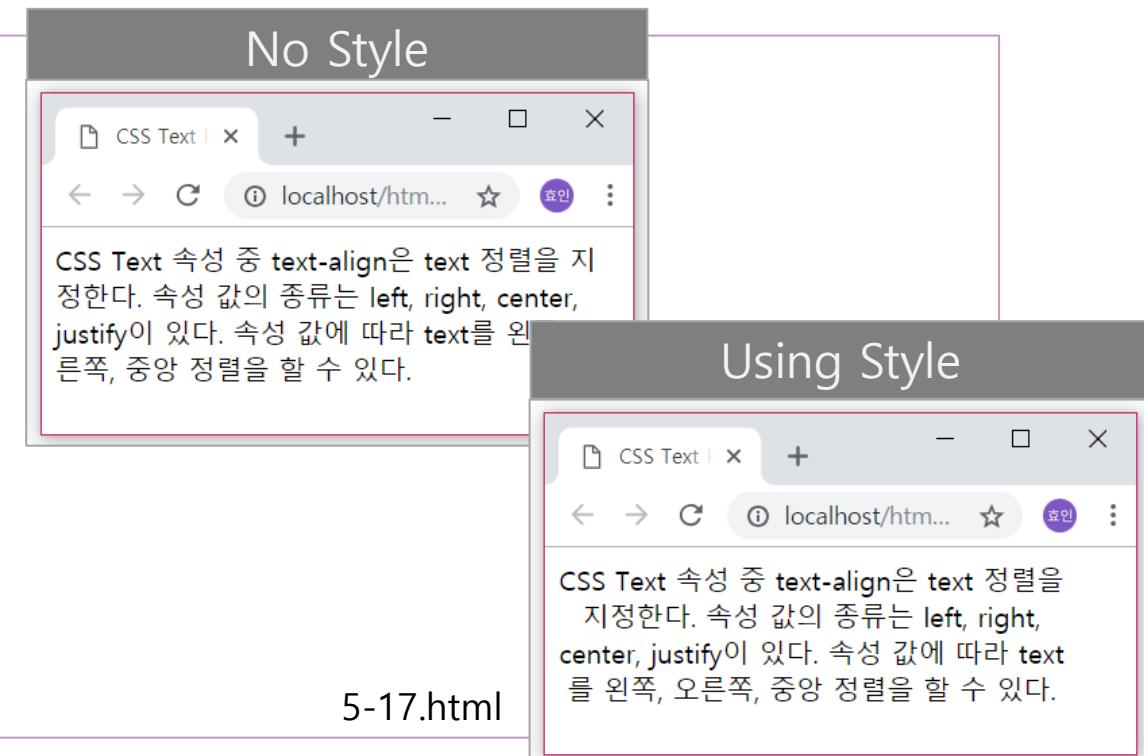
속성	의미	CSS
text-align	text 정렬 방식 지정	1
test-decoration	text 장식 지정.	1
text-ident	Text-block안 첫 라인의 들여쓰기 지정.	1
text-transform	text 대문자화.	1
white-space	엘리먼트 안의 공백 지정.	1
vertical-align	수직 정렬 지정.	1
letter-spacing	문자 간의 space 간격을 줄이거나 늘림.	1
word-spacing	단어 간의 간격 지정.	1
line-height	줄(행) 간격 지정.	1
color	text 색상 지정.	1

# 5. CSS 이해

## 5-3. Text 속성 (2/8) – text-align

- ✓ **text-align** 속성 사용법은 “E { text-align: left | right | center | justify }” 이다.
- ✓ **justify**는 각 라인의 너비가 모두 동일 하도록 간격을 늘린다.
- ✓ 과거에는 <center> tag를 사용하여 가운데 정렬을 적용 했지만 HTML5부터는 **text-align** 속성을 사용하도록 함.

```
<style type="text/css">
body { width: 300px; }
/* div { text-align: lef; }
div { text-align: right; }
div { text-align: justify; } */
div { text-align: center; }
</style>
</head>
<body>
  <div>
    CSS Text 속성 중 text-align은 text 정렬을 지정한다.
    속성 값의 종류는 left, right, center, justify이 있다.
    속성 값에 따라 text를 왼쪽, 오른쪽, 중앙 정렬을 할 수 있다.
  </div>
</body>
```



# 5. CSS 이해

## 5-3. Text 속성 (3/8) – text-decoration

- ✓ **text-decoration** 속성 사용법은 “E { **text-decoration: none | underline | overline | line-through | blink** }” 이다.
- ✓ **line-through**는 text 사이로 라인생성.
- ✓ **blink**는 text가 깜빡이는 효과를 적용.(IE 지원 X)

```
<style type="text/css">
#none { text-decoration: none; }
#underline { text-decoration: underline; }
#overline { text-decoration: overline; }
#line-through { text-decoration: line-through; }
#blink { text-decoration: blink; }
</style>
</head>
<body>
    <div id="none">default(none) 속성이 적용된 문자열입니다.</div>
    <div id="underline">underline 속성이 적용된 문자열입니다.</div>
    <div id="overline">overline 속성이 적용된 문자열입니다.</div>
    <div id="Line-through">line-through 속성이 적용된 문자열입니다.</div>
    <div id="blink">blink 속성이 적용된 문자열입니다.</div>
</body>
```

The image shows two browser windows side-by-side, both displaying the same HTML content. The left window is titled 'No Style' and the right is titled 'Using Style'. Both windows show the same five 

elements with different text-decoration values. The 'Using Style' window highlights the text in each 

element to demonstrate the effect of each decoration type.

No Style

Using Style

default(**none**) 속성이 적용된 문자열입니다.  
underline 속성이 적용된 문자열입니다.  
overline 속성이 적용된 문자열입니다.  
~~line-through~~ 속성이 적용된 문자열입니다.  
blink 속성이 적용된 문자열입니다.

default(**none**) 속성이 적용된 문자열입니다.  
underline 속성이 적용된 문자열입니다.  
overline 속성이 적용된 문자열입니다.  
~~line-through~~ 속성이 적용된 문자열입니다.  
blink 속성이 적용된 문자열입니다.

5-18.html

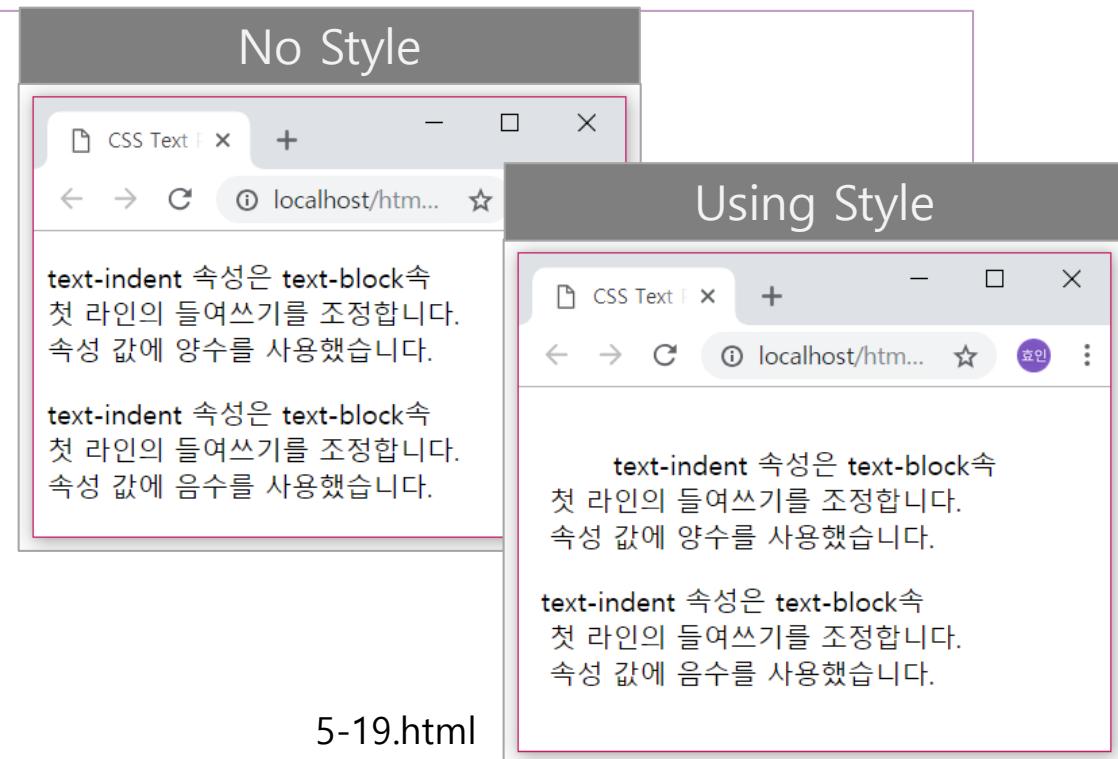


# 5. CSS 이해

## 5-3. Text 속성 (4/8) – text-indent

- ✓ **text-indent 속성 사용법은 “E { text-indent: 절대 값(px,pt,em,em etc) | 배율(%) }”** 이다.
- ✓ 절대 값의 기본 값은 0이다.
- ✓ 배율 값은 부모 엘리먼트 너비의 비율로 들여쓰기 한다.
- ✓ 속성 값에 음수를 허용함.(음수 값 사용 시 왼쪽으로 들여쓰기 함.)

```
<style type="text/css">
body { padding: 10px; }
p.positive { text-indent: 1cm; }
p.negative { text-indent: -5px; }
</style>
</head>
<body>
  <p class="positive">
    text-indent 속성은 text-block속<br>
    첫 라인의 들여쓰기를 조정합니다.<br>속성 값에 양수를 사용했습니다.
  </p>
  <p class="negative">
    text-indent 속성은 text-block속<br>
    첫 라인의 들여쓰기를 조정합니다.<br>속성 값에 음수를 사용했습니다.
  </p>
</body>
```



No Style

Using Style

text-indent 속성은 text-block속  
첫 라인의 들여쓰기를 조정합니다.  
속성 값에 양수를 사용했습니다.

text-indent 속성은 text-block속  
첫 라인의 들여쓰기를 조정합니다.  
속성 값에 음수를 사용했습니다.

text-indent 속성은 text-block속  
첫 라인의 들여쓰기를 조정합니다.  
속성 값에 양수를 사용했습니다.

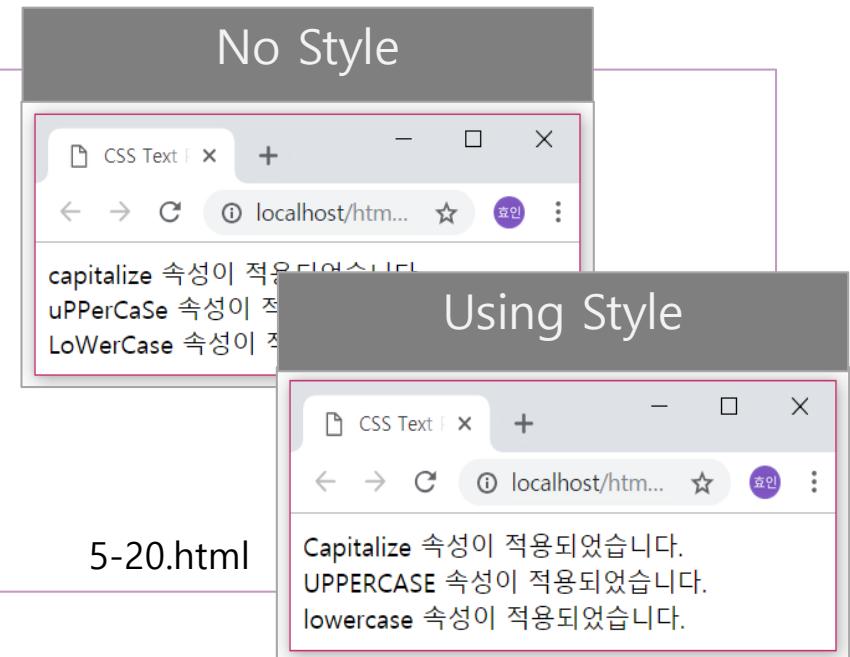
text-indent 속성은 text-block속  
첫 라인의 들여쓰기를 조정합니다.  
속성 값에 음수를 사용했습니다.

# 5. CSS 이해

## 5-3. Text 속성 (5/8) – text-transform

- ✓ **text-transform** 속성 사용법은 “E { text-transform: **capitalize | uppercase | lowercase | none** }”이다.
- ✓ **capitalize**는 첫 글자를 대문자로, **uppercase**는 글자 전체를 대문자로, **lowercase**는 글자 전체를 소문자로 변경.

```
<style type="text/css">
.cap { text-transform: capitalize; }
.up { text-transform: uppercase; }
.Low { text-transform: lowercase; }
</style>
</head>
<body>
  <div class="cap">capitalize 속성이 적용되었습니다.</div>
  <div class="up">uPPERCaSe 속성이 적용되었습니다.</div>
  <div class="Low">LoWERCaSe 속성이 적용되었습니다.</div>
</body>
```





## 5. CSS 이해

### 5-3. Text 속성 (6/8) – white-space

- ✓ **white-space 속성 사용법은 “E { white-space: normal | pre | nowrap | pre-line | pre-wrap }”** 이다.
- ✓ **normal은 정해진 영역에 따라 줄이 바뀌며, 하나의 whitespace만 허용.**
- ✓ **pre는 <pre>처럼 사용자가 입력한 모습 그대로 공백을 화면에 출력.**
- ✓ **nowrap은 하나의 whitespace만 허용하며, 줄 바꿈을 금지.(<br>을 만나기 전까진 같은 줄에 출력).**

```
<style type="text/css">
body { border: 1px solid black; }
.white1 { white-space: normal; }
.white2 { white-space: pre; }
.white3 { white-space: nowrap; }
</style>
</head>
<body>
  <p class="white1"
      onmouseover="this.className='white2'"
      onmousedown="this.className='white3'"
      onmouseout="this.className='white1'">
    basic : white1, mouseover : white2,
    mousedown : white3, mouseout : white1,
    white-space 속성 : normal, pre, nowrap,
    pre-line, pre-wrap
  </p>
</body>
```

No Style

basic : white1, mouseover : white2,  
mousedown : white3, mouseout : white1,  
white-space 속성 : normal, pre, nowrap,  
pre-line, pre-wrap

Using Style

basic : white1, mouseover : white2,  
mousedown : white3, mouseout : white1,  
white-space 속성 : normal, pre, nowrap,  
pre-line, pre-wrap

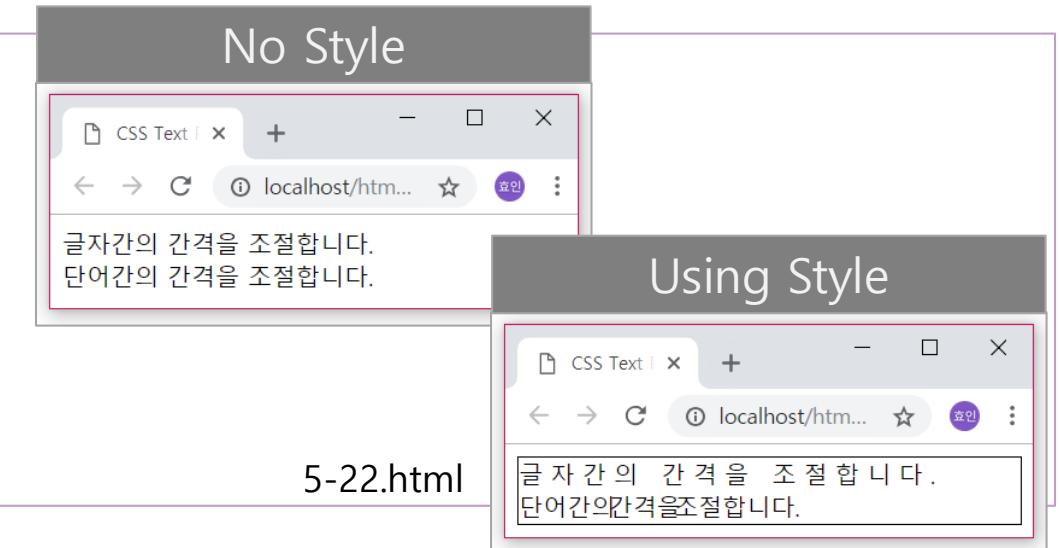


# 5. CSS 이해

## 5-3. Text 속성 (7/8) – letter-spacing

- ✓ **letter-spacing** 속성 사용법은 “E { letter-spacing: normal | 길이 값(length) }” 이다.
- ✓ 글자간의 간격을 조절.(자간을 늘릴 땐 양수, 줄일 땐 음수를 사용.)
- ✓ **word-spacing** 속성 사용법은 “E { word-spacing: normal | 길이 값(length) }” 이다.
- ✓ 단어간의 간격을 조절.

```
<style type="text/css">
body { border: 1px solid black; }
.letter { letter-spacing: 5px; }
.word { word-spacing: -10px; }
</style>
</head>
<body>
    <div class="Letter">글자간의 간격을 조절합니다.</div>
    <div class="word">단어간의 간격을 조절합니다.</div>
</body>
```



5-22.html



# 5. CSS 이해

## 5-3. Text 속성 (8/8) – line-height

- ✓ **line-height** 속성 사용법은 “`E { line-height: 상대 값 | 절대 값 | 비율 }`”이며 기본값은 `normal`이다.
- ✓ 비율 값을 사용하면 현재 글자 크기를 기본으로 50%, 200%와 같은 비율로 행 간격을 설정.
- ✓ 절대 값을 사용하면 행 간격을 px, cm 등으로 직접 설정.
- ✓ 상대 값을 사용하면 현재 글자 크기에 상대 값을 곱한 수 만큼 행 간격을 설정.

```
<style type="text/css">
p { line-height: normal; color: steelblue; }
.small { line-height: 0.7; color: orange; }
.big { line-height: 2; color: magenta; }
</style>
</head>
<body>
  <p>paragraph element입니다.<br>행 간격은 표준입니다.</p>
  <p class="small">paragraph element입니다.<br>행 간격을 좁게(0.7) 설정합니다.</p>
  <p class="big">paragraph element입니다.<br>행 간격을 넓게(2) 설정합니다.</p>
</body>
```

No Style

paragraph element입니다.  
행 간격은 표준입니다.

5-23.html

Using Style

paragraph element입니다.  
행 간격은 표준입니다.

paragraph element입니다.  
행 간격을 좁게(0.7) 설정합니다.

paragraph element입니다.  
행 간격을 넓게(2) 설정합니다.



# 5. CSS 이해

## 5-4. 사용자 인터페이스 속성 (1/5) - 개요

- ✓ 화면에 출력될 엘리먼트들에 디자인 요소를 추가하는 속성.
- ✓ 커서의 모양이나 리스트 형태를 변경.
- ✓ 문서의 배경색과 배경 이미지를 변경.
- ✓ 엘리먼트가 화면에 출력되는 방식을 조정.

속성	의미	CSS
cursor	사용자 환경의 마우스 모양을 변경.	2
classification	리스트의 글머리 기호를 변경.	1
display	엘리먼트가 화면에 출력되는 방식을 조정.	1
background-color	배경색을 지정.	1
background-image	배경을 이미지로 지정.	1
background-attachment	배경 이미지를 고정하거나 scroll여부를 지정.	1
Background-repeat	배경 그림의 반복 여부를 지정.	1
Background-position	배경 그림의 위치를 지정.	1
background	배경 관련 속성을 한번에 지정.(font 속성과 달리 속성 값 순서에 구애 받지 않음.)	1

# 5. CSS 이해

## 5-4. 사용자 인터페이스 속성 (2/5) - display

- ✓ 화면에 나오는 엘리먼트는 **Inline**과 **Block** 두 가지로 분류된다.
- ✓ **Inline-Level** 엘리먼트는 줄 바꿈 없이 연속으로 이어지며 **<span>**이 대표적이다.
- ✓ **Block-Level** 엘리먼트는 줄 바꿈이 생기며 **<div>**가 대표적이다.
- ✓ **display 속성 사용법은 “E { display: none | block | inline | ... }”** 이다.

```
<style type="text/css">
span { background-color: lightgray; }
div { background-color: tomato; }
div.target1 { display: inline; }
span.target2 { display: block; }
div.last { display: none; }
</style>
</head>
<body>
    <span>span : Inline-Level.</span>
    <div class="target1">div(target1) : Block >> Inline Change.</div>
    <span>span : Inline-Level.</span>
    <span class="target2">span(target2) : Inline >> Block Change.</span>
    <div class="Last">Last Element(display : none)</div>
</body>
```

The image shows two side-by-side browser developer tool windows, both titled "CSS Display".

**No Style** window content:

```
span : Inline-Level.
div(target1) : Block >> Inline Change.
span : Inline-Level. span(target2) : In
Block Change.
Last Element(display : none)
```

**Using Style** window content:

```
span : Inline-Level. div(target1) : Block >>
Inline Change. span : Inline-Level.
span(target2) : Inline >> Block Change.
```

The "Using Style" window highlights the "display: none" rule in red, indicating it is being applied to the last element.

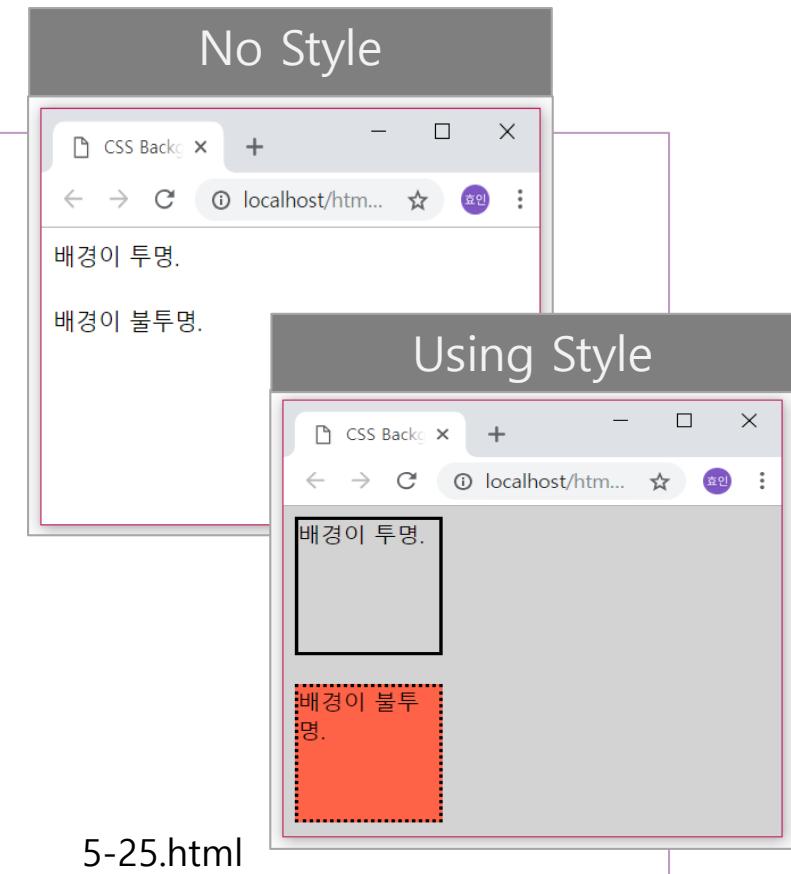


## 5. CSS 이해

### 5-4. 사용자 인터페이스 속성 (3/5) – background-color

- ✓ **background-color** 속성 사용법은 “E { background-color: color | transparent }” 이다.
- ✓ 엘리먼트의 padding, border와 같은 여백 없이 전체 크기만큼 배경색이 지정.(margin 여백은 제외)
- ✓ **transparent color**는 투명색으로 상위 엘리먼트의 배경색이 표현.
- ✓ **color**는 색 이름 | RGB 값 | RGB 코드를 사용.

```
<style type="text/css">
body { background-color: lightgray; }
.transparent {
    border: 2px solid black; background-color: transparent;
    width: 100px; height: 100px;
}
.notTransparent {
    border: 2px dotted black; background-color: tomato;
    width: 100px; height: 100px;
}
</style>
</head>
<body>
    <div class="transparent">배경이 투명.</div><br>
    <div class="notTransparent">배경이 불투명.</div>
</body>
```



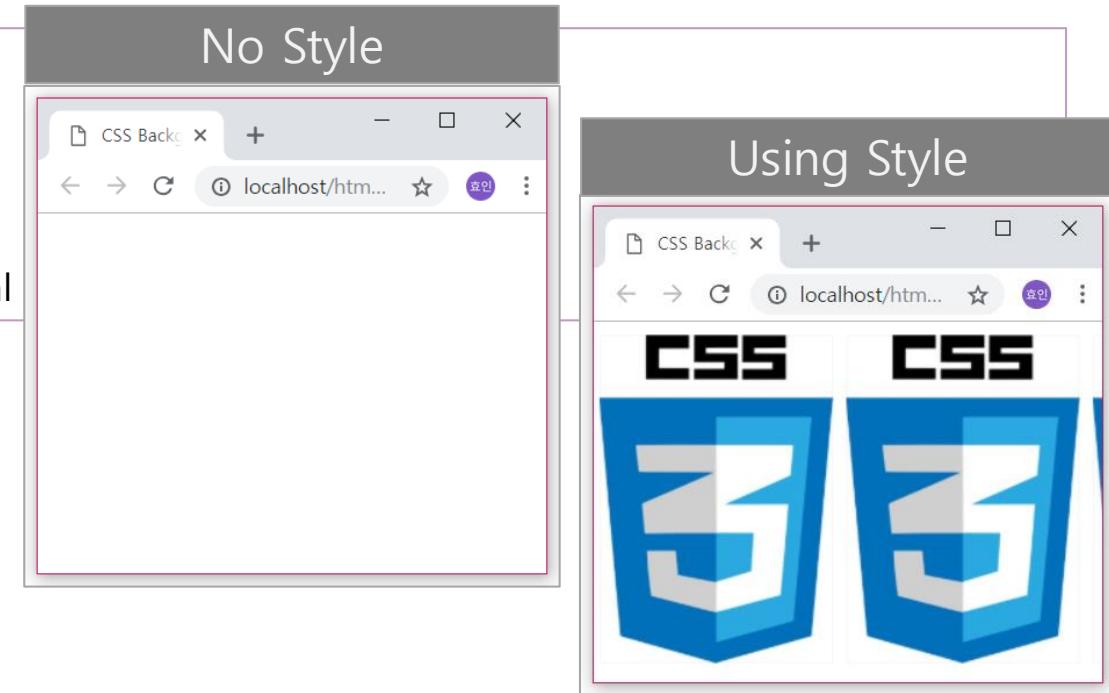
# 5. CSS 이해

## 5-4. 사용자 인터페이스 속성 (4/5) – background-image

- ✓ **background-image** 속성 사용법은 “E { background-image: none | url(“그림파일경로”) }” 이다.
- ✓ 배경 이미지가 격자모양으로 반복하여 나타남.
- ✓ 스크롤 할 경우 배경 이미지도 반복적으로 나타나며 스크롤됨.

```
<style type="text/css">  
body { background-image: url("../img/css3.png"); }  
</style>  
</head>  
<body>  
</body>
```

5-26.html



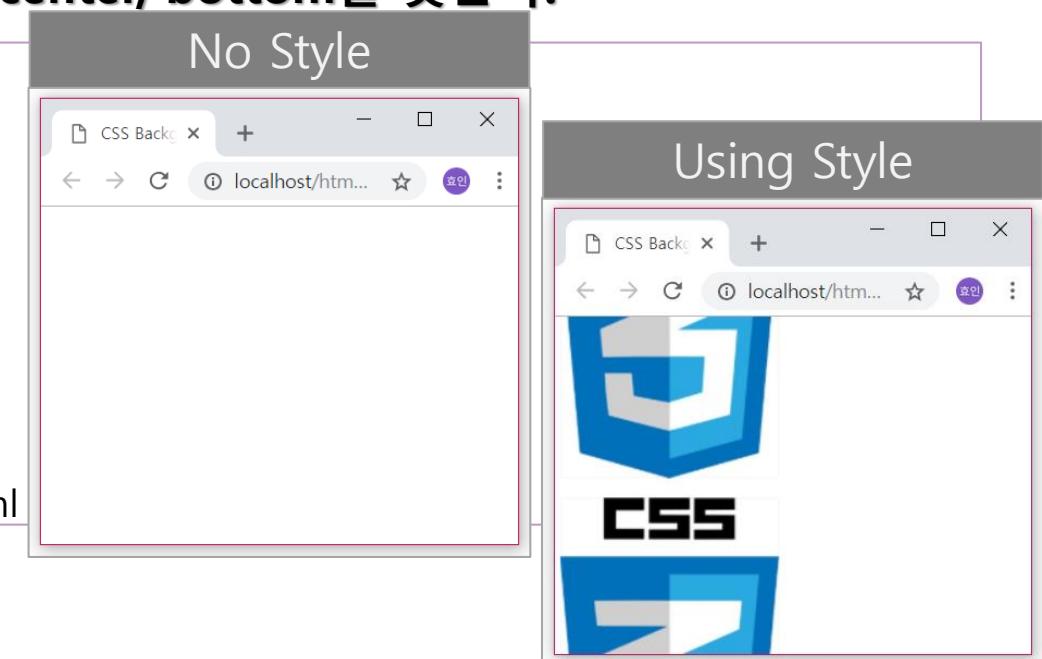
# 5. CSS 이해

## 5-4. 사용자 인터페이스 속성 (5/5) – background-repeat

- ✓ **background-repeat** 속성 사용법은 “E { background-repeat: repeat-x | repeat-y | no-repeat }” 이다.
- ✓ **repeat-x**는 배경을 수평으로 반복, **repeat-y**는 배경을 수직으로 반복.
- ✓ **background-position** 속성 사용법은 “E { background-position: 백분율(%) | 길이 값 | 수평 값 수직 값 }” 이다.
- ✓ 수평값은 **left, center, right**을 가지며 수직 값은 **top, center, bottom**을 갖는다.

```
<style type="text/css">
body {
    background-image: url("../img/css3.png");
    background-repeat: repeat-y;
    background-position: left center;
}
</style>
</head>
<body>
</body>
```

5-27.html





## 5. CSS 이해

### 5-5. 테이블 & 테두리 속성 (1/7) - 개요

- ✓ <table> 엘리먼트의 관련 속성은 테이블의 너비나 높이를 지정하고, Cell 내부 내용을 정렬.
- ✓ 관련 속성은 **table-layout**, **width**, **height**, **text-align**, **vertical-align** 등이 있다.
- ✓ 테두리 관련 속성은 테두리 모델을 설정하여, 테두리 스타일과 너비 등을 지정.
- ✓ 관련 속성은 **border-collapse**, **border-style**, **border-width**, **border-color** 등이 있다.

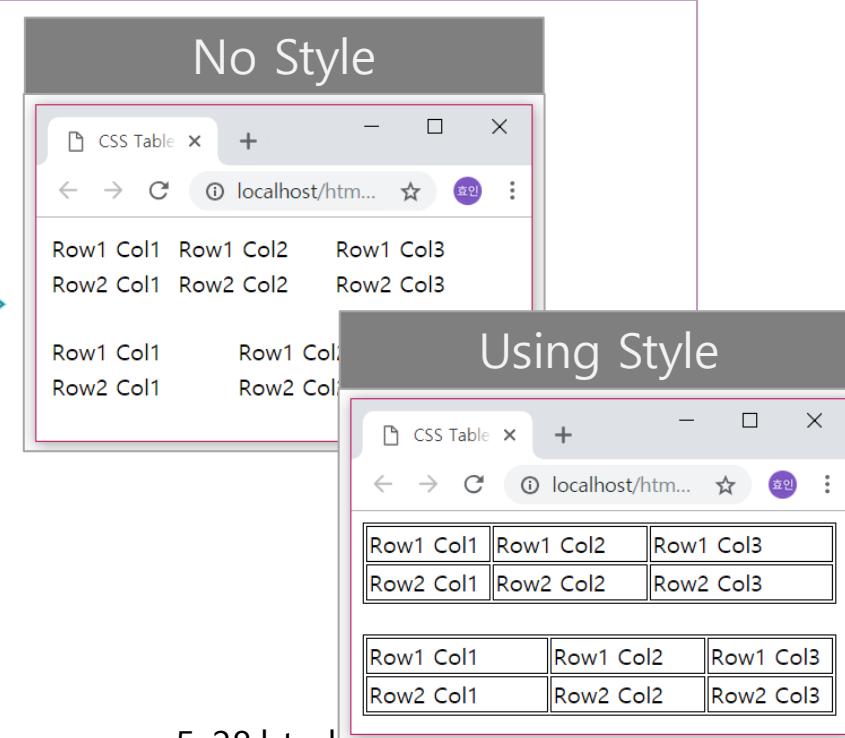
속성	의미	CSS
table-layout	table layout을 설정.	2
width	Table의 너비를 지정.	1
height	Table의 높이를 지정.	1
text-align	Cell 내부 내용을 수평 정렬.	1
vertical-align	Cell 내부 내용을 수직 정렬.	1
border-collapse	분리된 테두리모델, 통합된 테두리모델을 설정.	2
border-style	테두리 스타일을 설정.	1
border-width	엘리먼트의 4개 테두리 너비를 설정.	1
border-color	엘리먼트의 테두리 색을 설정.	1
border	테두리 관련 속성을 한번에 지정하는 단축형 속성.	1

# 5. CSS 이해

## 5-5. 테이블 & 테두리 속성 (2/7) – table-layout

- ✓ **table-layout 속성 사용법은 “E { table-layout: auto(default) | fixed }” 이다.**
- ✓ **table cell의 width, height 고정 여부를 지정.**
- ✓ **<table> tag의 첫 번째 <td>에 지정하거나, <colgroup>과 <col> 요소를 통해 table-layout 지정.**
- ✓ **cell의 width를 지정하지 않은 경우는 데이터가 cell을 넘치는 경우 사용.**

```
<style type="text/css">
table { table-layout: fixed; border: 1px solid black; }
tr, td { border: 1px solid black; }
</style>
</head>
<body>
<table>
    <tr><td width="150">Row1 Col1</td><td width="200">Row1 Col2</td>
        <td width="250">Row1 Col3</td></tr>
    <tr><td>Row2 Col1</td><td>Row2 Col2</td><td>Row2 Col3</td></tr>
</table><br>
<table>
    <colgroup>
        <col width="250"><col width="200"><col width="150">
    </colgroup>
    <tr><td>Row1 Col1</td><td>Row1 Col2</td><td>Row1 Col3</td></tr>
    <tr><td>Row2 Col1</td><td>Row2 Col2</td><td>Row2 Col3</td></tr>
</table>
</body>
```





## 5. CSS 이해

### 5-5. 테이블 & 테두리 속성 (3/7) – border-collapse

- ✓ 테두리 모델은 분리된 테두리 모델과 통합된 테두리 모델로 분류.
- ✓ **border-collapse** 속성 사용법은 “**E { border-collapse: separate(default) | collapse }**” 이다.
- ✓ **cell**은 개별 테두리를 갖으며, **border-spacing**은 분리된 테두리 모델에서 인접 **cell** 사이의 거리를 지정.
- ✓ 사용법은 “**E { border-spacing: 수평길이 수직길이 }**”이며, 길이 값은 양수를 사용.

```
<style type="text/css">
table { table-layout: fixed; border: 1px solid black; border-collapse: collapse; }
tr, td { border: 1px solid black; }
</style>
</head>
<body>
<table>
  <tr><td width="150">Row1 Col1</td><td width="200">Row1 Col2</td>
    <td width="250">Row1 Col3</td></tr>
  <tr><td>Row2 Col1</td><td>Row2 Col2</td><td>Row2 Col3</td></tr>
</table><br>
<table>
  <colgroup>
    <col width="250"><col width="200"><col width="150">
  </colgroup>
  <tr><td>Row1 Col1</td><td>Row1 Col2</td><td>Row1 Col3</td></tr>
  <tr><td>Row2 Col1</td><td>Row2 Col2</td><td>Row2 Col3</td></tr>
</table>
</body>
```

The image displays two side-by-side browser windows. Both windows show the same HTML code for a table with three columns and two rows. The top window is titled 'No Style' and shows the table with separate borders around each cell. The bottom window is titled 'Using Style' and shows the table with collapsed borders, resulting in a single continuous border for each row. This visual comparison demonstrates the effect of the border-collapse property.

No Style

Row1 Col1	Row1 Col2	Row1 Col3
Row2 Col1	Row2 Col2	Row2 Col3

Using Style

Row1 Col1	Row1 Col2	Row1 Col3
Row2 Col1	Row2 Col2	Row2 Col3

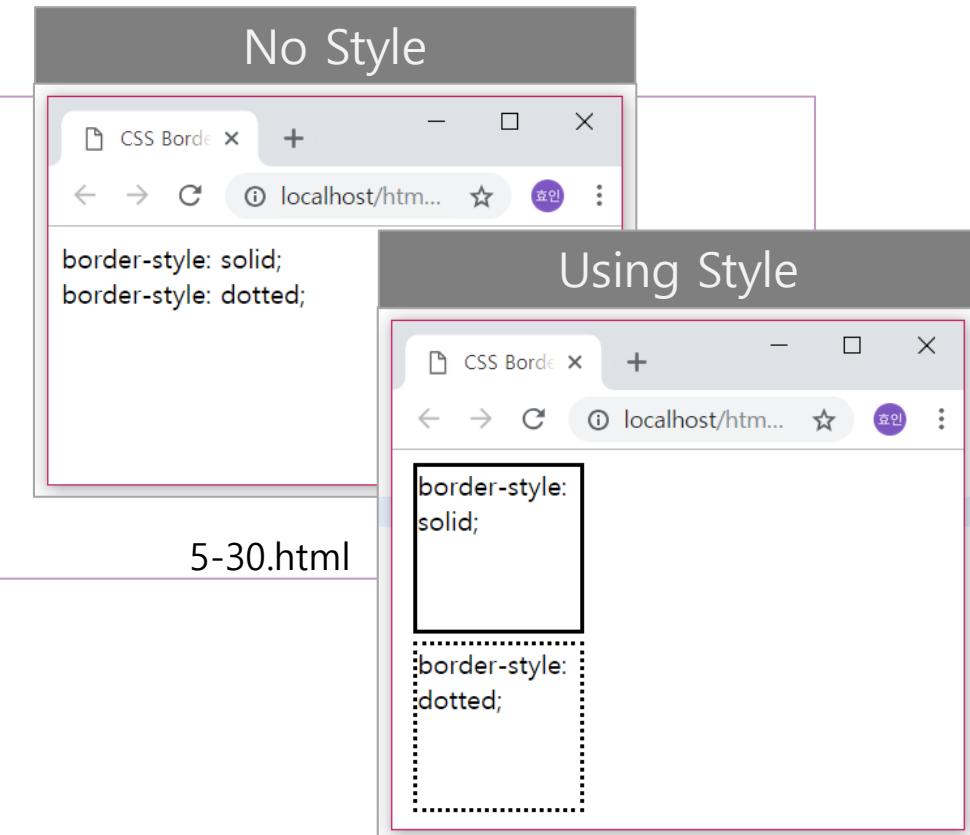
5-29.html

# 5. CSS 이해

## 5-5. 테이블 & 테두리 속성 (4/7) – border-style

- ✓ **border-style** 속성 사용법은 “E { border-style: none | solid | hidden | ... }” 이다.
- ✓ 값이 1개일 때 모든 면에 적용, 값이 2개일 때 1번째는 {top, bottom}에 2번째는 {right, left}에 적용.
- ✓ 값이 3개일 때 1번째 값은 {top}에, 2번째 값은 {right, left}에, 3번째 값은 {bottom}에 적용.
- ✓ 값이 4개일 때는 {top, right, bottom, left} 순으로 적용.

```
<style type="text/css">
div { width: 150px; height: 150px; margin: 5px; }
div.target1 { border-style: solid; }
div.target2 { border-style: dotted; }
</style>
</head>
<body>
  <div class="target1">border-style: solid;</div>
  <div class="target2">border-style: dotted;</div>
</body>
```



5-30.html

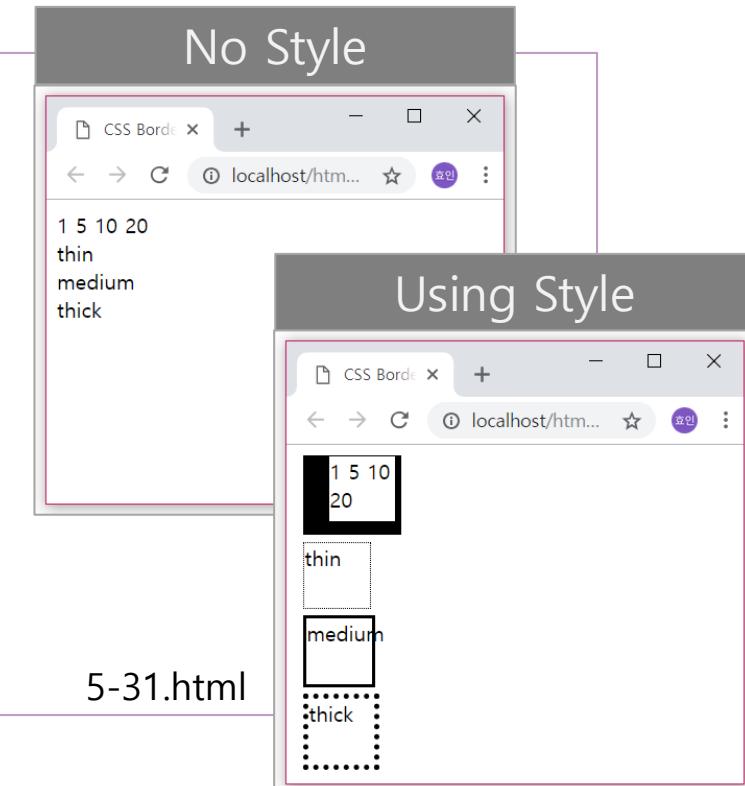


# 5. CSS 이해

## 5-5. 테이블 & 테두리 속성 (5/7) – border-width

- ✓ **border-width** 속성 사용법은 “E { border-width: **thin | medium | thick | 길이 값** }” 이다.
- ✓ 이 값은 테두리 너비 설정에 사용하며, 미리 정의된 “thin”, “medium”, “thick” 문자열 중 선택 가능.
- ✓ 1 ~ 4개의 값을 가질 수 있으며, 속성 값 적용 규칙은 **border-style**과 같다.
- ✓ **border-style**이 **border-width**보다 앞에 정의 되어야 한다.

```
<style type="text/css">
div { width: 50px; height: 50px; margin: 5px; }
div.target1 { border-style: solid; border-width: 1px 5px 10px 20px; }
div.target2 { border-style: dotted; border-width: thin; }
div.target3 { border-style: solid; border-width: medium; }
div.target4 { border-style: dotted; border-width: thick; }
</style>
</head>
<body>
    <div class="target1">1 5 10 20</div>
    <div class="target2">thin</div>
    <div class="target3">medium</div>
    <div class="target4">thick</div>
</body>
```

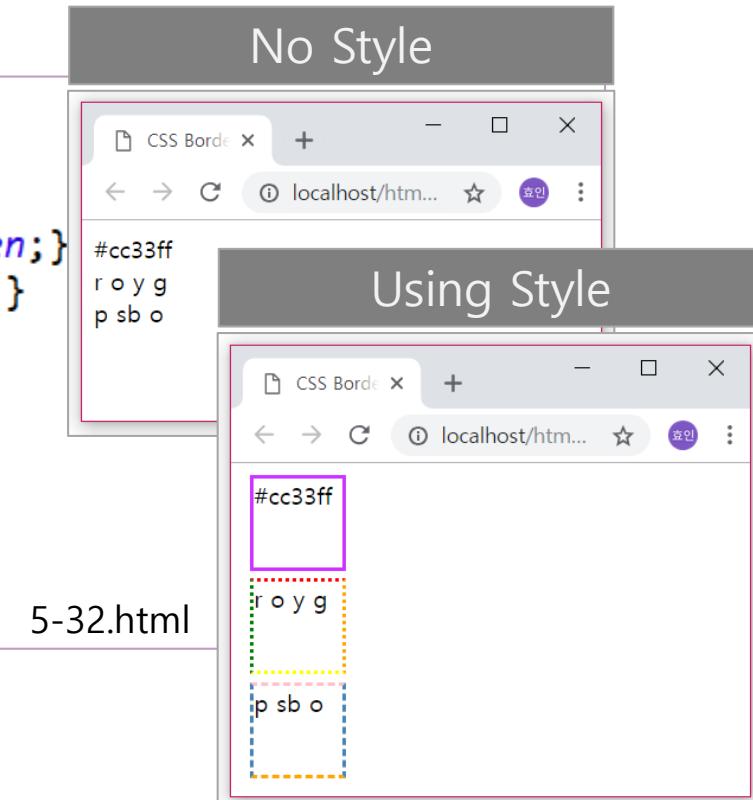


# 5. CSS 이해

## 5-5. 테이블 & 테두리 속성 (6/7) – border-color

- ✓ **border-color** 속성 사용법은 “E { border-color: color | transparent }” 이다.
- ✓ **테두리에 color값을 적용.**
- ✓ 1 ~ 4개의 값을 가질 수 있으며, 속성 값 적용 규칙은 **border-style**과 같다.
- ✓ **border-color가 border-width보다 앞에 정의 되어야 한다.**

```
<style type="text/css">
div { width: 60px; height: 60px; margin: 5px; }
div.target1 { border-style: solid; border-color: #cc33ff; }
div.target2 { border-style: dotted; border-color: red orange yellow green; }
div.target3 { border-style: dashed; border-color: pink steelblue orange; }
</style>
</head>
<body>
  <div class="target1">#cc33ff</div>
  <div class="target2">r o y g</div>
  <div class="target3">p sb o</div>
</body>
```



5-32.html

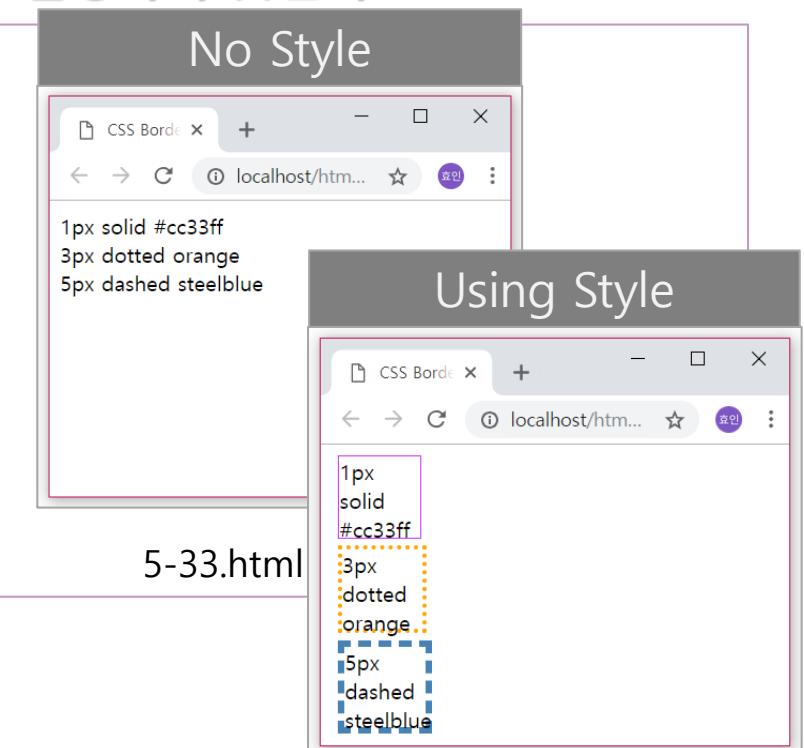


# 5. CSS 이해

## 5-5. 테이블 & 테두리 속성 (7/7) - border

- ✓ **border** 속성은 약식 표현 속성으로, 한번에 테두리 관련 속성들을 적용.
- ✓ 모든 4개의 테두리에 같은 너비, 스타일, 색상을 설정.
- ✓ **border-width**, **border-style**, **border-color** 순으로 작성해야 한다.
- ✓ **margin**, **padding**의 약식 속성과 달리 4개의 테두리에 다른 값을 설정하지 못한다.

```
<style type="text/css">
div { width: 60px; height: 60px; margin: 5px; }
div.target1 { border: 1px solid #cc33ff; }
div.target2 { border: 3px dotted orange; }
div.target3 { border: 5px dashed steelblue; }
</style>
</head>
<body>
  <div class="target1">1px solid #cc33ff</div>
  <div class="target2">3px dotted orange</div>
  <div class="target3">5px dashed steelblue</div>
</body>
```

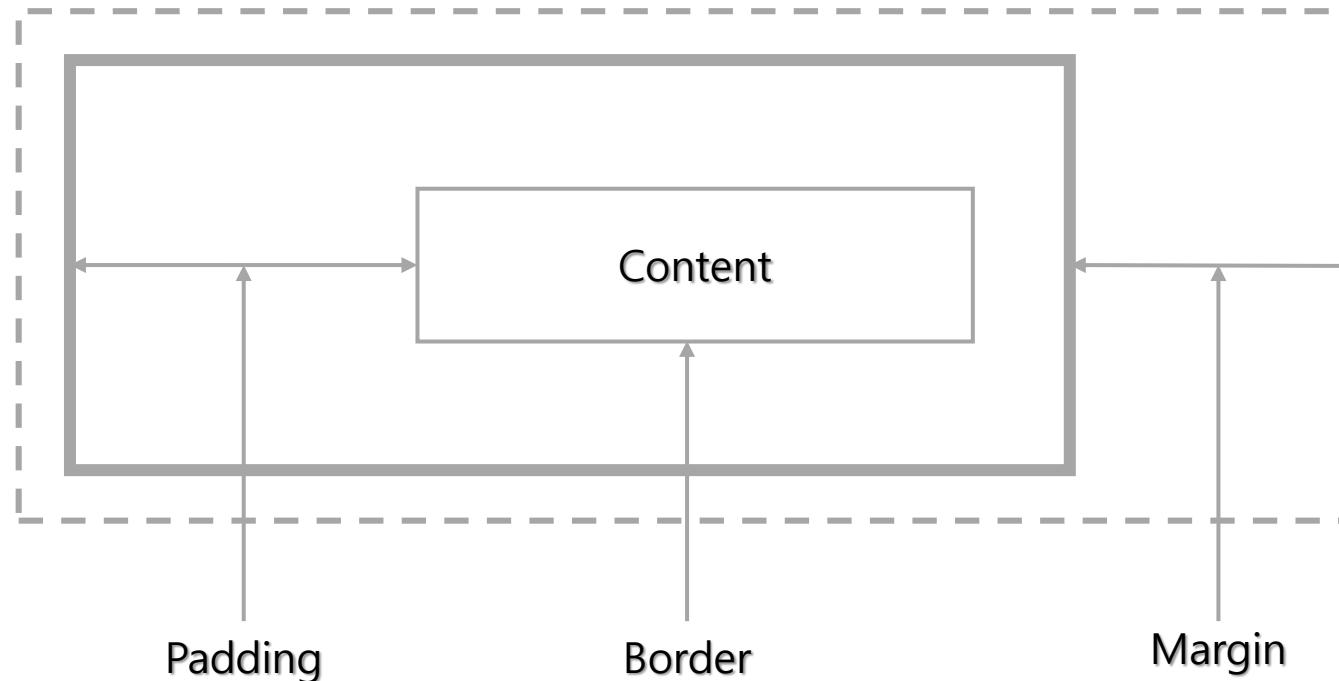




## 5. CSS 이해

### 5-6. 박스모델 (1/8) - 개요

- ✓ CSS는 모든 엘리먼트는 여러 겹의 상자로 둘러 쌓여 있다고 가정한다.
- ✓ 컨텐츠(Content), 패딩(Padding), 테두리(Border), 마진(Margin)으로 구분.
- ✓ 컨텐츠를 정렬 또는 위치를 지정하기 위해 Padding, Margin 속성을 활용.
- ✓ 박스모델을 활용하여 다양한 형태의 테이블이나 버튼을 직접 작성하여 활용.





# 5. CSS 이해

## 5-6. 박스모델 (2/8) – 예제 (1/2)

### ✓ 마진(margin), 패딩(padding), 테두리(border) example 1

```
<style type="text/css">
ul {
    background: lightgray; margin: 12px;
    border-style: dotted; border-width: medium;
    border-color: black; padding: 3px;
}
li {
    color: tomato; background: skyblue;
    border-style: solid; margin: 12px;
    padding: 12px; list-style: none;
}
li.withborder {
    border-style: dashed; border-width: thick;
    border-color: maroon;
}
</style>
</head>
<body>
    <ul>
        <li>첫번째 리스트입니다.</li>
        <li class="withborder">두번째 리스트입니다.</li>
    </ul>
</body>
```

The image shows two browser windows side-by-side. The left window, titled 'No Style', displays a simple list of two bullet points. The right window, titled 'Using Style', shows the same list where each item has been styled according to the provided CSS. The first list item has a solid black border and a light gray background. The second list item has a thick red dashed border and a light blue background. Both items have increased margins and padding compared to the 'No Style' version.

No Style

Using Style

첫번째 리스트입니다.  
두번째 리스트입니다.

첫번째 리스트입니다.  
두번째 리스트입니다.

5-34.html



# 5. CSS 이해

## 5-6. 박스모델 (3/8) – 예제 (2/2)

### ✓ 마진(margin), 패딩(padding), 테두리(border) example 2

```
<style type="text/css">
table { border: 0; padding: 2px; width: 100%; }
.horMargin { margin: 0% 5%; }
.cellPadding { padding-left: 15px; border: 2px solid steelblue; }
.para { text-align: justify; }
</style>
</head>
<body class="horMargin">
  <p class="para">현재 화면은 P Element를 사용하여 문단을 정리하면서 좌우 여백을  
Table을 만들어 Cell의 좌측에 15px의 여백을 주는 화면입니다.</p>
  <table>
    <tr>
      <td class="cellPadding">Row1 Col1</td>
      <td class="cellPadding">Row1 Col2</td>
    </tr>
    <tr>
      <td class="cellPadding">Row2 Col1</td>
      <td class="cellPadding">Row2 Col2</td>
    </tr>
  </table>
</body>
```

No Style

현재 화면은 P Element를 사용하여 문단을 정리하면서 좌우 여백을 브라우저 창의 5%로 하고, Table을 만들어 Cell의 좌측에 15px의 여백을 주는 화면입니다.

Row1 Col1 Row1 Col2  
Row2 Col1 Row2 Col2

Using Style

현재 화면은 P Element를 사용하여 문단을 정리하면서 좌우 여백을 브라우저 창의 5%로 하고, Table을 만들어 Cell의 좌측에 15px의 여백을 주는 화면입니다.

Row1 Col1	Row1 Col2
Row2 Col1	Row2 Col2

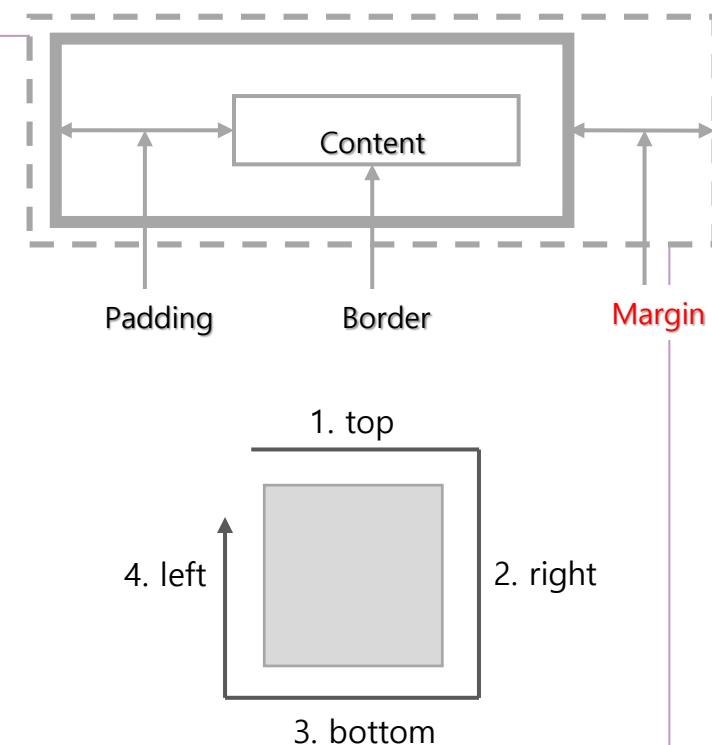


## 5. CSS 이해

### 5-6. 박스모델 (4/8) – margin (1/2)

- ✓ 마진(margin) 속성은 Box의 마진영역(margin area) 너비를 지정.
- ✓ 값이 1개일 때 모든 면에 적용, 값이 2개일 때 1번째는 {top, bottom}에 2번째는 {right, left}에 적용.
- ✓ 값이 3개일 때 1번째 값은 {top}에, 2번째 값은 {right, left}에, 3번째 값은 {bottom}에 적용.
- ✓ 값이 4개일 때는 {top, right, bottom, left} 순으로 적용.

```
<style type="text/css">
/* 모든 margin을 2em으로 설정. */
body { margin: 2em; }
/* top & bottom margin을 1em, right & left margin을 2em으로 설정. */
body { margin: 1em 2em; }
/* margin을 top = 1em, right = 2em, bottom = 3em, left = 2em으로 설정. */
body { margin: 1em 2em 3em; }
body {
    margin-top: 1em;
    margin-right: 2em;
    margin-bottom: 3em;
    margin-left: 2em;
}
</style>
</head>
<body>
</body>
```



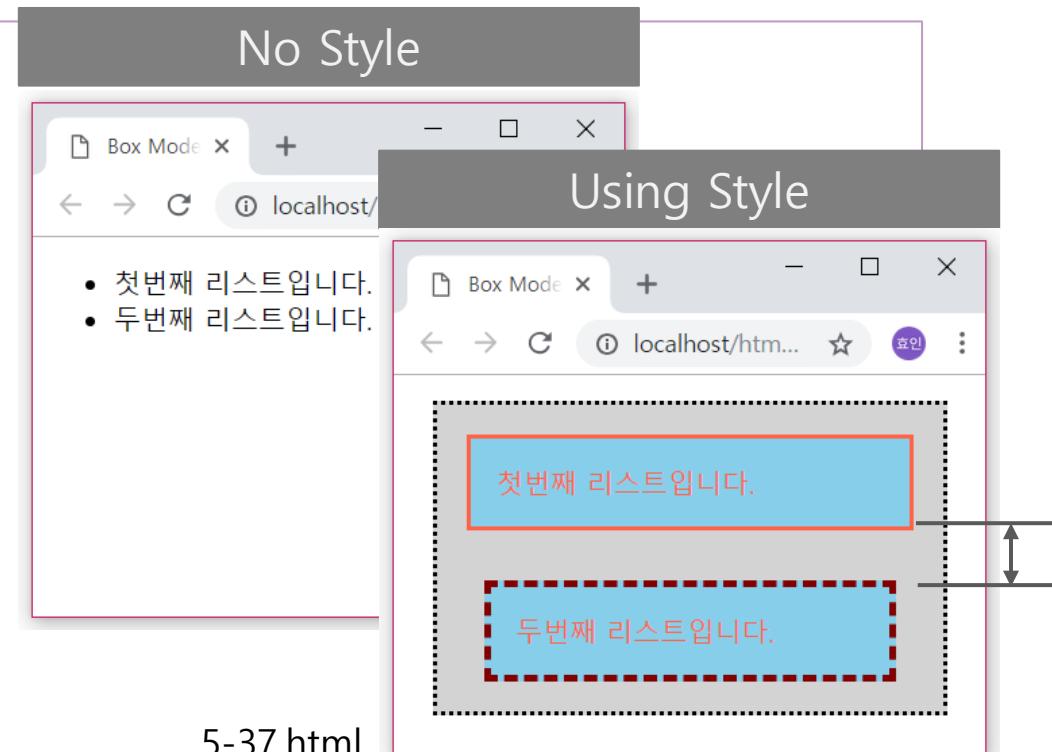


# 5. CSS 이해

## 5-6. 박스모델 (5/8) – margin (2/2)

- ✓ CSS2.0 부터 인접된 두 개 이상의 박스들의 인접마진이 통합되어 단일마진을 형성한다.
- ✓ 블록레벨(block level)의 경우, 두 개 이상의 인접 수직 마진은 통합되나, 수평마진은 통합되지 않는다.
- ✓ 유동된(floated)박스와 다른 박스의 수직 마진은 통합되지 않는다.
- ✓ 위치 값(position)이 absolute와 relative로 위치된 박스의 마진들은 통합되지 않는다.

```
<style type="text/css">
ul {
    background: lightgray; margin: 15px;
    border-style: dotted; border-width: medium;
    border-color: black; padding: 3px;
}
li {
    color: tomato; background: skyblue;
    border-style: solid; margin: 15px;
    padding: 15px; list-style: none;
}
li.withborder {
    margin-top: 30px; margin-right: 25px;
    margin-bottom: 15px; margin-left: 25px;
    border-style: dashed; border-width: thick;
    border-color: maroon;
}
</style>
```



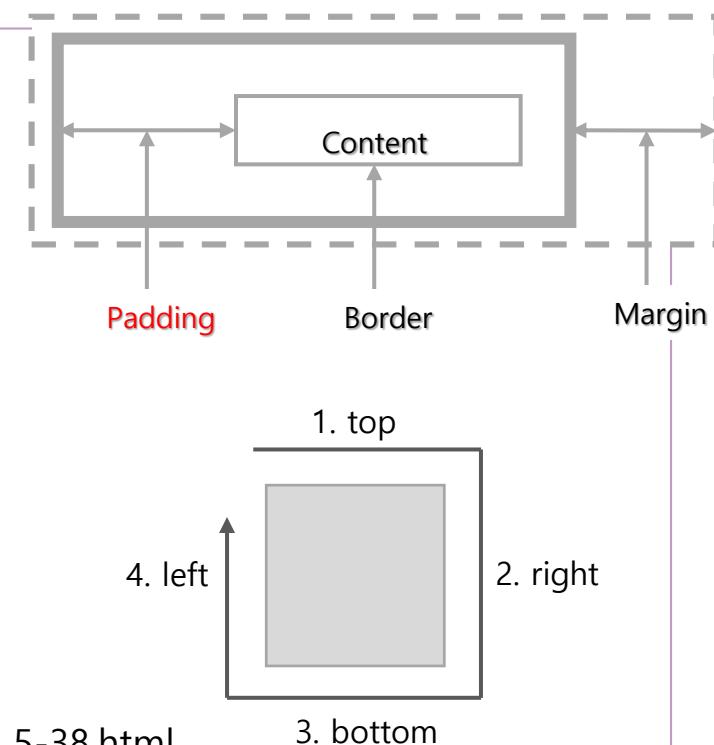


# 5. CSS 이해

## 5-6. 박스모델 (6/8) - padding

- ✓ 패딩(padding)의 속성은 box의 패딩영역(padding area) 너비를 지정.
- ✓ 값이 1개일 때 모든 면에 적용, 값이 2개일 때 1번째는 {top, bottom}에 2번째는 {right, left}에 적용.
- ✓ 값이 3개일 때 1번째 값은 {top}에, 2번째 값은 {right, left}에, 3번째 값은 {bottom}에 적용.
- ✓ 값이 4개일 때는 {top, right, bottom, left} 순으로 적용.

```
<style type="text/css">
ul {
    background: Lightgray; margin: 15px;
    border-style: dotted; border-width: medium;
    border-color: black; padding: 3px;
}
li {
    color: tomato; background: skyblue;
    border-style: solid; margin: 15px;
    padding: 15px; /* 모든 패딩들을 15px로 설정. */
    list-style: none;
}
li.withborder {
    padding-top: 30px; padding-right: 25px;
    padding-bottom: 15px; padding-left: 25px;
    border-style: dashed; border-width: thick;
    border-color: maroon;
}
</style>
```

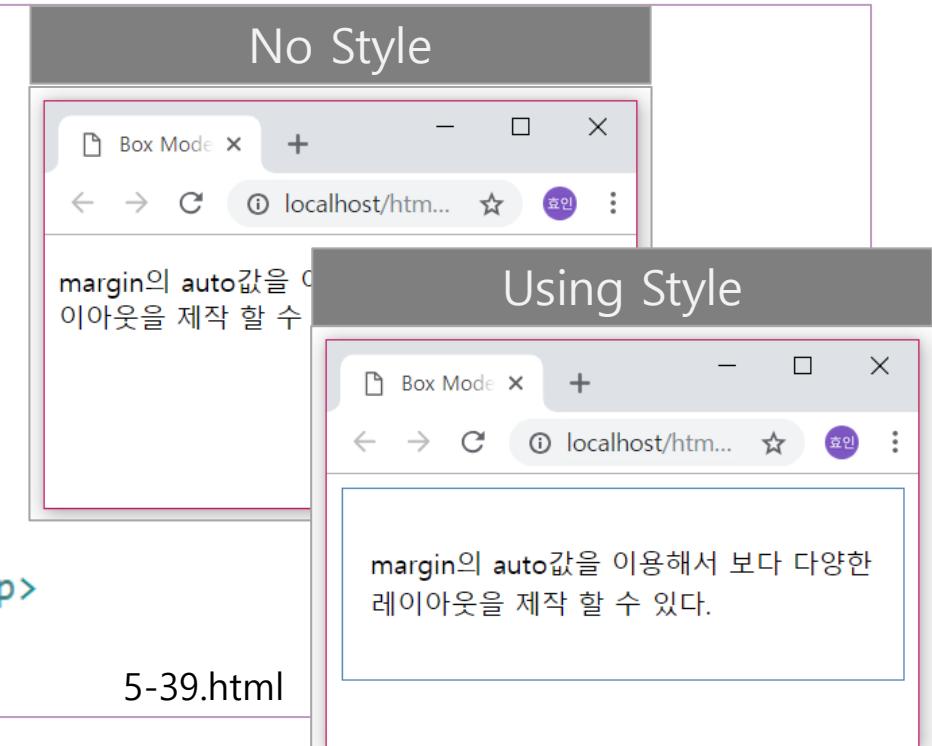


# 5. CSS 이해

## 5-6. 박스모델 (7/8) - 가운데정렬

- ✓ margin 속성을 사용하여 컨텐츠를 브라우저 화면의 가운데에 정렬되도록 설정 가능.
- ✓ 기존 HTML에서는 align 속성을 가운데로 설정 했으나, CSS에서는 여백을 조절하는 방법을 이용.
- ✓ 사용법은 “E { margin: 0 auto }” 이다. 첫 번째 값은 상,하 여백이고 두 번째 값은 좌,우 여백이다.
- ✓ 속성값 auto는 현재 엘리먼트를 중심으로 상,하 또는 좌,우의 여백을 균등하게 분배.(가운데정렬 효과)

```
<style type="text/css">
#content {
    width: 300px;
    padding: 1em;
    border: 1px solid steelblue;
    margin: 0 auto;
    line-height: 1.5em;
}
</style>
</head>
<body>
    <div id="content">
        <p>margin의 auto값을 이용해서 보다 다양한 레이아웃을 제작 할 수 있다.</p>
    </div>
</body>
```



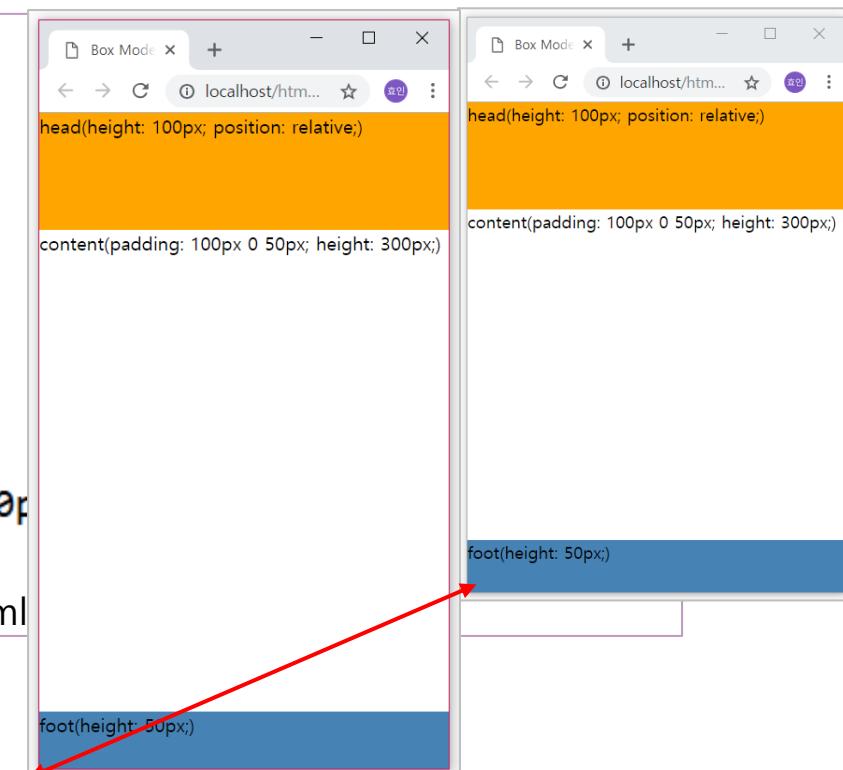
# 5. CSS 이해

## 5-6. 박스모델 (8/8) – 100% 높이를 유지하는 레이아웃

- ✓ 엘리먼트의 최소 높이를 보장하기 위해서 `min-height` 속성을 사용.
- ✓ `height`, `min-height` 속성을 같이 사용하여 레이아웃의 높이를 100%로 유지 가능.
- ✓ `<html><body>` 엘리먼트 하위에 100%높이를 유지하는 `<div>` 엘리먼트를 사용.
- ✓ 브라우저 화면 크기를 늘리더라도 컨텐츠는 항상 브라우저의 100% 높이를 유지.

```
<style type="text/css">
html, body { height: 100%; margin: 0; padding: 0; }
#head { height: 100px; background: orange; position: relative; }
#body { min-height: 100%; margin: -100px 0 -50px; }
#content-area { padding: 100px 0 50px; height: 300px; }
#foot { height: 50px; background: steelblue; }
</style>
</head>
<body>
  <div id="head">head(height: 100px; position: relative;)</div>
  <div id="body"><div id="content-area">content(padding: 100px 0 50px; height: 300px;)</div>
    <div id="foot">foot(height: 50px;)</div>
  </div>
</body>
```

5-40.html





## 5. CSS 이해

---

### 5-7. 포지셔닝 (1/10) - 개요

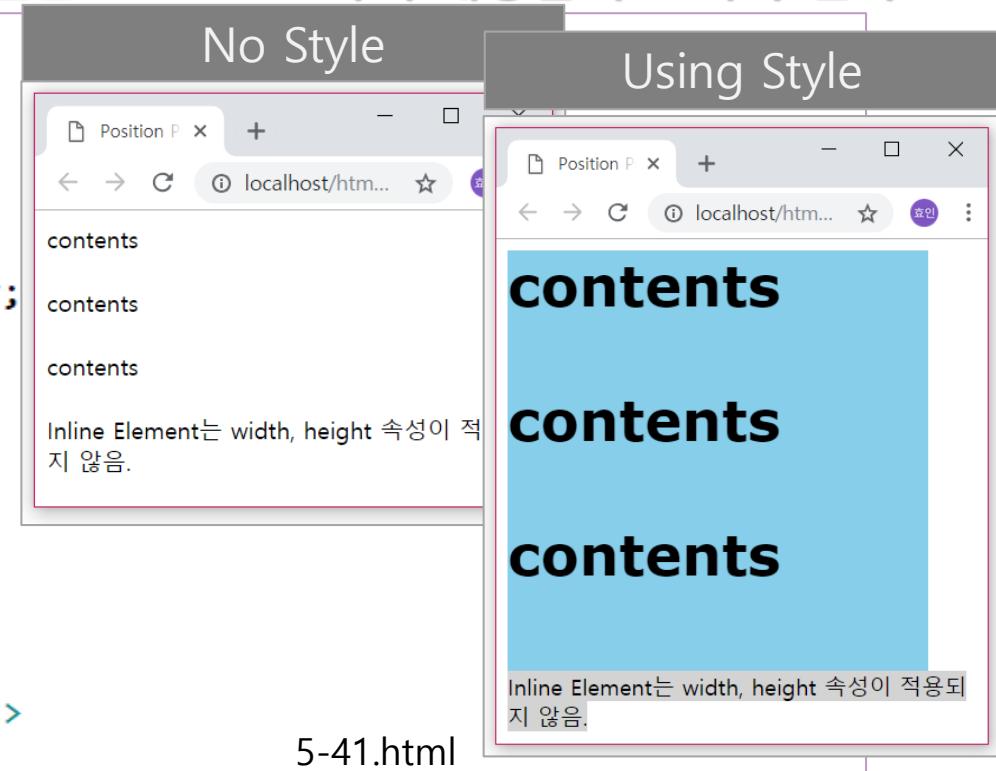
- ✓ 포지셔닝(Positioning)은 시각적인 측면에서 HTML의 가장 중요한 요소이다.
- ✓ HTML 내 부분 문서의 위치를 지정하거나 객체(Object)의 보임과 안보임(visibility)을 다룬다.
- ✓ 엘리먼트의 위치를 고정시키거나 브라우저의 크기에 따라 이동하는 등의 설정을 한다.
- ✓ 정적인 HTML을 JavaScript를 이용하여 동적(Dynamic)으로 만들기 위한 가장 기본적인 속성.

# 5. CSS 이해

## 5-7. 포지셔닝 (2/10) – width, height

- ✓ **length(길이 값)** : px, pt, cm, mm, in등의 길이단위 사용.
- ✓ **백분율(%)** : 상위 block에 대한 백분율 단위로 상위 block의 크기가 바뀌면 자신의 크기도 자동으로 변경.
- ✓ **auto (width의 경우)** : 100%, 자신의 상위 block이 허용하는 width 크기만큼 채운다.
- ✓ **auto (height의 경우)** : 0%, 이 경우 height를 결정하는 요인은 block box속의 내용물의 크기가 된다.

```
<style type="text/css">
div {
    width: 300px; height: 300px; top: 0px; left: 0px;
    font: bold 40px Verdana; background-color: skyblue;
}
span {
    width: 300px; height: 300px; background-color: lightgray;
}
</style>
</head>
<body>
    <div>
        contents<br><br>
        contents<br><br>
        contents<br><br>
    </div>
    <span>Inline Element는 width, height 속성이 적용되지 않음.</span>
</body>
```

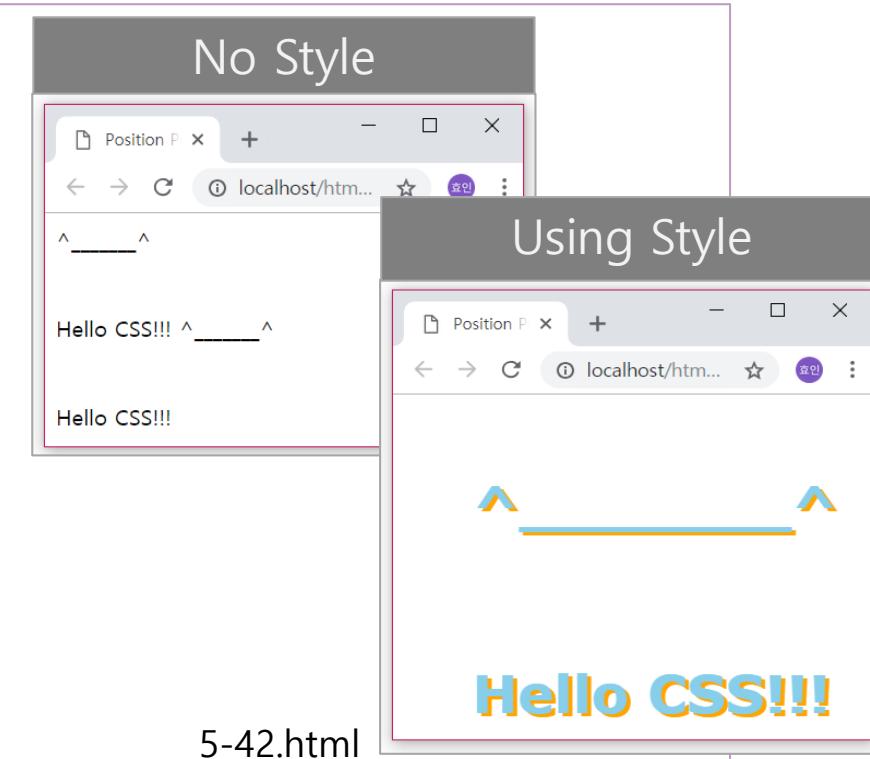


# 5. CSS 이해

## 5-7. 포지셔닝 (3/10) - position

- ✓ static : 기본(default)값으로 일반적인 내용물의 흐름. 상단(top)과 좌측(left)에서의 거리를 지정할 수 없다.
- ✓ relative : HTML 문서에서의 일반적인 내용물의 흐름을 말하지만 top, left 거리를 지정.
- ✓ absolute : 자신의 상위 box속에서의 top, left, right, bottom등의 절대적인 위치를 지정.
- ✓ fixed : 스크롤(scroll)이 일어나도 항상 화면상의 지정된 위치에 있다.

```
<style type="text/css">
div {
    position: relative; font: bold 40px Verdana;
}
.span1 {
    position: absolute; top: 53px; left: 53px; color: orange;
}
.span2 {
    position: absolute; top: 50px; left: 50px; color: skyblue;
}
</style>
</head>
<body>
    <div>
        <span class="span1">^____^<br><br><br>Hello CSS!!!</span>
        <span class="span2">^____^<br><br><br>Hello CSS!!!</span>
    </div>
</body>
```



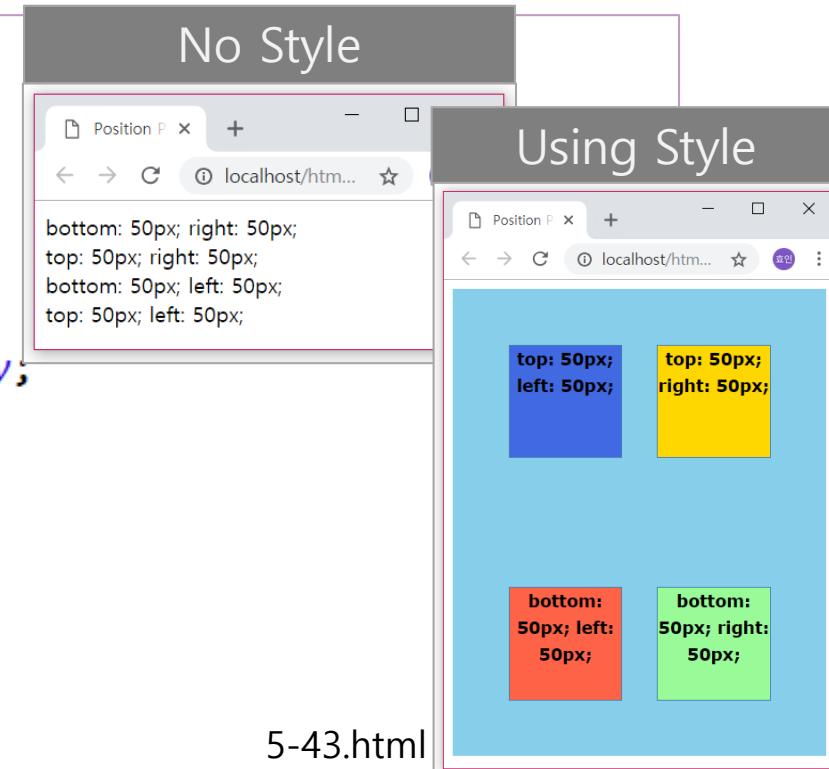


# 5. CSS 이해

## 5-7. 포지셔닝 (4/10) – top, left, bottom, right

- ✓ **top, left, bottom, right** 속성은 엘리먼트의 위치를 지정하기 위해 사용.
- ✓ 자신이 포함되어 있는 박스 속에서의 **top, left, bottom, right**에서의 거리를 지정하는 속성.
- ✓ 사용법은 “**E { top: 길이 값(length:px,cm) | 백분율(%) | auto }"** 이다.
- ✓ 각 **<div>** 엘리먼트의 **position** 속성이 **absolute**로 설정이 되어있기 때문에 절대적인 위치 지정 가능.

```
<style type="text/css">
.side {
    position: absolute; border: 1px solid steelblue;
    width: 100px; height: 100px; text-align: center;
}
.output {
    height: 400px; position: relative; background: skyblue;
    font: bold 15px/1.6em Verdana; padding: 10px; text-align: justify,
}
.palegreen {
    bottom: 50px; right: 50px; background-color: palegreen;
}
.gold { top: 50px; right: 50px; background-color: gold; }
.tomato { bottom: 50px; left: 50px; background-color: tomato; }
.royalblue { top: 50px; left: 50px; background-color: royalblue; }
</style>
```





## 5. CSS 이해

### 5-7. 포지셔닝 (5/10) – overflow (1/2)

- ✓ **overflow** 속성은 상위 엘리먼트에 속한 내용이 엘리먼트의 크기보다 클 경우 어떻게 처리할 것인지 설정.
- ✓ 속성값을 **visible**로 설정하면 box속의 내용을 모두 표시. 내용의 크기에 따라 box의 가로, 세로 폭이 늘어남.
- ✓ 속성값을 **hidden**으로 설정하면 box의 **width**, **height**를 지정했을 경우, 지정된 범위를 넘치는 내용은 보이지 않는다.
- ✓ 속성값을 **auto**로 설정하면 지정된 범위를 넘치는 내용은 스크롤바를 이용하여 표시.

```
<style type="text/css">
.div1 { width: 300px; height: 300px; border: 3px solid tomato; align-content: center; }
.table1 { border-collapse: collapse; width: 100%; border: 3px solid steelblue; }

.non { width: 200px; height: 200px; border: 3px solid orange; }
.auto1 { width: auto; height: auto; border: 3px solid skyblue; }
.visible { width: 200px; height: 200px; border: 3px solid pink; overflow: visible; }
.hidden { width: 200px; height: 200px; border: 3px solid magenta; overflow: hidden; }
.auto2 { border: 3px solid lightgray; overflow: auto; }
.scroll { border: 3px solid darkgray; overflow: scroll; }
</style>
</head>
<body>
<div class="visible">안녕하세요. 안녕하세요. 안녕하세요. 안녕하세요. 안녕하세요. * 100 </div>
<table class="table1">
<tr>
```

# 5. CSS 이해

## 5-7. 포지셔닝 (5/10) – overflow (2/2)

```
width: 200px; height: auto; overflow: visible; position: relative;
```

non

```
width: 200px; height: auto; overflow: auto; position: relative;
```

auto1

```
width: 200px; height: auto; overflow: visible; position: relative;
```

visible

5-44.html

```
width: 200px; height: auto; overflow: hidden; position: relative;
```

hidden

```
width: 200px; height: auto; overflow: auto; position: relative;
```

auto2

```
width: 200px; height: auto; overflow: scroll; position: relative;
```

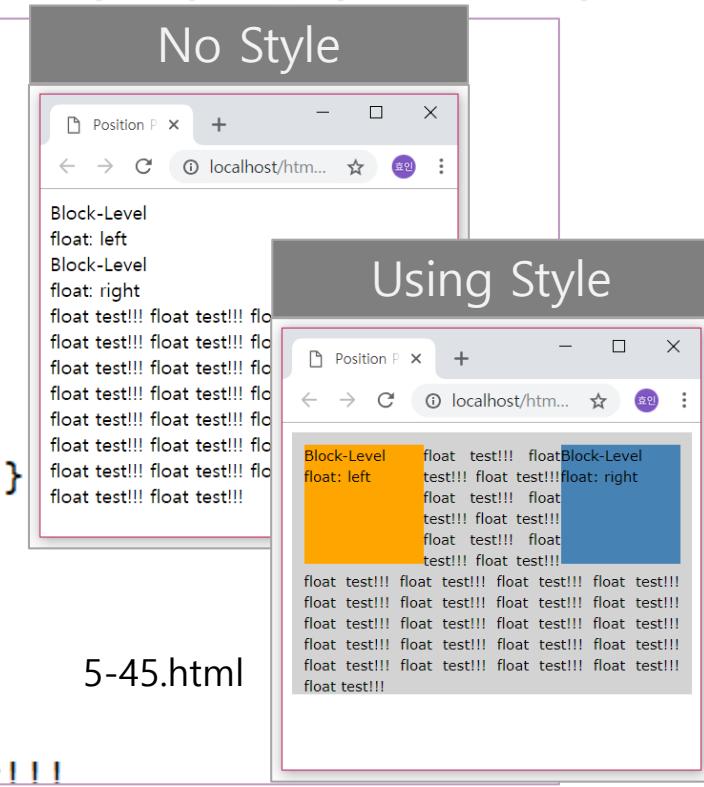
scroll

# 5. CSS 이해

## 5-7. 포지셔닝 (6/10) - float

- ✓ **float 속성은 박스가 화면의 어느 위치에 배치할 것인지를 설정하기 위해 사용.**
- ✓ 속성값을 **left**로 설정하면 그림이나 박스가 왼쪽에 배치되고, 글씨는 박스의 오른쪽으로 흐른다.
- ✓ 속성값을 **right**로 설정하면 그림이나 박스가 오른쪽에 배치되고, 글씨는 박스의 왼쪽으로 흐른다.
- ✓ 속성값을 **none**으로 설정하면 그림이나 박스가 왼쪽에 배치되고, 글씨는 첫 줄만 박스의 오른쪽으로 흐른다.

```
<style type="text/css">
.output {
    background: lightgray;
    font: 12px/1.5em Verdana;
    font-family: "Verdana", "gulim", "굴림", "굴림체";
    padding: 10px;
    text-align: justify;
    height: 200px;
}
#div1 { width: 100px; height: 100px; background: orange; float: left; }
#div2 { width: 100px; height: 100px; background: steelblue; float: right; }
</style>
</head>
<body>
    <div class="output">
        <div id="div1">Block-Level<br>float: left</div>
        <div id="div2">Block-Level<br>float: right</div>
        float test!!! float test!!! float test!!! float test!!!
    </div>
</body>
```



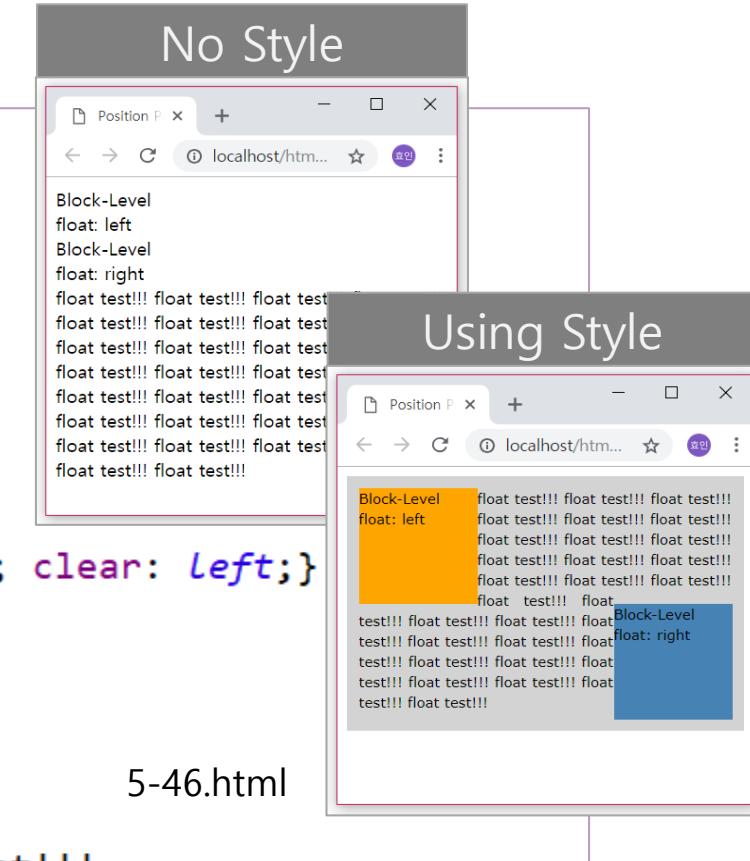
5-45.html

# 5. CSS 이해

## 5-7. 포지셔닝 (7/10) - clear

- ✓ **clear 속성은 float 속성이 가지고 있는 값을 초기화 하기 위해 사용.**
- ✓ **속성값을 left, right로 설정하여 왼쪽 또는 오른쪽 float 속성값을 취소할 수 있다.**
- ✓ **속성값을 both로 설정하여 양쪽 모두의 float 속성값을 취소할 수 있다.**
- ✓ **속성값을 none으로 설정하면 clear를 설정하지 않은 것과 같다.**

```
<style type="text/css">
.output {
    background: Lightgray;
    font: 12px/1.5em Verdana;
    font-family: "Verdana", "gulim", "굴림", "굴림체";
    padding: 10px;
    text-align: justify;
    height: 200px;
}
#div1 { width: 100px; height: 100px; background: orange; float: left; }
#div2 { width: 100px; height: 100px; background: steelblue; float: right; clear: left; }
</style>
</head>
<body>
    <div class="output">
        <div id="div1">Block-Level<br>float: left</div>
        <div id="div2">Block-Level<br>float: right</div>
        float test!!! float test!!! float test!!! float test!!! float test!!!
    </div>
</body>
</html>
```





## 5. CSS 이해

### 5-7. 포지셔닝 (8/10) – clip (1/2)

- ✓ **clip 속성은 이미지 사이즈가 클 경우 이미지를 일부 가려서 표시하기 위해 사용.**
- ✓ **속성값으로 rect()에 명시된 사각형 크기만큼 가려서 화면에 표시.**
- ✓ **rect(**top, right, bottom, left**)순으로 픽셀 값을 설정.**
- ✓ **auto로 설정하면 이미지를 가리지 않고 모두 보여줌.**

```
<style type="text/css">
.box { position: absolute; border: 1px solid royalblue; font: 12px verdana; }
.auto { left: 50; clip: rect(auto); }
.top { left: 150; clip: rect(15px auto auto auto); }
.right { left: 250; clip: rect(auto 15px auto auto); }
.bottom { left: 350; clip: rect(auto auto 15px auto); }
.left { left: 450; clip: rect(auto auto auto 15px); }
</style>
</head>
<body>
  

<!-- 
  

-->
</body>
```



## 5. CSS 이해

### 5-7. 포지셔닝 (8/10) – clip (2/2)

No Style



Using Style



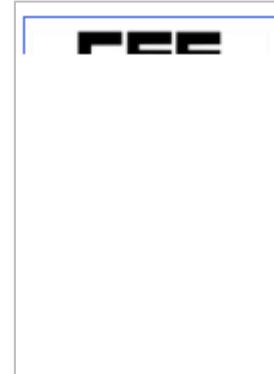
auto



top



right



bottom



left



## 5. CSS 이해

### 5-7. 포지셔닝 (9/10) – visibility (1/2)

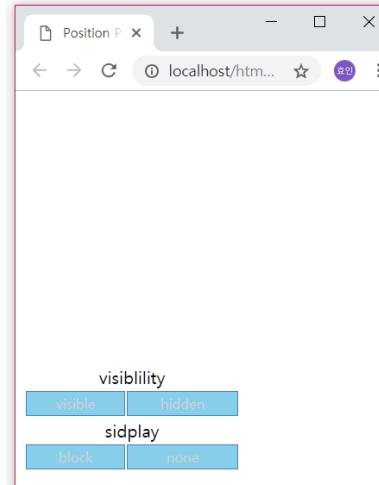
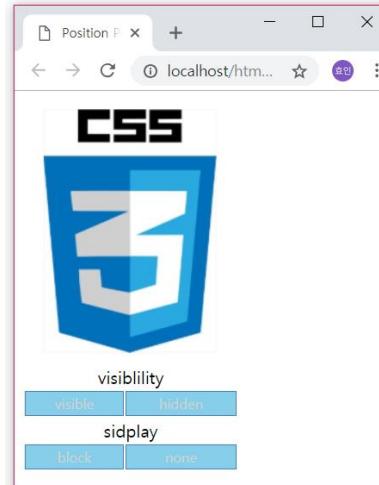
- ✓ **visibility** 속성은 엘리먼트를 화면에 보이거나 숨기기 위해서 사용.
- ✓ 속성값을 **visible**로 설정하면 화면에 표시하고 **hidden**으로 설정하면 화면에서 숨김.
- ✓ **hidden**으로 설정된 엘리먼트는 화면에 표시되지는 않지만 면적은 차지하고 있음.
- ✓ 화면에서 숨기고 면적도 차지하지 않도록 하기 위해서는 **display**속성을 사용.

```
<style type="text/css">
div,table { width: 200px; text-align: center; }
.td_button {
    height: 20px; background-color: skyblue;
    border: 1px solid steelblue; cursor: pointer;
    font-size: 13px; color: lightgray;
    text-align: center;
}
.output { background: white; }
.visible { visibility: visible; }
.hidden { visibility: hidden; }
.block { display: block; }
.none { display: none; }
</style>
```

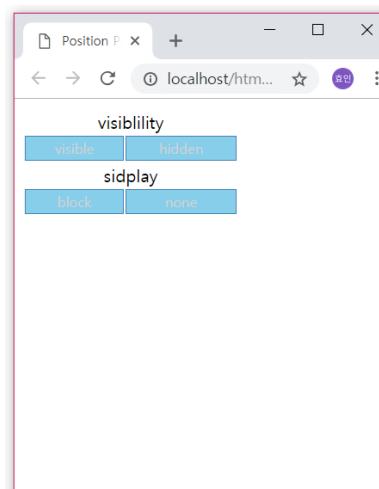
# 5. CSS 이해

## 5-7. 포지셔닝 (9/10) – visibility (2/2)

```
<script src="https://ajax.googleapis.com/ajax/Libs/jquery/3.3.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).on("click", "td", function() {
    $("#imgview").attr("class", $(this).text());
});
</script>
</head>
<body>
    <div id="imgview"></div>
    <table>
        <tr>
            <td colspan="2"> visibility</td>
        </tr>
        <tr>
            <td class="td_button">visible</td>
            <td class="td_button">hidden</td>
        </tr>
        <tr>
            <td colspan="2"> sidplay</td>
        </tr>
        <tr>
            <td class="td_button">block</td>
            <td class="td_button">none</td>
        </tr>
    </table>
</body>
```



visibility : hidden



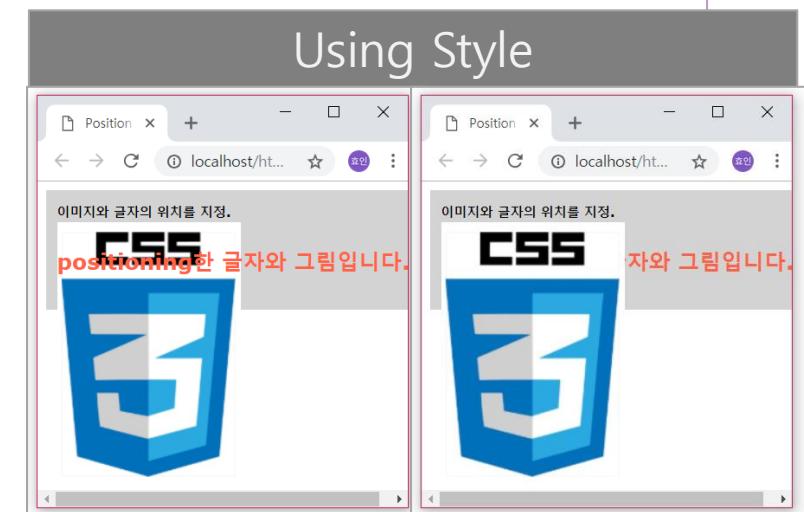
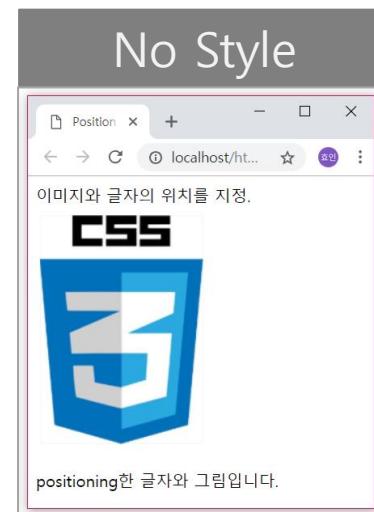
display : none

# 5. CSS 이해

## 5-7. 포지셔닝 (10/10) – z-index

- ✓ z-index 속성은 여러 개의 엘리먼트를 화면에 쌓아서 표시하기 위해 사용.
- ✓ 사용법은 “E { z-index: 정수 값(양수, 음수가능) }” 이다.
- ✓ z-index 값이 큰 엘리먼트를 위에 표시.
- ✓ z-index 값을 auto로 설정하면 부모 엘리먼트의 레벨과 같은 값으로 설정되며 이 값이 기본값이다.

```
<style type="text/css">
#img1 { position: absolute; z-index: 1; }
#p1 { position: absolute; color: tomato; font: bold 20px/1.6em Verdana; z-index: 2; }
.output {
    width: 100%;
    background: lightgray;
    font: bold 12px/1.6em Verdana;
    padding: 10px;
    text-align: justify;
    word-break: keep-all;
    position: relative;
    height: 90px;
}
</style>
</head>
<body>
    <div class="output">
        이미지와 글자의 위치를 지정.<br>
        
        <p id="p1">positioning한 글자와 그림입니다.</p>
    </div>
</body>
```





# 5. CSS 이해

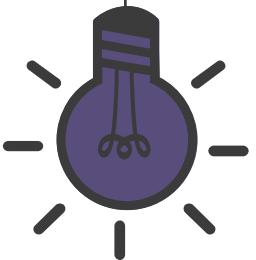
---

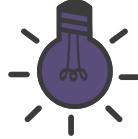
## 5-8. 요약

- ✓ CSS 선택자는 HTML문서 내에서 스타일을 적용하고자 하는 element를 선택.
- ✓ 일반 선택자, 복합 선택자, 가상 클래스 선택자, 가상 엘리먼트 선택자, 속성 선택자가 존재.
- ✓ 포지셔닝 속성은 HTML문서 내에서 element의 화면배치나 스크롤 여부 등을 설정.
- ✓ font 속성은 글꼴, 크기 등의 글자 꾸밈을 담당.
- ✓ text 속성은 여백, 들여쓰기, 정렬 등의 공간 설정 기능을 담당.

06

## HTML5와 Semantic





# 6. HTML5와 Semantic

## 6-1. semantic?

- ✓ Semantic의 사전적 의미는 '의미론적인', '의미가 통하는'이다.
- ✓ 검색 로봇 또는 스크린 리더등의 기계가 쉽게 해석하고 분석 할 수 있도록 만들어진 태그.
- ✓ HTML4에서 <div>와 id 속성을 이용하여 화면을 구성.
- ✓ id의 이름을 정하는 규칙이 없기때문에 문서의 구조를 이해하는데 한계. 웨어러블 장치나 스마트 기기를 이용하여 웹문서를 읽는 데 부적합 구조.
- ✓ 해결책으로 HTML5에서는 웹문서 레이아웃을 표준화하는 semantic tag를 이용.
- ✓ 사이트의 제목과 로고, 검색 창 등이 있는 헤더(header), 여러 내용이 있는 본문(contents : section + articles), 본문 외 내용을 나타내는 사이드바(aside), 저작권 정보와 제작자 정보를 표시하는 푸터(footer)로 이루어져 있다.



# 6. HTML5와 Semantic

## 6-1. semantic?

1. `<font size="6"><b>Hello HTML5</b></font>`
2. `<h1>Hello HTML5</h1>`

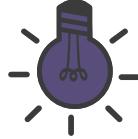
- 위 예시의 1과 2는 출력되는 화면은 같다.
- 1행의 요소는 의미론적으로 어떤 의미도 가지고 있지 않다. 개발자가 의도한 요소의 의미를 명확하게 나타내지 않고 다만 폰트 크기와 볼드체를 지정하는 메타데이터만을 브라우저에게 알리고 있다.
- 2행의 요소는 header(제목) 중 가장 상위 레벨이라는 의미를 내포하고 있어서 개발자가 의도한 요소의 의미가 명확히 드러나고 있다. 이것은 코드의 가독성을 높이고 유지보수를 쉽게 한다.

- ✓ semantic tag란 브라우저, 검색엔진, 개발자 모두에게 콘텐츠의 의미를 명확히 설명하는 역할을 한다.
- ✓ semantic 웹이란 웹에 존재하는 수많은 웹페이지들에 메타데이터(Metadata)를 부여하여, 기존의 잡다한 데이터 집합이었던 웹페이지를 ‘의미’ 와 ‘관련성’을 가지는 거대한 데이터베이스로 구축하고자 하는 발상이다.

Hello HTML5

Hello HTML5

실행화면



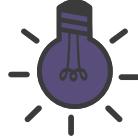
# 6. HTML5와 Semantic

## 6-1. semantic?

- ✓ HTML 요소는 non-semantic 요소, semantic 요소로 구분.
  - [non-semantic 요소](#)는 div, span 등이 있으며 이들 태그는 content에 대하여 어떤 설명도 하지 않는다.
  - [semantic 요소](#)는 form, table, img 등이 있으며 이들 태그는 content의 의미를 명확히 설명한다.

tag	설명
header	헤더(머리말) 지정.
nav	문서 간의 네비게이션 지정.
aside	본문 이외의 내용표시.
section	본문의 여러 내용(article)을 포함.
article	본문의 주 내용이 포함.
footer	제작정보와 저작권 표시등 포함.

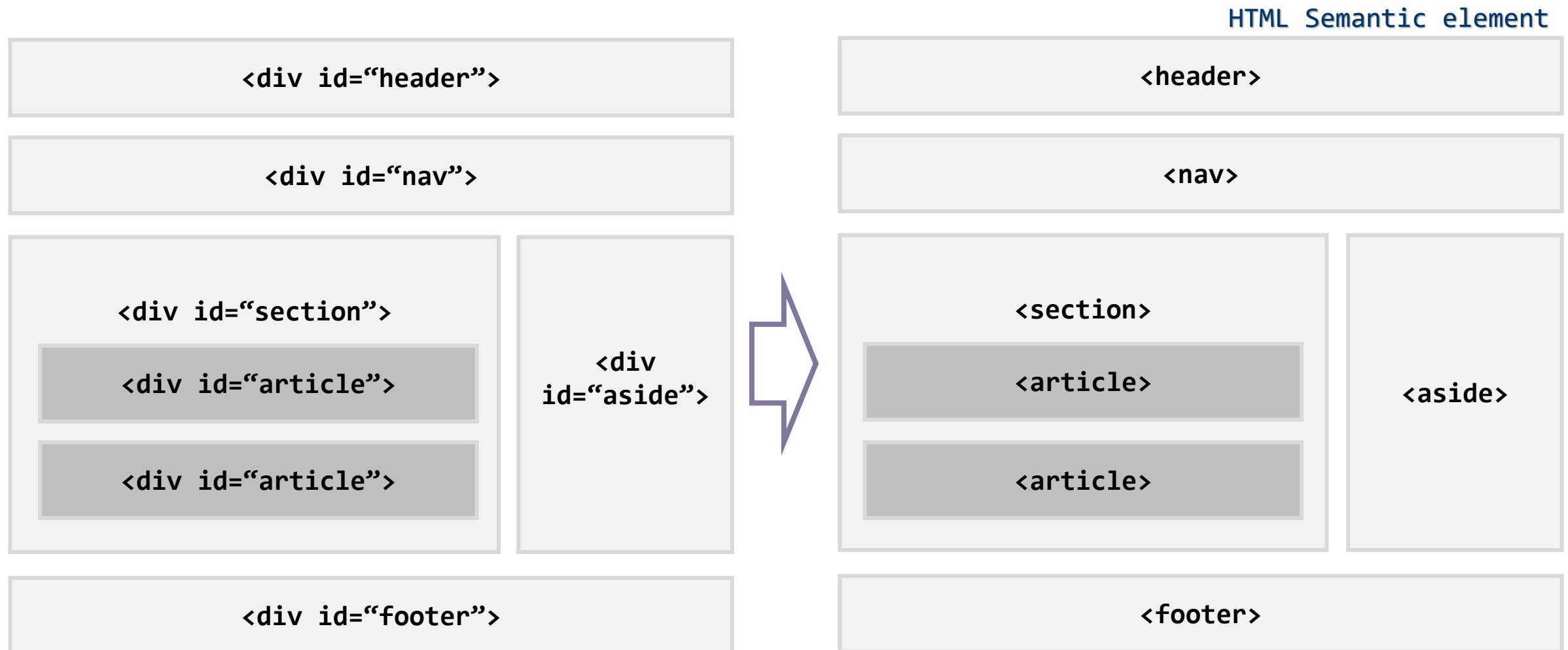
HTML5에 새롭게 추가된 semantic Tag



# 6. HTML5와 Semantic

## 6-1. semantic?

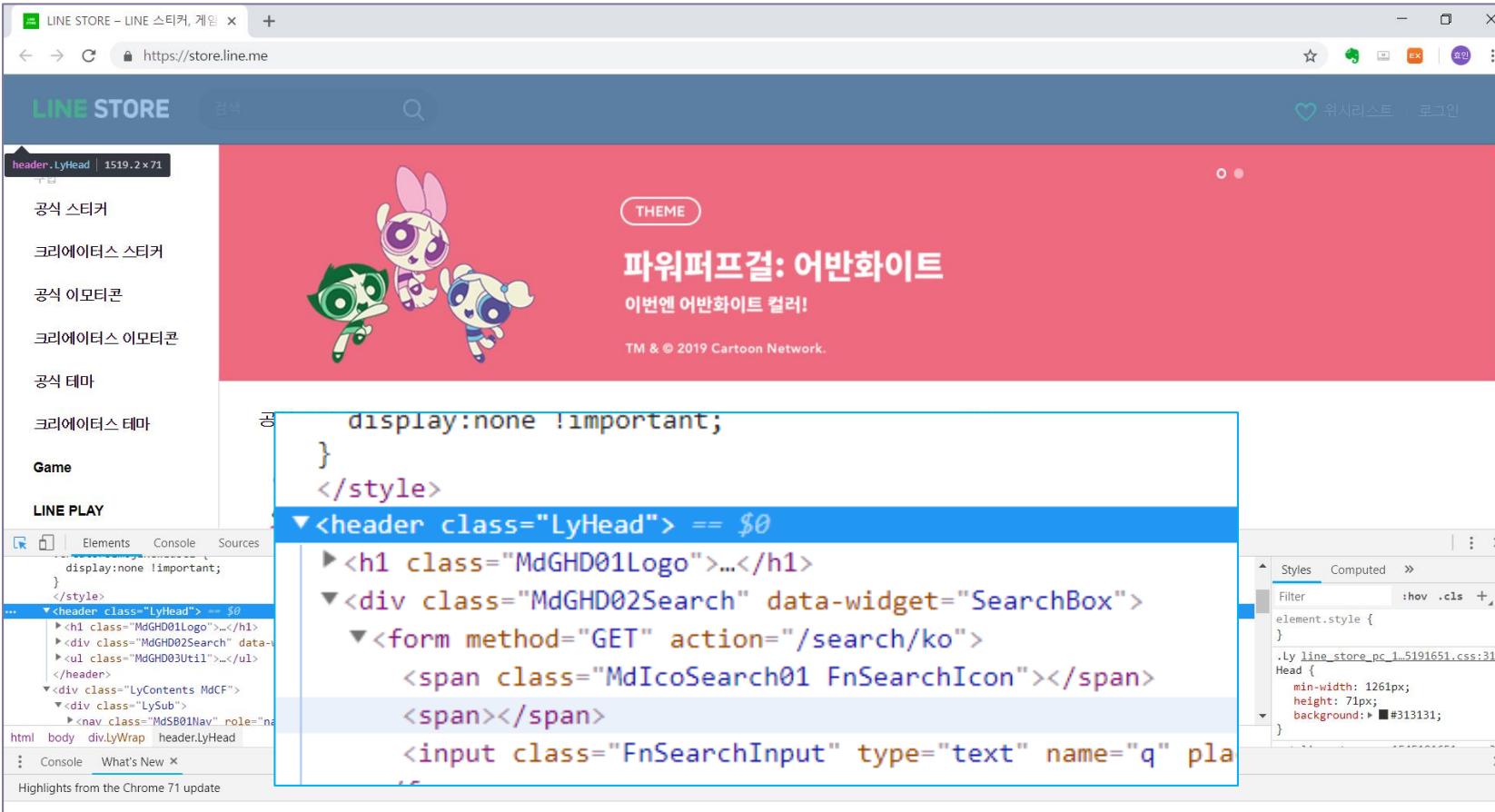
- ✓ HTML4와 HTML5(semantic)



# 6. HTML5와 Semantic

## 6-2. semantic (1/5) : header – 머리말 지정

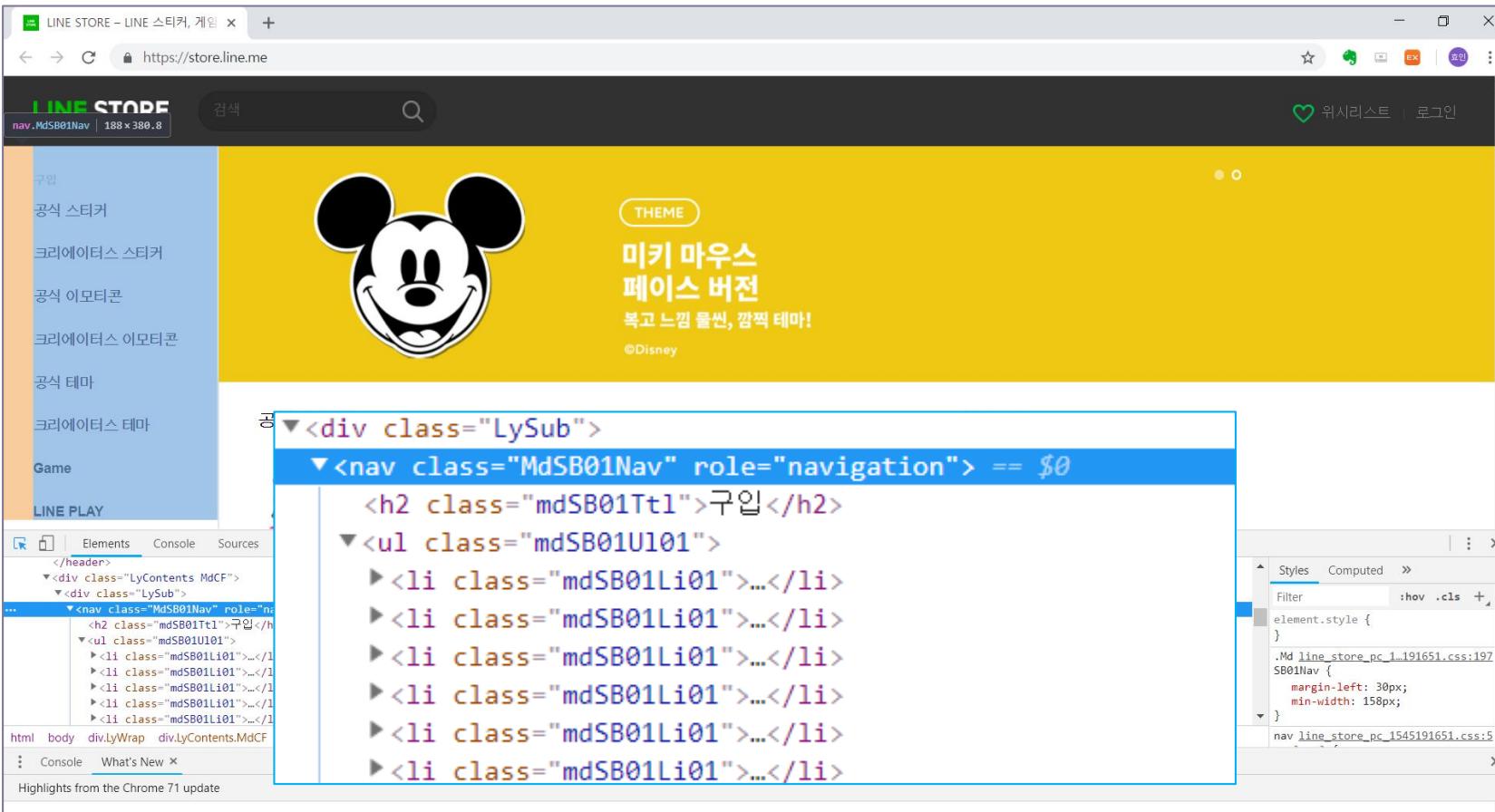
- ✓ **header**는 주로 **<form>** 태그를 사용하여 검색 창을 넣거나 **<nav>** 태그를 사용하여 사이트의 메뉴를 삽입. 본문에 사용하여 해당 부분의 머리말로도 사용.



# 6. HTML5와 Semantic

## 6-2. semantic (2/5) : nav – 문서를 연결하는 navigation link

- ✓ 동일 사이트 내의 문서나 다른 사이트의 문서로 연결하는 링크 모음.
- ✓ <nav>는 단독으로 사용되거나 <header>, <footer>, <aside>내에서 사용.



# 6. HTML5와 Semantic

## 6-2. semantic (3/5) : section – 주제별 콘텐츠 영역 표시

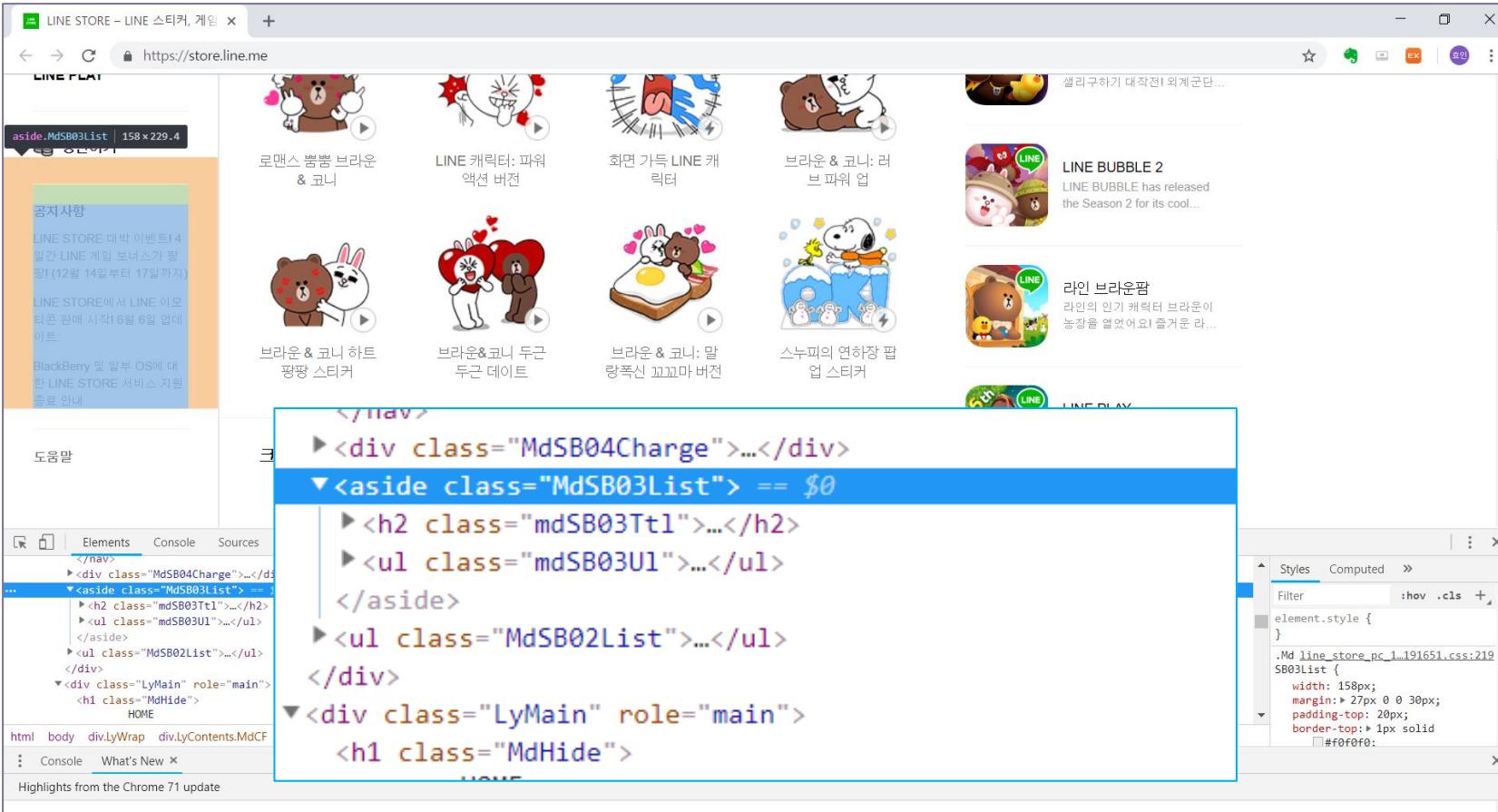
- ✓ <section>은 문맥 흐름 중에서 contents를 주제별로 묶을 때 사용.
- ✓ <section>안에 여러 개의 <article>을 넣어 contents의 내용을 표현.



# 6. HTML5와 Semantic

## 6-2. semantic (4/5) : aside – 본문 이외의 내용 표시

- ✓ 일반적으로 사이트의 왼쪽이나 오른쪽 또는 하단에 위치.
- ✓ 본문 내용 외에 주변에 표시되는 기타내용을 표현.(광고, 링크모음 등..)



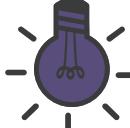
The screenshot shows the LINE STORE website at <https://store.line.me>. On the left, there is a sidebar with the title "공지사항" (Announcements) containing text about a LINE STORE event and a "도움말" (Help) section. The main content area displays various LINE stickers and their descriptions. A specific section of the sidebar is highlighted with a blue selection box, showing the HTML code for that part of the page. The code includes elements like <aside>, <h2>, and <ul>. The right side of the browser window shows the CSS styles applied to the selected element, including width, margin, padding, and border properties.

```
</nav>
  > <div class="MdSB04Charge">...</div>
  > <aside class="MdSB03List"> == $0
    > <h2 class="mdSB03Ttl">...</h2>
    > <ul class="mdSB03U1">...</ul>
  </aside>
  > <ul class="MdSB02List">...</ul>
</div>
  > <div class="LyMain" role="main">
    <h1 class="MdHide">
      HOME
    </h1>
  </div>
```

Styles Computed

```
:hover .cls + .Md_line_store_pc_1_191651.css:219
  S003list {
    width: 158px;
    margin: 27px 0 30px;
    padding-top: 20px;
    border-top: 1px solid #f0f0f0;
```

# 6. HTML5와 Semantic



## 6-2. semantic (5/5) : footer – 제작 정보와 저작권 정보등 표시

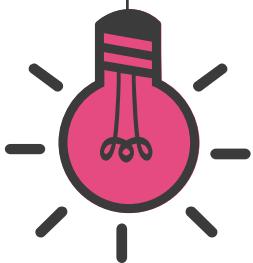
- ✓ 일반적으로 사이트의 하단부에 표현.

The screenshot shows the LINE STORE website at <https://store.line.me>. The main content area displays a grid of six sticker designs: 'galaxy pastel', 'Galaxy Pastel: milky way', '마법의 꿈', 'FACE (djungarian hamster)', 'SIMPLE / beige-blue.', and '딸기우유빙수'. Below this grid, there are two sections: 'LINE CREATORS STUDIO' and 'LINE CREATORS MARKET'. The developer tools (Elements, Console, Sources) are open in the bottom-left corner, showing the HTML code for the footer. The 'Elements' tab highlights the footer element with the class 'LyFoot'. The 'Console' tab shows some JavaScript logs. The 'Sources' tab shows the full page source code. The right side of the developer tools includes a 'Styles' panel with CSS rules for the footer element.

```
</div>
<footer class="LyFoot" role="contentinfo">...</footer> == $0
  <!-- friend item templates -->
  ><script type="text/template" class="FnPresentFriendTemplate"
  <!-- templates for radio buttons -->
  ><script type="text/template" class="FnCallPlanRadioTemplate"
</div>
<div>
</div>
<div class="MdOverlay FnOverlay" style="display:none;"></div>
```

07

멀티미디어





## 7. 멀티미디어

### 7-1. 플러그인 사용 - <object>, <embed> tag

- ✓ HTML5를 지원하지 못하는 브라우저에 적용.
- ✓ <object> tag.
  - 웹 브라우저에서 직접 재생할 수 없는 자바 애플릿이나 PDF, Flash 같은 콘텐츠를 포함할 때 사용.
  - <object data="경로" type="유형" [name="이름" width="너비" height="높이"]></object>

속성	설명
data	외부 파일의 경로를 지정.
type	포함시킨 내용의 유형을 지정.
name	다른 요소들과 구분할 수 있는 이름 지정.
width	너비 값 지정.
height	높이 값 지정.



## 7. 멀티미디어

### 7-1. 플러그인 사용 - <object>, <embed> tag

#### ✓ <embed> tag.

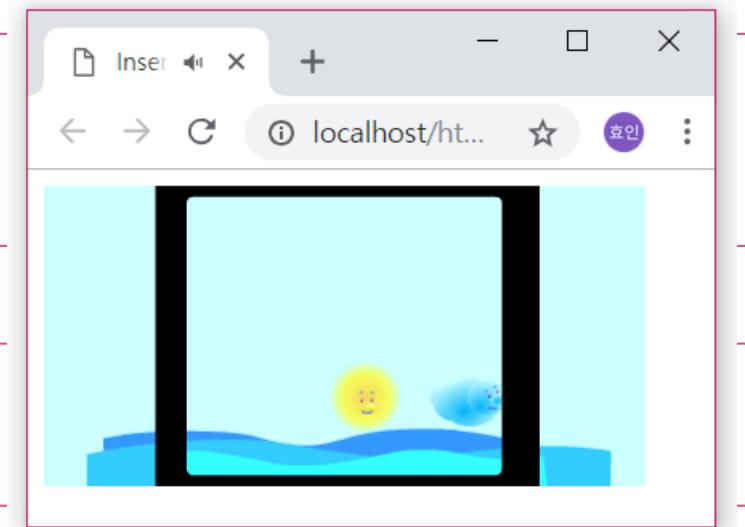
- <embed> tag도 웹 브라우저에서 직접 재생할 수 없는 콘텐츠를 포함할 때 사용.
- <embed src="경로" type="유형" width="너비" height="높이">
- <object> tag와 달리 닫는 태그가 없다.
- 주로 <object> tag를 지원하지 않는 이전 브라우저에서 사용.

```
<body>
<object data="test.swf" type="application/x-shockwave-flash"
        width="450" height="400"></object>
</body>
```

7-1.html

```
<body>
<embed src="test.swf" width="450" height="400">
</body>
```

7-2.html





# 7. 멀티미디어

## 7-2. 멀티미디어의 웹 표준화

- ✓ HTML5에서 추가된 멀티미디어 요소인 **video**와 **audio**는 브라우저 자체에서 동영상을 재생하도록 지원하는 멀티미디어 태그.

형식	브라우저					설명
	IE (Ver 9 이상)	chrome (Ver 6 이상)	firefox (Ver 3.6 이상)	safari (Ver 5 이상)	opera (Ver 11.6 이상)	
video						고화질 영상을 지원해 많은 사이트에서 사용. 라이센스가 있으므로 웹 용을 제외하면 유료.
	O	O	O	O	O	화질이 우수하고 무료로 제공. 최근 사용 증가.
	X	O	O	X	O	화질은 다른 유형보다 떨어지지만 무료이기 때문에 webm이 나오기 전까지 많이 사용.
audio	mp3	O	O	O	O	대부분의 음원에서 사용. 라이선스 때문에 유료.
	ogg	X	O	O	X	무료이므로 게임 등에서 많이 사용.

[브라우저별 video/audio codec 지원 현황](#)



## 7. 멀티미디어

### 7-3. audio & video 재생 – audio

- ✓ HTML5에서 배경 음악이나 효과음 등을 사용 시 `<audio>` tag 이용.
- ✓ 일반적으로 대부분의 브라우저에서는 mp3을 삽입하며, ogg(or ogv) 파일은 함께 지정.
- ✓ `<audio src="오디오 파일 경로" [속성] [속성값]></audio>`

속성	설명
autoplay	오디오를 자동 재생.
controls	웹 화면에 컨트롤 막대를 표시. 재생/멈춤, 진행 바, 볼륨 등이 표시.
loop	오디오를 반복 재생.
muted	오디오 재생해 진행 하지만 소리는 끔.
preload	재생 버튼을 눌러 재생하기 전에 오디오 파일을 다운로드해 준비.

`<audio>` 속성

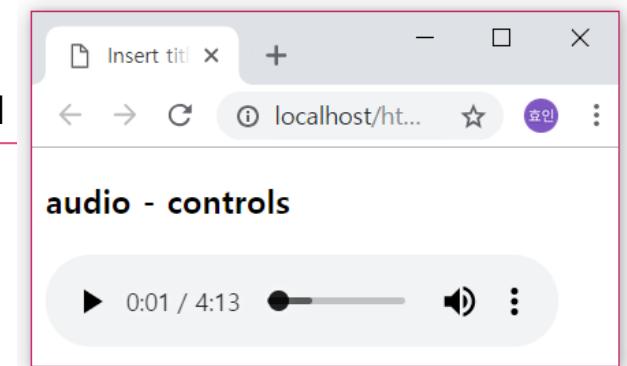


# 7. 멀티미디어

## 7-3. audio & video 재생 – audio

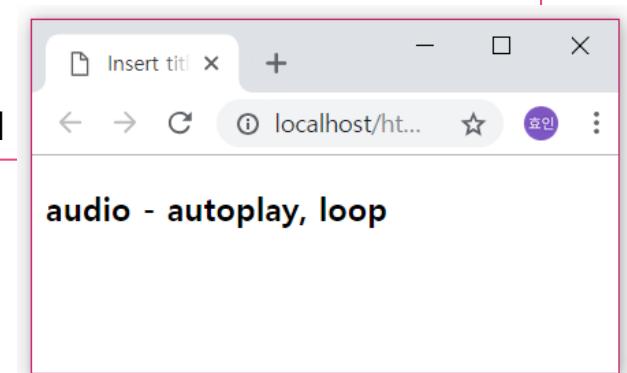
```
<body>
<h3>audio - controls</h3>
<audio src="music.mp3" controls="controls"></audio>
</body>
```

7-3.html



```
<body>
<h3>audio - autoplay, loop</h3>
<audio src="../media/music.mp3" autoplay="autoplay" loop="Loop"></audio>
</body>
```

7-4.html





## 7. 멀티미디어

---

### 7-3. audio & video 재생 – video

- ✓ 지금 까지 웹 페이지에 동영상을 삽입하고 싶을 때는 object Element 나 Embed Element 를 사용하여 Flash, media player 와 같은 플러그인을 이용 하여야 했음.
- ✓ HTML5에서는 플러그인 없이도 동영상을 웹 페이지에 쉽게 삽입.
- ✓ `<video src=“비디오 파일 경로” [속성] [속성=“속성값”]></video>`
- ✓ Video Element의 특징.
  - 재생, 일시중지 등 Web Browser 자체 Control 이 있다.
  - Source에 파일을 여러개를 지정할 수 있어서 Web Browser의 지원 Format 에 따라 표시 할 수 있다.

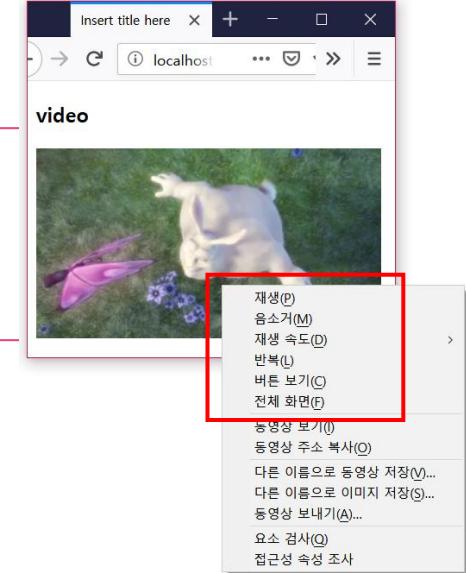
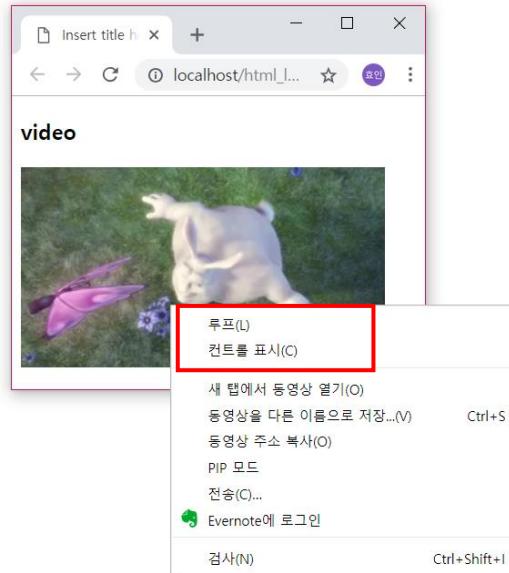


# 7. 멀티미디어

## 7-3. audio & video 재생 – video

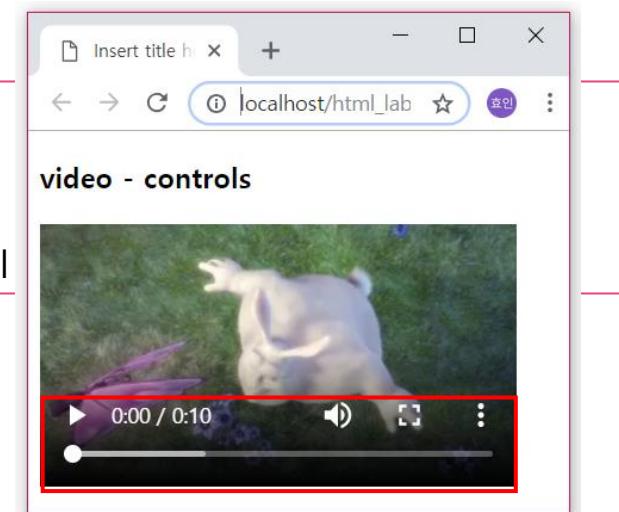
```
<body>
<h3>video</h3>
<video src="../media/video.mp4"></video>
</body>
```

7-5.html



```
<body>
<h3>video - controls</h3>
<video src="../media/video.mp4" controls="controls"></video>
</body>
```

7-6.html





## 7. 멀티미디어

### 7-3. audio & video 재생 – video

- ✓ 여러 미디어 파일 한꺼번에 지정.
- ✓ 브라우저마다 지원하는 오디오 또는 비디오 코덱이 다르기 때문에 한가지 미디어만 지정할 경우, 일부 브라우저에서는 지원하지 않을 수 있다.
- ✓ **<source src="미디어 파일 경로" type="미디어 유형" codecs="코덱">**

```
<body>
<h3>video</h3>
<video controls="controls">
    <source src="../media/video.mp4" type="video/mp4">
    <source src="../media/video.webm" type="video/webm">
    <source src="../media/video.ogv" type="video/ogg">
        이 영상을 보기 위해서는 HTML5를 지원하는 브라우저가 필요합니다. --- HTML5 미지원 브라우저용.
</video>
or
<video controls="controls">
    <source src="../media/video.mp4" type="video/mp4">
    <source src="../media/video.webm" type="video/webm">
    <source src="../media/video.ogv" type="video/ogg">
        <object data="../media/video.swf" type="application/x-shockwave-flash"></object>
</video>
</body>
```



# 7. 멀티미디어

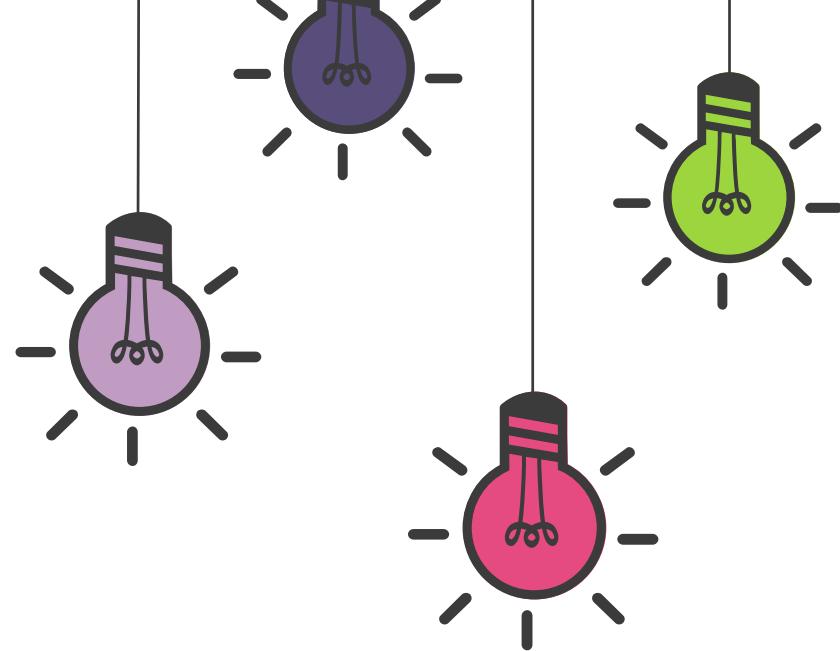
## 7-3. audio & video 재생 – video

- ✓ **video, audio 공통 속성.**

속성	설명
autoplay	비디오나 오디오 파일을 다운로드 하자마자 자동 재생. HTML5 권장 X.
controls	웹 화면에 컨트롤 막대를 표시. 재생/멈춤, 진행 바, 볼륨 등이 표시.
loop	미디어를 반복 재생.
muted	비디오를 재생 할 때 소리는 끄고 화면만 재생.
preload	재생 버튼을 눌러 재생하기 전에 오디오 파일을 다운로드해 준비.
poster	브라우저 또는 인터넷 문제로 재생이 불가능 할 경우 화면자리에 이미지로 대체. <video src="video file" controls poster="image file">
width, height	비디오 크기 조절.

preload 속성 값

속성 값	설명
none	미디어 파일을 미리 다운로드하지 않고 사용자가 재생 버튼을 눌러야 다운로드 시작.
metadata	미디어 파일을 즉시 사용하지 않을 것이라고 생각해 미디어 파일 전체를 다운로드하지 않고 메타 정보만 다운로드.
auto	사용자가 즉시 이용할 수 있도록 웹 문서를 로드할 때 미디어 파일도 모두 다운로드. 다운로드가 끝나도 자동 재생은 아니며, 속성값을 지정하지 않고 preload라고만 설정할 경우 기본이 auto값이다.



THANK YOU FOR WATCHING

감사합니다



안효인  
troment@nate.com