

출석수업

# HTML5 웹 프로그래밍

방송대 컴퓨터과학과  
ZOOM강의 : 은 인 기



## 수업 범위

1강. 웹 프로그래밍 개요  
& HTML 요소 : 텍스트

2강. HTML 요소  
: 리스트, 이미지, 하이퍼링크, 테이블

3강. HTML 요소  
: 미디어, 시멘틱 요소

5강. CSS  
: CSS의 개요, CSS 적용 방법

평가. 과제물 설명



HTML5 Web Programming

1강

# 웹 프로그래밍 개요 & HTML 요소: 텍스트

>>>>

- 
- 컴퓨터과학과 이관용 교수

# 학습목차

01 / 웹 개요

02 / 웹 프로그래밍 과정

03 / HTML 요소: 텍스트 표현



# 01

/ HTML5 Web Programming

## 웹 개요



# 인터넷?, 웹?

**INTERNET** derived by ARPANET

- INTERconnected NETwork
- 전 세계적으로 연결되어 있는 컴퓨터 네트워크 시스템

**World Wide Web("웹")** by Tim Berners-Lee

- 하이퍼링크 개념과 GUI를 사용하여 다양한 형태의 데이터와 정보에 접근할 수 있는 인터넷 서비스
- HTML로 작성된 웹 문서를 HTTP 프로토콜 기반으로 브라우저를 통해서 내려 받아서 읽을 수 있도록 구성됨

➡ '인터넷' ≠ '웹'

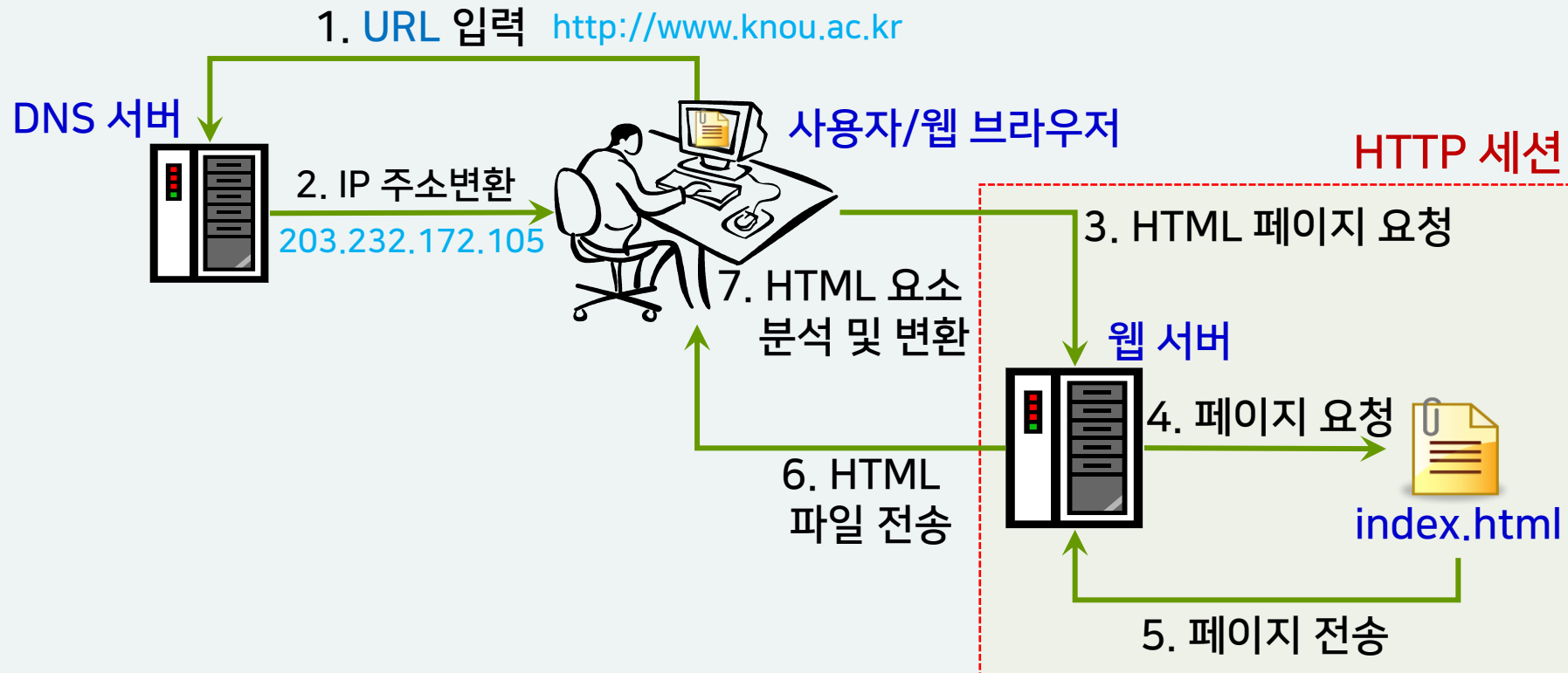
# 기본 용어

## ✓ 웹 관련 용어

- ① 웹 사이트 → 웹 페이지들의 의미 있는 묶음
- ① 웹 문서("HTML 문서") → 브라우저로 볼 수 있는 형식의 문서
- ① 웹 페이지 → 각 브라우저 창에 표시되는 개별적인 문서
  - 하나의 웹 문서는 주제/내용에 따라 여러 웹 페이지 파일로 구분/작성
- ① 하이퍼텍스트 → 단위 텍스트들과 이를 결합하는 하이퍼링크로 구성된 텍스트
- ① 웹 서버 → 웹 서비스를 제공하는 컴퓨터 시스템
- ① 웹 클라이언트 → 웹 서비스를 제공받아 사용하는 사용자/시스템
  - 웹은 서버-클라이언트 구조에서 동작

# 기본 용어

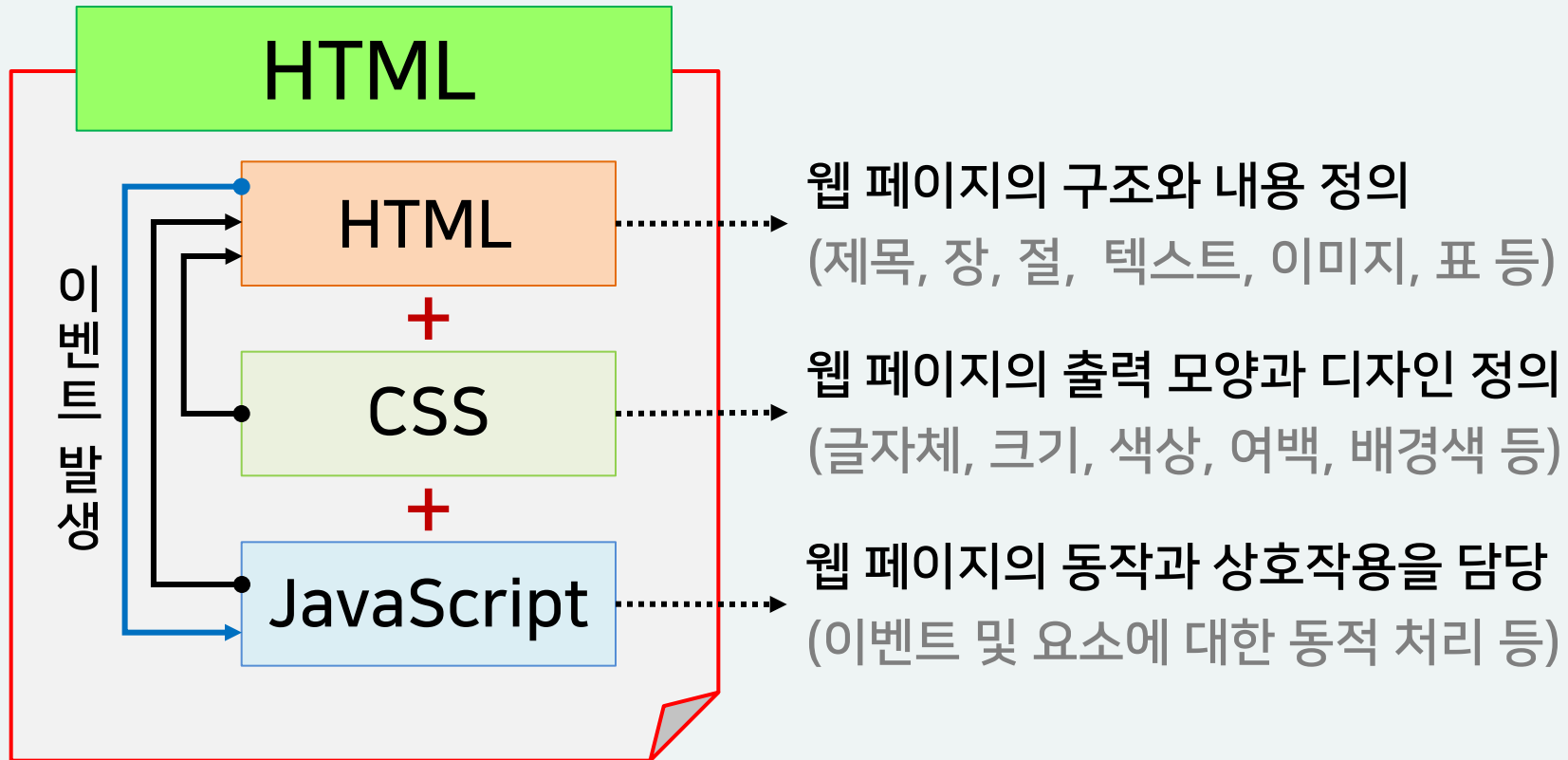
- ② HTTP → 웹 서버와 웹 클라이언트 간의 필요한 데이터를 주고 받기 위한 통신 규약





# 웹 페이지의 구성

✓ HTML을 기반으로 웹 페이지를 제작한다는 것은?



# 02

/ HTML5 Web Programming

## 웹 프로그래밍 과정



# 웹 프로그래밍?

## ✓ 웹 사이트 또는 웹 페이지를 만드는 과정

### ① Front-end 프로그래밍("클라이언트 사이드 프로그래밍")

- HTML+CSS+JavaScript → 모든 화면의 페이지 제작
- 프레임워크 → React, Angular, Vue.js 등

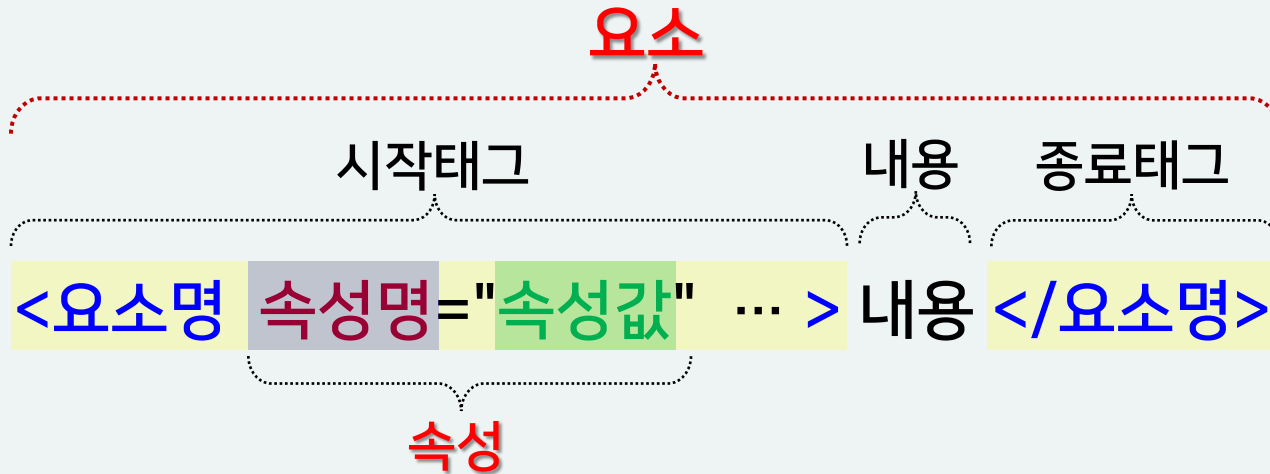
학습 대상

### ② Back-end 프로그래밍("서버 사이드 프로그래밍")

- 클라이언트에서의 동작/작업의 원활한 수행을 위한 처리 담당
- C/C++, Java, Python, JSP, Perl, PHP 등의 언어 사용
- 프레임워크 → Node.js, Django, Spring 등

# HTML 문서의 구성 요소

✓ HTML 문서 → “다양한 요소<sup>element</sup>들의 조합”



<p>HTML5</p>

<a href="http://cs.knou.ac.kr" target=\_blank>컴퓨터과학과</a>

종료태그 없이 사용 → <br>, <img> 등

└───────────> <br></br>    <br/>

# HTML 문서 작성 시 주의사항

- ✓ 요소명과 속성명은 대소문자를 구분하지 않음
  - <TITLE> = <Title> = <title> → 소문자 표기를 권고
- ✓ 파일명 지정 → '파일명.htm' 또는 '파일명.html'
- ✓ 화이트스페이스는 입력한 의도대로 적용되지 않음
  - 두 칸의 이상의 연속된 공백은 하나의 공백으로 처리
- ✓ 요소 안에 다른 요소 포함 가능 → 포함 관계 주의
  - <요소명1> ... <요소명2> ~~ </요소명2> ... </요소명1>

# HTML 문서의 기본 구조

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

문서 제목(title), 메타 데이터(meta),  
자바스크립트(script), CSS(style, link) 등

```
</head>
```

```
<body>
```

문서의 본문 내용(텍스트, 이미지, 링크, 테이블 등)

```
<!-- 자바스크립트 코드 위치 가능 -->
```

```
</body>
```

```
</html>
```

# HTML 문서의 기본 구조

## ✓ <!DOCTYPE html>

- 브라우저에게 페이지를 올바르게 표시할 수 있도록 문서 형식을 알려주는 지시어
  - 반드시 첫 줄에 위치

## ✓ <html> ... </html>

- 웹 페이지의 루트 요소 → HTML 문서의 시작과 끝을 알려 줌

## ✓ <head> ... </head>

- 웹 문서에 대한 메타 정보를 포함 → 화면에 표시되지 않음
  - 메타 요소 → title, link, style, script, meta, base

# HTML 문서의 기본 구조

✓ `<body> ... </body>`

- 브라우저 화면에 실제로 출력해서 보여주려는 문서의 본문에 해당하는 모든 내용을 포함하는 부분

✓ 주석문: `<!-- 주석 내용 -->`

- 문서 소스 내에 간단한 설명을 추가할 때 사용
  - 브라우저에 출력되지 않으므로 소스에서만 확인 가능
  - 문서의 어느 곳에든 위치 가능
  - '`<!--`' 와 '`-->`'는 하나의 문자 → 각 문자 사이에 공백 불가
  - 주석 내용으로 `<`, `>`, `--`는 사용하지 말아야 함



# 헤드 부분에 포함되는 메타 요소

## ✓ title 요소

- 브라우저 상단(탭)에 문서의 제목을 표시할 때 사용
  - 즐겨찾기 제목, 검색 엔진의 출력된 결과 페이지 제목으로도 사용

## ✓ link 요소

- CSS 스타일 파일(\*.css)을 가져와 웹 문서에 적용할 때 사용
  - 문서의 body 부분에 포함 가능

```
<link rel="stylesheet" href="파일명.css">
```

# 헤드 부분에 포함되는 메타 요소

## ✓ style 요소

- 웹 문서 전체에 적용할 CSS 스타일을 정의할 때 사용

```
<style>  
  CSS 스타일 정의 (예: h1 { color: red; })  
</style>
```

## ✓ script 요소

- 자바스크립트 코드를 작성(또는 외부 코드 링크)할 때 사용

```
<script>  
  자바스크립트 코드  
</script>
```

\* body 부분에 포함 가능

```
<script src="파일명.js"></script>
```

# 헤드 부분에 포함되는 메타 요소

## ✓ meta 요소

- 🕒 웹 문서에 대한 다양한 메타 데이터를 나타낼 때 사용

문자 인코딩 방식: `<meta charset="UTF-8">`

내용 설명: `<meta name="description" content="문서 설명">`

키 워 드: `<meta name="keywords" content="키워드1, 키워드2, ...">`

저 작 자: `<meta name="author" content="문서작성자">`

# 헤드 부분에 포함되는 메타 요소

## ✓ base 요소

### ➤ 웹 문서의 기본 URL 경로 지정

- body 요소 내의 상대 경로로 표시된 URL을 절대 경로로 취급

```
<base href="http://cs.knou.ac.kr/html5">
```

```
<a href="ex.html"> → <a href="http://cs.knou.ac.kr/html5/ex.html">
```

### ➤ 웹 페이지가 출력될 타깃 윈도우 지정

```
<base target="_blank">
```

```
<a href="ex.html"> → 연결된 문서가 새로운 창에서 열림
```

# HTML 문서의 작성과 실행

문서 작성을 위한 텍스트 편집기

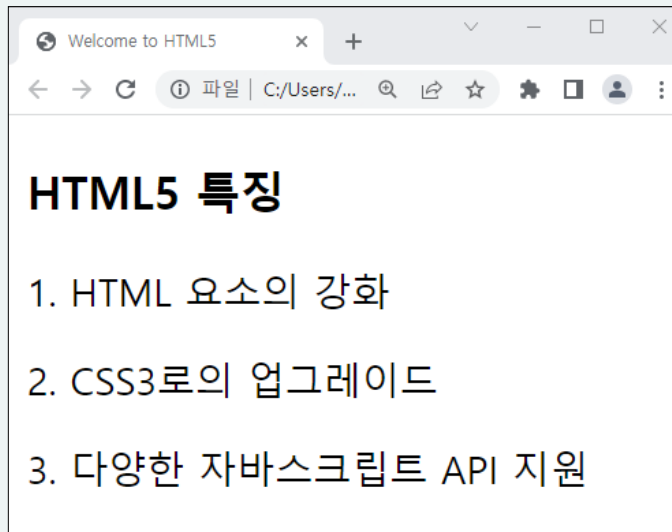


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Welcome to HTML5</title>
  </head>
  <body>
    <h3> HTML5 특징 </h3>
    <p>1. HTML 요소의 강화 </p>
    <p>2. CSS3로의 업그레이드 </p>
    <p>3. 다양한 자바스크립트 API 지원 </p>
  </body>
</html>
```

UltraEdit, EditPlus, Eclipse, Sublime, ...

드림위버, 나모 웹에디터, ...

문서 실행을 위한 웹 브라우저



# 03

/ HTML5 Web Programming

## HTML 요소: 텍스트 표현



# 줄을 바꾸는 **br** 요소

## ✓ 편집기에서 엔터기와 같은 기능

### ⦿ 종료태그 </br> 없이 사용

```
<body>
과목명: HTML5 웹 프로그래밍 <br/>
<br/>
<br/>
개설학과: 컴퓨터과학과 <br/>
개설학년: 2학년 <br/>
<br/>
담당교수: 이관용
</body>
```

```
과목명: HTML5 웹 프로그래밍 ↵
↵
↵
개설학과: 컴퓨터과학과 ↵
개설학년: 2학년 ↵
↵
담당교수: 이관용
```

br 요소를 사용하지 않은 경우

과목명: HTML5 웹 프로그래밍 개설학과: 컴퓨터과학과 개설학년: 2학년 담당교수: 이관용

# 단락을 나누는 p 요소

## ✓ 문단을 나누어 독립된 단락을 생성

- 단락 구분을 위해 단락 전후에 빈 줄이 자동으로 추가됨

<body>

독립된 단락을 생성하기 위해서는

<p> p 요소를 사용하며, </p>

<p><p><p> p 요소를 연속해서 여러 개 사용해도  
한 번 사용한 것과 같은 효과를 가짐 </p> </p> </p>

단락의 정렬은 CSS를 사용한다.

</body>

독립된 단락을 생성하기 위해서는

p 요소를 사용하며,

p 요소를 연속해서 여러 개 사용해도 한 번 사용한 것과 같은 효과를 가짐

단락의 정렬은 CSS를 사용한다.



# 수평선을 그리는 **hr** 요소

- ☑ 텍스트 사이에 수평선을 그림
  - 주제가 변경되는 콘텐츠를 구분하는 용도
- ⌚ 자동 줄바꿈, 종료태그 없이 사용

```
<body>
```

```
  hr 요소는 수평선을 그릴 때 사용한다.
```

```
  <hr/>
```

```
  의미론적으로는 단락 단위로 주제를 구분할 때 사용한다.
```

```
  <hr/>
```

```
</body>
```

```
hr 요소는 수평선을 그릴 때 사용한다.
```

```
의미론적으로는 단락 단위로 주제를 구분할 때 사용한다.
```

# 단락의 제목을 지정하는 **h**<sub>n</sub> 요소

✓  $n=1, \dots, 6 \rightarrow h_1, h_2, h_3, h_4, h_5, h_6$

⌚ 제목 크기  $\rightarrow h_1 > h_2 > h_3 > h_4 > h_5 > h_6$

⌚ 굵은 글자(볼드체), 자동 줄바꿈

<body>

<h1>h1로 제목을 지정한 경우</h1>

<h2>h2로 제목을 지정한 경우</h2>

<h3>h3으로 제목을 지정한 경우</h3>

<h4>h4로 제목을 지정한 경우</h4>

<h5>h5로 제목을 지정한 경우</h5>

<h6>h6으로 제목을 지정한 경우</h6>

</body>

**h1로 제목을 지정한 경우**

**h2로 제목을 지정한 경우**

h3으로 제목을 지정우

h4로 제목을 지정한 경우

h5로 제목을 지정한 경우

h6으로 제목을 지정한 경우

# 텍스트를 자유롭게 표시하는 pre 요소

## ✓ 편집기에서 문자가 입력된 형태를 그대로 유지하여 출력

<body>

pre 요소 설명

<pre>

- <b>pre 요소는 'preformatted text'의 약자이다.</b>
- 편집기에서작성된 코드의 형태를 그대로 유지하여 출력한다.

-- 공백, 줄바꿈 등을 자유롭게 입력할 수 있다.

- 텍스트는 고정폭,  
행의 길이가 브라우저의 폭 보다 길어져도 자동 줄바꿈이 되지 않음

- 특수 문자가 많은 텍스트(예, 프로그램 소스)를 표시할 때 유용

</pre>

</body>

pre 요소 설명

- pre 요소는 'preformatted text'의 약자이다.
- 편집기에서작성된 코드의 형태를 그대로 유지하여 출력한다.
  - 공백, 줄바꿈 등을 자유롭게 입력할 수 있다.
- 텍스트는 고정폭,  
행의 길이가 브라우저의 폭 보다 길어져도 자동 줄바꿈이 되지 않음
- 특수 문자가 많은 텍스트(예, 프로그램 소스)를 표시할 때 유용

# 인용문을 작성하는 **blockquote** 요소

## ☑ 상하 빈 줄로 구분되는 인용문 단락의 생성 및 들여쓰기

- ⌚ 들여쓰기 간격의 인위적인 조정은 불가
- ⌚ cite 속성 → 인용된 부분의 URL 지정

<body>

인생은 내일도 계속된다.

<blockquote cite="http://www.heartsaying.com">

좋지 않는 일은 심플하게 생각하고 <br/>

좋은 일은 복잡하게 자꾸 꼬집어 내자!

</blockquote>

- &quot;한 번쯤은 위로 받고 싶은 나&quot; 중에서-

</body>

인생은 내일도 계속된다.

좋지 않는 일은 심플하게 생각하고  
좋은 일은 복잡하게 자꾸 꼬집어 내자!

- "한 번쯤은 위로 받고 싶은 나" 중에서-

# 공백 및 특수 문자 입력

✓ 두 칸 이상의 연속된 공백, 예약어로 사용되는 문자,  
키보드로 직접 입력할 수 없는 문자를 입력하는 방법

① 연속된 여러 개의 공백

② <, >, ", &

③ ©, →, ♥, 👉, ...



&특수이름;

&#10진수코드;

&#x16진수코드;

# 공백 및 특수 문자 입력

문자	특수 이름	10진수 코드	16진수 코드
<	&lt;	&#60;	&#x3c;
>	&gt;	&#62;	&#x3e;
&	&amp;	&#38;	&#x26;
"	&quot;	&#34;	&#x22;
공백	&nbsp;	&#32;	&#x20;
©	&copy;	&#169;	&#xa9;
→	&rarr;	&#8594;	&#x2192;
♡			&#x2661;
☞			&#x261e;

# 공백 및 특수 문자 입력

<body>

줄바꿈 요소 &rarr; &lt;br&gt; <br/>

&nbsp; 문단 구분 요소 &rarr; &#60;p&#62; <br/>

&nbsp;&nbsp; 공백 문자 &rarr; &amp;nbsp; <br/>

&nbsp;&nbsp;&nbsp; 키보드로 입력 불가 문자

&rarr; &copy;, &#x2661;, &#x261e; 등

</body>

줄바꿈 요소 → <br>

문단 구분 요소 → <p>

공백 문자 → &nbsp;

키보드로 입력 불가 문자 → ©, ♥, 👁 등

# 글자 스타일 지정 요소

## ✓ 물리적 스타일 관련 요소

- 단순히 웹 브라우저에 표시되는 출력 모양만을 지정하는 요소

**b**, *i*, ~~s~~, u, <sup>sup</sup>, <sub>sub</sub>, small

## ✓ 논리적 스타일 관련 요소

- 출력 형태보다는 요소 자체가 의미를 갖고 사용되는 요소

*cite*, code, samp, *var*, *dfn*, *em*, **strong**,  
abbr, *address* ↴, kbd, "q", mark



# 물리적 스타일 관련 요소

<body>

&lt;b&gt; : **<b>**진하게 표시</b> <br/>

&lt;i&gt; : *<i>*기울어진 형태(이탤릭체)로 표시</i> <br/>

&lt;s&gt; : ~~<s>~~글자 가운데를 통과하는 줄(취소선)을 표시</s> <br/>

&lt;u&gt; : <u>밑줄을 표시</u> <br/>

&lt;sup&gt; : 글자를 <sup><sup></sup>위첨자</sup>로 표시 <br/>

&lt;sub&gt; : 글자를 <sub><sub></sub>아래첨자</sub>로 표시 <br/>

&lt;small&gt; : 기본 글자의 크기보다 <small>작은 크기의 글자</small>로 표시

</body>

<b> : 진하게 표시

<i> : 기울어진 형태(이탤릭체)로 표시

<s> : ~~글자 가운데를 통과하는 줄(취소선)을 표시~~

<u> : 밑줄을 표시

<sup> : 글자를 위첨자로 표시

<sub> : 글자를 아래첨자로 표시

<small> : 기본 글자의 크기보다 작은 크기의 글자로 표시

# 물리적 스타일 관련 요소

## ✓ 적용 의미

요소	용도
<b>b</b>	강조의 의미는 사라지고 주목해야 할 단어를 표시
<i>i</i>	보통의 서술과 구분되는 다른 어조/분위기를 갖는 텍스트를 표시
s	더 이상 정확하지 않거나 관련이 없는 내용을 표시
<u>u</u>	철자가 틀린 단어나 불충분한 내용을 표시
small	작게 표시해야 하는 내용(부수적인 해설이나 이용조건, 법적 공지 등)

# 논리적 스타일 관련 요소

<body>

&lt;cite&gt; : <cite>책, 시, 그림, 영화 등의 작품 제목을 표시</cite> <br/>

&lt;code&gt; : <code>컴퓨터 코드를 표시</code> <br/>

&lt;samp&gt; : <samp>프로그램의 샘플 출력을 표시</samp> <br/>

&lt;var&gt; : <var>변수를 표시</var> <br/>

&lt;dfn&gt; : <dfn>용어에 대한 정의를 표시</dfn> <br/>

&lt;em&gt; : <em>강조하는 내용을 표시</em> <br/>

&lt;strong&gt; : <strong>중요한 내용을 표시</strong> <br/>

&lt;abbr&gt; : <abbr>축약형을 표시</abbr> <br/>

&lt;address&gt; : <address>실제 우편물 주소를 표시</address> <br/>

&lt;kbd&gt; : <kbd>키보드로 입력한 내용임을 표시</kbd> <br/>

&lt;q&gt; : <q>짧은 인용구를 표시</q> <br/>

&lt;mark&gt; : <mark>시각적/의미적으로 특정 문구/단어 강조</mark>

</body>

<cite> : 책, 시, 그림, 영화 등의 작품 제목을 표시

<code> : 컴퓨터 코드를 표시

<samp> : 프로그램의 샘플 출력을 표시

<var> : 변수를 표시

<dfn> : 용어에 대한 정의를 표시

<em> : 강조하는 내용을 표시

<strong> : 중요한 내용을 표시

<abbr> : 축약형을 표시

<address> : ↵

실제 우편물 주소를 표시 ↵

<kbd> : 키보드로 입력한 내용임을 표시

<q> : “짧은 인용구를 표시”

<mark> : 시각적/의미적으로 특정 문구/단어 강조

# 블록 요소 vs 인라인 요소

블록 요소	인라인 요소
문서 구조 요소	텍스트 요소
항상 새로운 줄에서 시작해서 하나의 줄을 전부 차지해서 내용 표시	입력하는 내용만큼의 공간을 차지해서 내용 표시
자동으로 줄바꿈 수행	줄바꿈 미수행 → 앞뒤 내용이 이어져 한 줄에 꼭 표시
<code>div</code> , <code>p</code> , <code>h1</code> , ...	<code>span</code> , <code>img</code> , <code>strong</code> , ...

# 콘텐츠 컨테이너 요소: **div**, **span**

div 요소	span 요소
블록 요소	인라인 요소
HTML 요소를 묶어 하나의 논리적인 단위의 영역을 구성	텍스트를 위한 컨테이너
논리적으로 구분된 대량의 내용에 대해 CSS 스타일 지정 또는 전체 페이지의 공간을 분할하여 레이아웃을 구성하는 데 사용	텍스트의 일부에 대해 CSS 스타일 지정 또는 자바스크립트에서 조작할 때 유용

# 콘텐츠 컨테이너 요소: **div, span**

```
<style>
```

```
.color1 { background-color: lightgreen }
```

```
.color2 { height: 50px; background-color: yellow }
```

```
</style>
```

```
<body>
```

```
<div class="color1">
```

div 요소의 특징1 : 한 줄 공간을 제공

```
</div>
```

```
<div class="color2">
```

div 요소의 특징2: 앞뒤로 자동으로 줄바꿈이 수행됨

```
</div>
```

```
<hr/>
```

span 요소는 

```
<span class="color1">
```

 입력하는 내용만큼의 공간을 차지하며, 

```
</span>
```

```
<span class="color2">
```

 줄바꿈이 없어서 앞뒤 내용이 이어져서 나타난다. 

```
</span>
```

```
</body>
```

div 요소의 특징1 : 한 줄 공간을 제공

div 요소의 특징2: 앞뒤로 자동으로 줄바꿈이 수행됨

span 요소는 입력하는 내용만큼의 공간을 차지하며,  
줄바꿈이 없어서 앞뒤 내용이 이어져서 나타난다.



Programming



CSS



HTML5



JavaScript

HTML5 Web Programming

2강

# HTML 요소:

## 리스트, 이미지, 하이퍼링크, 테이블

>>>>

- 
- 컴퓨터과학과 이관용 교수

# 학습목차

01 / 리스트 표현

02 / 이미지 표현

03 / 하이퍼링크 표현

04 / 테이블 표현





# 01

/ HTML5 Web Programming

## 리스트 표현



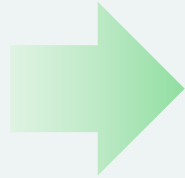
# 리스트의 종류

종류	설명
순서 없는 리스트 Unordered List	<ul style="list-style-type: none"><li>■ 글머리기호(•)를 붙여 리스트의 항목을 표시</li><li>■ 사용 요소 → ul ~ li</li></ul>
순서 있는 리스트 Ordered List	<ul style="list-style-type: none"><li>■ 순서(1,2,3,...)를 매겨서 리스트의 항목을 표시</li><li>■ 사용 요소 → ol ~ li</li></ul>
서술 리스트 Description List	<ul style="list-style-type: none"><li>■ 어떤 용어나 이름에 대해서 서술하는 형식</li><li>■ 사용 요소 → dl ~ dt/dd</li></ul>

# 순서 없는 리스트를 지정하는 ul 요소

- ✓ 각 항목 앞에 순서를 나타내지 않는 기호를 붙여서 표시

```
<ul>
  <li> 항목1 </li>
  <li> 항목2 </li>
  ...
  <li> 항목n </li>
</ul>
```



- 항목1
- 항목2
- ...
- 항목n

기호 변경 → CSS `list-style-type` 속성  
none, disc (●),  
circle (○), square (■)

- ✓ ul, li 요소 내에서 공통으로 사용되는 자식 요소
- ✓ 리스트에서 각 항목의 내용을 표시할 때 사용
  - 들여쓰기
  - 자동 줄바꿈
  - 종료태그 없이 사용 가능

# 순서 없는 리스트를 지정하는 **ul** 요소

```
<body>
```

```
<h3>리스트 종류</h3>
```

```
<ul>
```

```
<li> 순서 없는 리스트 &lt;ul&gt; ~ &lt;/ul&gt;
```

```
<li> 순서 있는 리스트 &lt;ol&gt; ~ &lt;/ol&gt;
```

```
<li> 서술 리스트 &lt;dl&gt; ~ &lt;/dl&gt;
```

```
</ul>
```

```
</body>
```

## 리스트 종류

- 순서 없는 리스트 <ul> ~ </ul>
- 순서 있는 리스트 <ol> ~ </ol>
- 서술 리스트 <dl> ~ </dl>

# 순서 있는 리스트를 지정하는 **ol** 요소

✓ 각 항목 앞에 순서를 나타내는 기호를 붙여서 표시

➡ 순서 있는 기호 → 숫자(기본값), 영문자, 로마자 등

<ol>

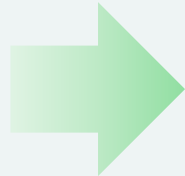
<li> 항목1 </li>

<li> 항목2 </li>

...

<li> 항목n </li>

</ol>



1. 항목1

2. 항목2

...

n. 항목n

# 순서 있는 리스트를 지정하는 **li** 요소

## ☑ **li** 요소의 속성

- `type="값"` → 순서가 부여된 기호의 유형 지정
- `start="숫자"` → 항목의 시작 번호 지정
- `reversed` → 항목 번호를 역순으로 지정

```
1 2 3 4 ... (기본)
a b c d ...
A B C D ...
i ii iii iv ...
I II III IV ...
```

## ☑ **li** 요소의 속성

- `value="숫자"` → 항목 번호를 중간에 바꿀 때 사용
  - `li` 요소에서만 사용

# 순서 있는 리스트를 지정하는 **ol** 요소

```
<body>
```

```
<h4>리스트의 종류</h4>
```

```
<ol reversed>
```

```
<li> 순서 없는 리스트 <ul> ~ </ul>
```

```
<li> 순서 있는 리스트 <ol> ~ </ol>
```

```
<li> 서술 리스트 <dl> ~ </dl>
```

```
</ol>
```

```
</body>
```

## 리스트의 종류

3. 순서 없는 리스트 <ul> ~ </ul>
2. 순서 있는 리스트 <ol> ~ </ol>
1. 서술 리스트 <dl> ~ </dl>



# 순서 있는 리스트를 지정하는 **ol** 요소

<body>

<h4><li> 요소의 특징</h4>

<ol type="a" start="2">

<li> List Item

<li> 순서 없는 리스트와 순서 있는 리스트에서 공통적으로 사용

<li value="6"> <li value="6">를 통해

항목 번호가 d가 아닌 6번째 알파벳(f)부터 시작하도록 재지정

<li> 들여쓰기 기능, 줄바꿈 기능 포함

<li> 종료태그 없이 시작태그만 사용 가능

</ol>

</body>

## <li> 요소의 특징

b. List Item

c. 순서 없는 리스트와 순서 있는 리스트에서 공통적으로 사용

f. <li value="6">를 통해 항목 번호가 d가 아닌 6번째 알파벳(f)부터 시작하도록 재지정

g. 들여쓰기 기능, 줄바꿈 기능 포함

h. 종료태그 없이 시작태그만 사용 가능

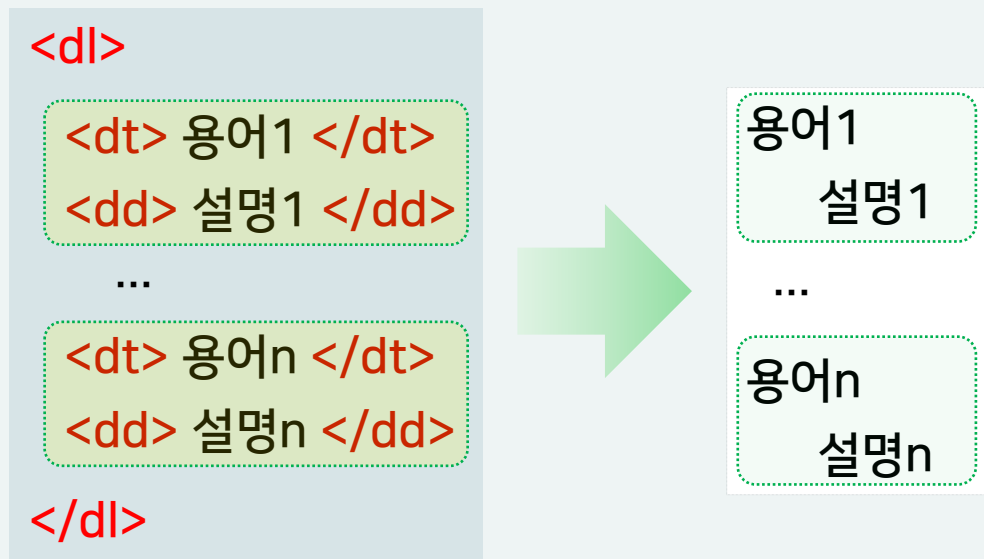
# 서술 리스트를 지정하는 **dl** 요소

✓ 용어/이름을 정의하고, 이에 대한 설명을 제공하는 리스트

➤ **dt** 요소 → 용어/이름을 나열. 자동 줄바꿈

➤ **dd** 요소 → 나열된 용어/이름에 대한 설명을 제공할 때 사용

- 자동 줄바꿈, 들여쓰기



# 서술 리스트를 지정하는 dl 요소

```
<body>
```

```
<h4>웹페이지를 작성하기 위해서는</h4>
```

```
<dl>
```

```
<dt>HTML</dt>
```

```
<dd>웹페이지의 구조와 내용 정의</dd>
```

```
<dt>CSS</dt>
```

```
<dd>웹페이지의 출력 모양과 디자인 정의</dd>
```

```
<dt>JavaScript</dt>
```

```
<dd>웹페이지의 동작과 동적 상호작용을 제공</dd>
```

```
</dl><br/>
```

이러한 부분들의 협업이 필요하다.

```
</body>
```

웹페이지를 작성하기 위해서는

HTML

웹페이지의 구조와 내용 정의

CSS

웹페이지의 출력 모양과 디자인 정의

JavaScript

웹페이지의 동작과 동적 상호작용을 제공

이러한 부분들의 협업이 필요하다.

# 02 / HTML5 Web Programming

## 이미지 표현



# 이미지를 삽입하는 **img** 요소

✓ 웹 페이지에 이미지를 삽입할 때 사용

⌚ 종료태그 없이 사용

✓ 속성

속성	설명
<code>src</code> ="파일의 URL"	(필수) 문서에 표시할 이미지 파일의 경로/이름
<code>width</code> ="숫자"	이미지의 폭(픽셀, %)
<code>height</code> ="숫자"	이미지의 높이(픽셀, %)
<code>alt</code> ="텍스트"	이미지가 표시되지 않을 때 대체 표시될 텍스트
<code>usemap</code> ="#맵이름"	클라이언트 측의 이미지맵

# 이미지를 삽입하는 **img** 요소의 속성

## ☑ **width, height** 속성

- ⌚ 이미지의 폭과 높이 지정 → 기본 크기 = 원본 크기
- ⌚ 값 → 픽셀, % (브라우저 화면 크기에 대한 상대적인 크기)
- ⌚ 폭과 높이 중 하나의 크기만 지정되면  
→ 나머지 하나의 값은 원본 크기에 비례해서 자동 지정

## ☑ **alt** 속성

- ⌚ 이미지에 대해 대체 표시될 텍스트 지정
  - 이미지가 화면에 표시되지 않거나 깨지는 경우를 대비해서 사용

# 이미지를 삽입하는 **img** 요소

```
<body>
```

```
<br/>
```

```

```

```

```

```

```

```

```

```
</body>
```



방송대 로고입니다.

# 03

/ HTML5 Web Programming

## 하이퍼링크 표현





# 하이퍼링크를 삽입하는 a 요소

- ✓ 링크의 클릭을 통해 특정한 다른 위치로 이동할 때 사용

## 기본 형식

`<a href="이동할 위치">링크를 걸어줄 내용물(텍스트, 이미지)</a>`

- ✓ 속성

속성	설명
<code>href="URL"</code>	링크를 통해 이동하려는 곳의 경로/주소
<code>target="창이름"</code>	링크를 클릭하였을 때 링크된 내용이 출력될 창 지정 • 값 → <code>_blank</code> (새로운 창), <code>_self</code> (기본값, 현재창), <code>창이름</code> 등
<code>download</code>	href 속성에 지정된 파일을 다운로드해서 저장

# a 요소의 기본적인 사용

<body>

<a href="http://www.knou.ac.kr">

한국방송통신대학교 (현재창)</a>

<br/><br/>

<a href="http://cs.knou.ac.kr"

target="\_blank">방송대 컴퓨터과학과 (새로운창)</a>

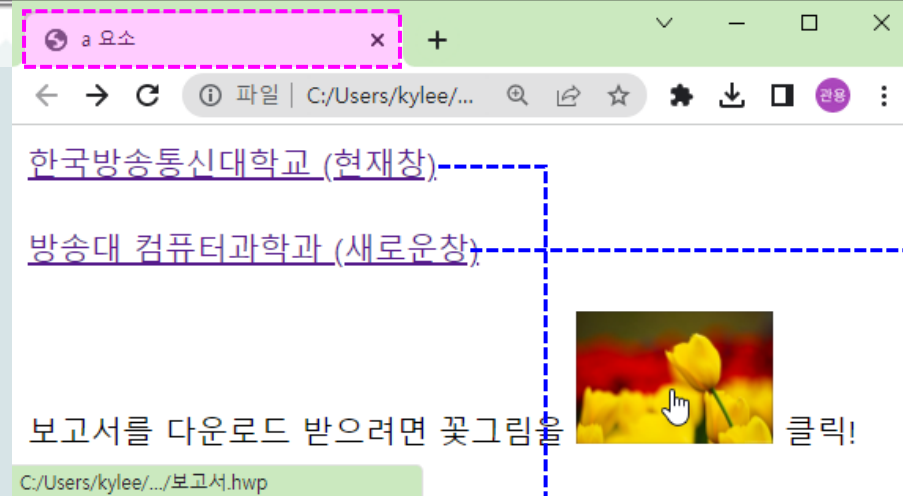
<br/><br/>

보고서를 다운로드 받으려면 꽃그림을

<a href="보고서.hwp" download>

</a> 클릭!

</body>



# id 속성을 활용한 특정 위치로의 이동

## ☑ 동일/다른 페이지의 특정 위치로 이동하는 경우

### 링크 부분

- 같은 페이지 내에서 이동하는 경우  
`<a href="#이동할 특정 위치의 이름"> 내용 </a>`
- 다른 페이지의 특정 위치로 이동하는 경우  
`<a href="페이지경로#이동할 특정 위치의 이름"> 내용 </a>`

### 이동할 특정 위치

`<a id="이동할 특정 위치의 이름"> 내용 </a>`

# id 속성을 활용한 특정 위치로의 이동

```
<body>
  <h3>목차</h3>
  <a href="#html">HTML이란 무엇인가?</a><br/>
  <a href="#css">CSS란 무엇인가?</a><br/>
  <a href="#js">JavaScript란 무엇인가?</a><br/>
  <a href="html5소개.html#html5특징">
    HTML5 특징 (다른 페이지로 이동)</a> <br/><br/>
  <hr> <br/><br/><br/><br/>
  <a id="html"> HTML</a>
  <p>HyperText Markup Language</p>
  <br/><br/><br/><br/><br/><br/><br/>
  <a id="css">CSS</a>
  <p>Cascading Style Sheet</p>
  <br/><br/><br/><br/><br/><br/><br/>
  <a id="js">JavaScript</a>
  <p>클라이언트 쪽에서 독립적으로 실행되는
    프로그램을 작성하기 위한 스크립트 언어</p>
  <br/><br/><br/><br/>
  <a href="#top">맨 위로</a>
</body>
```

```
<body>
  <h3>HTML5</h3>
  <p>HTML5는 차세대 웹문서 표준으로 W3C에서 채택되었다.</p>
  <br/>
  <a id="html5특징">HTML5 특징</a><br>
  <ol>
    <li>강화된 마크업 요소 </li>
    <li>CSS3의 지원 </li>
    <li>자바스크립트를 통한 다양한 API 지원 </li>
    <li>모바일 웹 환경 고려 </li>
  </ol>
</body>
```

## 목차

[HTML이란 무엇인가?](#)  
[CSS란 무엇인가?](#)  
[JavaScript란 무엇인가?](#)  
[HTML5 특징 \(다른 페이지로 이동\)](#)

HTML

HyperText Markup Language

CSS

Cascading Style Sheet

JavaScript

클라이언트 쪽에서 독립적으로 실행되는 프로그램을 작성하기 위한  
스크립트 언어

[맨 위로](#)

# id 속성을 활용한 특정 위치로의 이동

The diagram illustrates the use of the `id` attribute for navigation. On the left, a table of contents (목차) lists sections: HTML, CSS, and JavaScript, each with a link to a specific `id`. On the right, the corresponding content sections are shown, each with a unique `id` attribute. Red dashed arrows indicate the navigation path from the links in the table of contents to the specific sections.

**목차**

- [HTML이란 무엇인가?](#)
- [CSS란 무엇인가?](#)
- [JavaScript란 무엇인가?](#)
- [HTML5 특징 \(다른 페이지로 이동\)](#)

**HTML**

HyperText Markup Language

**CSS**

Cascading Style Sheet

**JavaScript**

클라이언트 쪽에서 독립적으로 실행되는 프로그램을 작성하기 위한 스크립트 언어

**HTML5 특징**

1. 강화된 마크업 요소
2. CSS3의 지원
3. 자바스크립트를 통한 다양한 API 지원
4. 모바일 웹 환경 고려

**CSS**

Cascading Style Sheet

**JavaScript**

클라이언트 쪽에서 독립적으로 실행되는 프로그램을 작성하기 위한 스크립트 언어

**HTML5 특징**

HTML5는 차세대 웹문서 표준으로 W3C에서 채택되었다.

**id 속성을 활용한 특정 위치로의 이동**

목차

- [HTML이란 무엇인가?](#)
- [CSS란 무엇인가?](#)
- [JavaScript란 무엇인가?](#)
- [HTML5 특징 \(다른 페이지로 이동\)](#)

HTML

HyperText Markup Language

CSS

Cascading Style Sheet

JavaScript

클라이언트 쪽에서 독립적으로 실행되는 프로그램을 작성하기 위한 스크립트 언어

맨 위로

# 다양한 형태의 링크 지정

궁금한 사항이 있으면 관리자에게 메일을 보내 주세요.

`<p><a href="mailto:admin@knou.ac.kr">이메일 보내기</a></p><hr/>`

자바스크립트 코드 실행

`<p><a href="javascript:alert('Hello! World!');">자바스크립트 실행</a></p><hr/>`

음악 파일 연결

`<p><a href="song.mp3" title="Military March Op.51">군대행진곡</a></p><hr/>`

동영상 파일 연결

`<p><a href="golf.wmv" title="Tiger Woods">골프</a></p><hr/>`

압축 파일(실행 파일) 다운로드

`<p><a href="files.zip" title="zip 파일">다운로드</a></p>`

# 다양한 형태의 링크 지정

궁금한 사항이 있으면 관리자에게 메일을 보내 주세요.

[이메일 보내기](#)

자바스크립트 코드 실행

[자바스크립트 실행](#)

음악 파일 연결

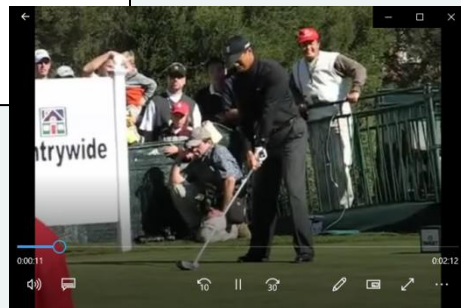
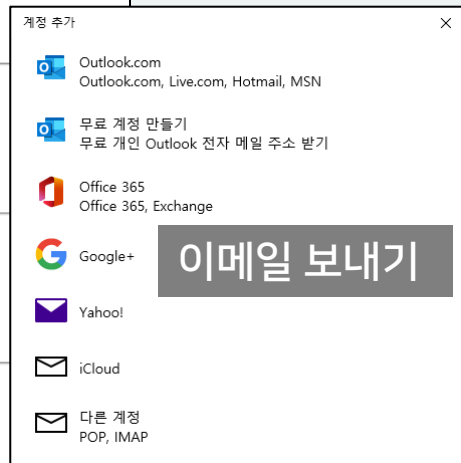
[군대행진곡](#)

동영상 파일 연결

[골프](#)

압축 파일(실행 파일) 다운로드

[다운로드](#)

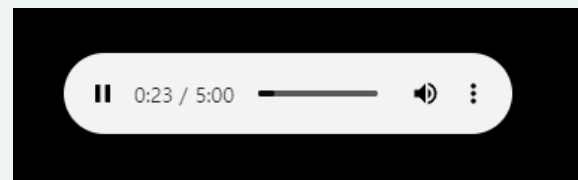


이 페이지 내용:

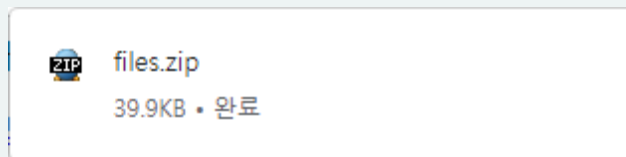
Hello! World!

확인

자바스크립트 코드 실행



음악 파일 연결



파일 다운로드

# 이미지맵

- ✓ 하나의 이미지를 여러 개의 영역으로 나누어  
각 영역마다 링크를 지정하는 것

```

```

```
<map name="맵이름">
```

```
  <area
```

```
    shape="이미지맵에서 각 영역의 모양(default, rect, circle, poly)"
```

```
    coords="각 영역을 구성하는 좌표값"
```

```
    href="URL"
```

```
    target="링크된 내용이 열릴 창"
```

```
    download="href로 지정된 파일을 다운로드"
```

```
    alt="대체 텍스트 내용">
```

```
  <!-- 영역의 개수만큼 area 요소를 반복 사용 -->
```

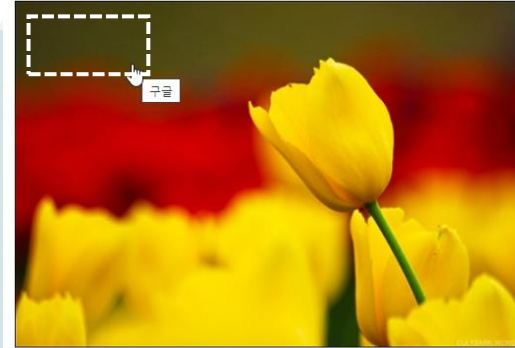
```
</map>
```



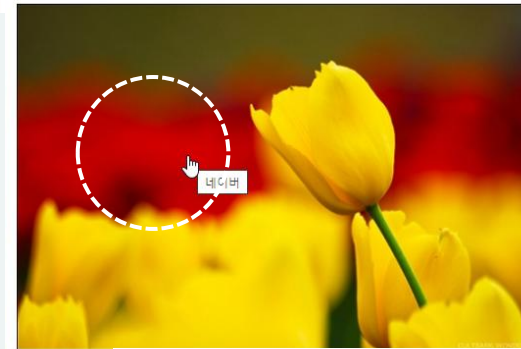
# 이미지맵

```

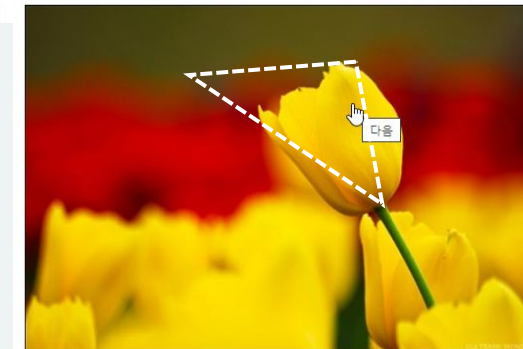
<map name="knou">
  <area shape="rect" coords="5,5,100,50"
    href="http://www.google.com"
    title="구글" target="_blank">
  <area shape="circle" coords="100,100,50"
    href="http://www.naver.com"
    title="네이버" target="_blank">
  <area shape="poly"
    coords="100,50,230,40,250,150"
    href="http://www.daum.net"
    title="다음" target="_blank">
</map>
```



www.google.com



www.naver.com



www.daum.net

# 새로운 창을 삽입하는 **iframe** 요소

✓ 현재의 웹 페이지 안에서 다른 웹 페이지를 표시하는 창을 삽입할 때 사용

✓ 속성

속성	설명
<code>name="이름"</code>	창의 이름 • target 속성의 값으로 사용
<code>src="URL"</code>	창에 표시될 문서의 URL
<code>srcdoc="HTML 코드"</code>	창에서 보여줄 HTML 콘텐츠 • srcdoc 속성이 지정되면 src 속성은 무시됨
<code>width="폭"</code>	창의 폭 (픽셀)
<code>height="높이"</code>	창의 높이 (픽셀)

# 새로운 창을 삽입하는 `iframe` 요소

&lt;/iframe&gt;



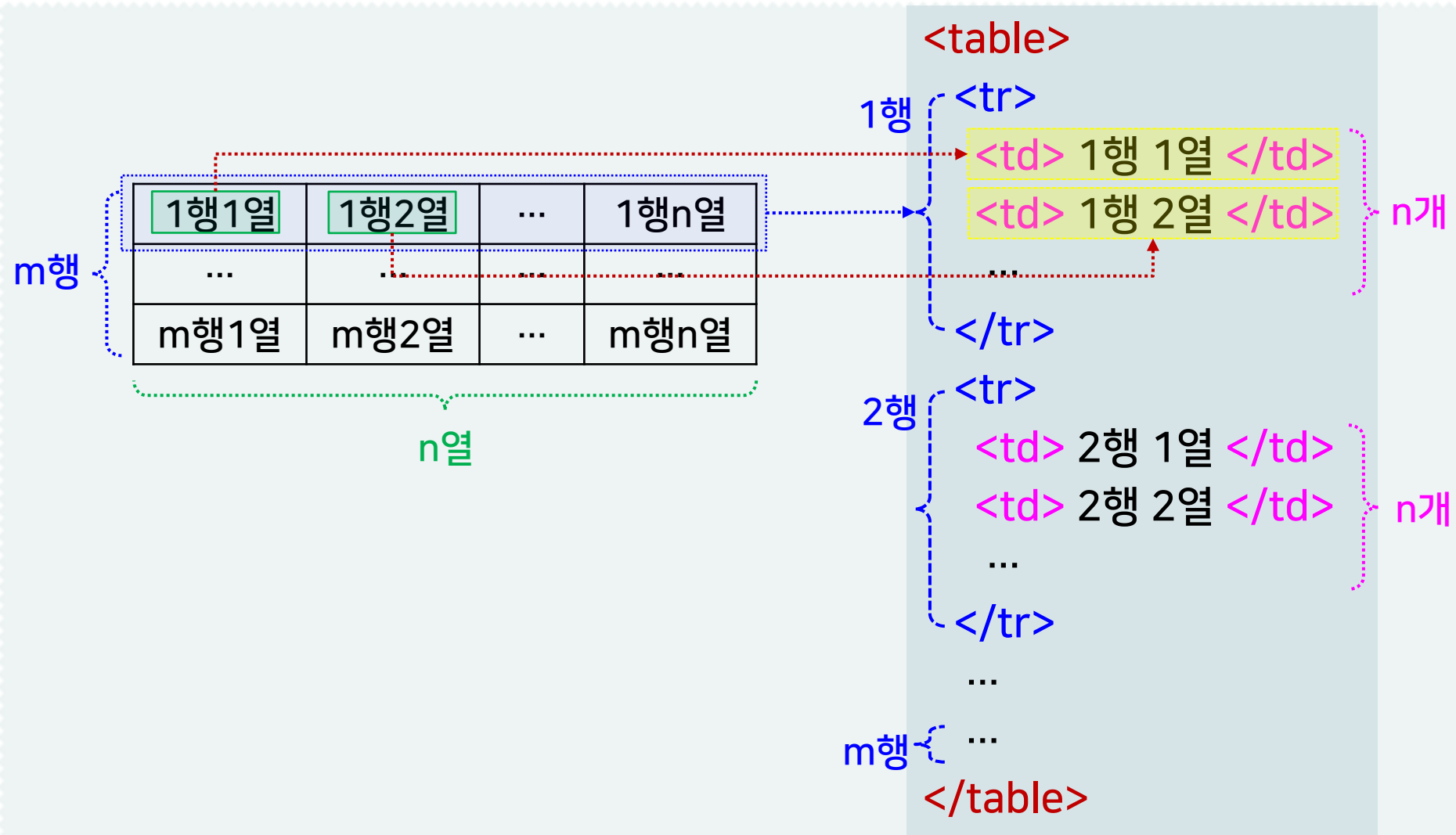
# 04

/ HTML5 Web Programming

## 테이블 표현



# 테이블의 기본 형식



# 테이블의 기본 형식

**<table>**

<caption>과목별 성적</caption>

<tr>

<th>과목</th><th>중간시험</th><th>기말시험</th>

</tr>

<tr>

<td>머신러닝 </td>

<td>30 </td>

<td>46 </td>

</tr>

<tr>

<td>알고리즘 </td>

<td>28 </td>

<td>50 </td>

</tr>

**</table>**

**<table>**

**<table border="0">**

과목별 성적

과목    중간시험    기말시험

머신러닝    30            46

알고리즘    28            50

**<table border>**,

**<table border="1">**

과목별 성적

과목

중간시험

기말시험

머신러닝    30            46

알고리즘    28            50

**<table border="5">**

과목별 성적

과목

중간시험

기말시험

머신러닝    30            46

알고리즘    28            50

# 테이블 생성을 위한 기본 요소

## ☑ table 요소

➤ HTML에서 테이블을 정의하는 요소

- m행×n열 → m개의 tr 요소, 각 tr 내에서 n개의 td 요소
- 테이블의 테두리 지정 → border 속성  
→ 테두리 스타일 지정은 CSS 사용

## ☑ tr 요소 'table row'

➤ 테이블에서 하나의 줄에 해당하는 행을 정의하는 요소

- td 요소(또는 th 요소)를 포함해야만 사용 가능

# 테이블 생성을 위한 기본 요소

## ✓ **td** 요소 'table data'

➤ 데이터(텍스트, 이미지, 테이블 등)를 표시하는 **표준 셀**을 정의

- 반드시 tr 요소 내에서 사용
- 셀 내용 → 왼쪽 정렬

## ✓ **th** 요소 'table header'

➤ 테이블 내부에서 행/열의 제목을 표시하는 **헤더 셀**을 정의

- td 요소의 위치에서 선택적으로 사용
- 셀 내용 → 진하게 가운데 정렬



# 테이블 생성을 위한 기본 요소

## ☑ caption 요소

- ⌚ 테이블 전체에 대한 제목을 표시할 때 사용
  - 테이블의 레이아웃에는 아무런 영향 없음
  - 반드시 table 요소의 시작태그 바로 다음에 위치시킴
    - tr/td 요소 내에서 사용되면 caption 요소는 무시됨
  - 테이블 상단에 진하게 가운데 정렬
    - CSS 속성(caption-side, text-align)으로 변경 가능

# 셀의 병합

✓ 인접한 여러 셀을 합쳐서 하나의 셀로 만드는 경우

➤ 보다 비정형적인 다양한 모양의 테이블 생성 가능

✓ `rowspan` 속성, `colspan` 속성

➤ `td` 또는 `th` 요소 내에서만 사용

➤ `rowspan="n"`

• 현재 셀 위치에서 인접한  $n$ 개의 행을 병합 → 세로 방향의 병합

➤ `colspan="n"`

• 현재 셀 위치에서 인접한  $n$ 개의 열을 병합 → 가로 방향의 병합

# 셀의 병합

## 3×4 크기의 기본 테이블

```
<table border="1">
  <tr>
    <td>1-1</td> <td>1-2</td> <td>1-3</td> <td>1-4</td>
  </tr>
  <tr>
    <td>2-1</td> <td>2-2</td> <td>2-3</td> <td>2-4</td>
  </tr>
  <tr>
    <td>3-1</td> <td>3-2</td> <td>3-3</td> <td>3-4</td>
  </tr>
</table>
```

1-1	1-2	1-3	1-4
2-1	2-2	2-3	2-4
3-1	3-2	3-3	3-4

# 셀의 병합

```
<table border="1">
  <tr>
    <td colspan="4">1-1 ~ 1-4</td>
  </tr>
  <tr>
    <td rowspan="2">2-1 <br> 3-1</td>
    <td rowspan="2">2-2 <br> 3-2</td>
    <td>2-3</td>
    <td>2-4</td>
  </tr>
  <tr>
    <td colspan="2">3-3, 3-4</td>
  </tr>
</table>
```

1-1	1-2	1-3	1-4
2-1	2-2	2-3	2-4
3-1	3-2	3-3	3-4



1-1 ~ 1-4			
2-1	2-2	2-3	2-4
3-1	3-2	3-3, 3-4	

# 행 단위 콘텐츠의 그룹핑

## ☑ **thead, tbody, tfoot** 요소

- ① 테이블에서 행 단위 콘텐츠를 **헤더, 몸체, 푸터** 부분으로 구분하여 그룹핑하는 요소
- ② 테이블 레이아웃에는 아무런 영향을 미치지 않음
- ③ 요소의 사용 위치
  - `<table> <caption> <colgroup> <thead> <tr> ... </table>`
  - `thead, tbody, tfoot` 요소는 순서대로 사용

# 행 단위 콘텐츠의 그룹핑

과목	점수	헤더
국어	90	
영어	95	
수학	85	
평균	90	푸터

몸체

```
<table border="1">
  <thead> <!-- thead { color : red } -->
    <tr> <th>과목</th><th>점수</th> </tr>
  </thead>

  <tbody> <!-- tbody { color : green } -->
    <tr> <td>국어</td><td>90</td> </tr>
    <tr> <td>영어</td><td>95</td> </tr>
    <tr> <td>수학</td><td>85</td> </tr>
  </tbody>

  <tfoot> <!-- tfoot { color : blue } -->
    <tr> <td>평균</td><td>90</td> </tr>
  </tfoot>
</table>
```

과목	점수
국어	90
영어	95
수학	85
평균	90

# 열 단위의 스타일 지정

## ✓ colgroup 요소

- 하나 이상의 열들을 모아서 한꺼번에 스타일을 지정할 때 사용
- 요소의 사용 위치
  - `<table> <caption> <colgroup> <thead> <tr> ... </table>`

## ✓ col 요소

- 지정된 열에 대해서 서로 다른 포매팅을 정의하는 요소
- colgroup 요소의 자식 요소
  - 종료태그 없이 사용
  - `span="n"` 속성 → 해당 요소의 스타일이 적용되는 열의 개수

# 열 단위의 스타일 지정

```
<table border="1">
```

```
<!-- colgroup 요소를 사용해서 열 단위로 스타일 지정 -->
```

```
<tr>
```

```
<th>과목</th> <th>중간시험</th> <th>기말시험</th><th>평균</th>
```

```
</tr>
```

```
<tr>
```

```
<td>국어</td><td>90</td><td>95</td><td>92.5</td>
```

```
</tr>
```

```
<tr>
```

```
<td>영어</td><td>96</td><td>90</td><td>93.0</td>
```

```
</tr>
```

```
<tr>
```

```
<td>수학</td><td>85</td><td>95</td><td>90.0</td>
```

```
</tr>
```

```
</table>
```

과목	중간시험	기말시험	평균
국어	90	95	92.5
영어	96	90	93.0
수학	85	95	90.0



# 열 단위의 스타일 지정

## colgroup 요소의 사용

```
<colgroup>
  <col class="subject">
  <col span="2" class="score">
  <col class="average">
</colgroup>
```

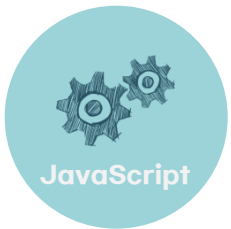
## CSS 스타일

```
<style>
  td { text-align: center }
  .subject { background-color: lightblue }
  .score { background-color: lightyellow }
  .average { background-color: lightgreen }
</style>
```

subject		score	average
과목	중간시험	기말시험	평균
국어	90	95	92.5
영어	96	90	93.0
수학	85	95	90.0



과목	중간시험	기말시험	평균
국어	90	95	92.5
영어	96	90	93.0
수학	85	95	90.0



HTML5 Web Programming

3강

# HTML 요소: 미디어, 시멘틱 요소

>>>>

- 
- 컴퓨터과학과 이관용 교수

# 학습목차

## 01 미디어 표현

## 02 시멘틱 요소



# 01

/ HTML5 Web Programming

## 미디어 표현



# 오디오를 재생하는 **audio** 요소

## ✓ 웹 페이지에서 오디오를 재생하기 위한 요소

오디오 재생: <br/><br/>

```
<audio src="song.ogg" controls autoplay loop>
```

<!-- 브라우저의 미지원을 대비한 대체 텍스트 표시 -->

```
</audio>
```

### 오디오 형식

MP3 (\*.mp3)

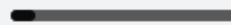
Wav (\*.wav)

Ogg (\*.ogg, \*.oga)

오디오 재생:



0:15 / 5:00



# 오디오를 재생하는 **audio** 요소

## ☑ 속성

속성	설명
autoplay	웹 문서 로딩 시에 오디오를 자동으로 즉시 재생 여부
controls	오디오의 제어 버튼(재생, 일시정지, 볼륨 등) 표시 여부
loop	반복 재생 여부
muted	음소거 여부
preload="값"	웹 페이지가 열릴 때 오디오의 로딩 방식 지정 <ul style="list-style-type: none"><li>• none → 재생 전까지 오디오 파일을 미리 로드하지 않음</li><li>• auto → (기본) 웹 페이지 로딩 시 바로 전체 파일을 자동으로 로드</li><li>• metadta → 재생 전까지 오디오의 메타 파일(크기, 첫 프레임 등)만 로드</li><li>• autoplay 속성이 지정되면 preload 속성은 무시됨</li></ul>
src="URL"	재생할 오디오 파일의 URL

속성명

속성명="속성명"  
예: loop="loop"

# 다양한 파일 형식을 지정하는 **source** 요소

✓ 하나의 미디어 소스에 대해 서로 다른 형식을 갖는 여러 개의 파일을 동시에 지정하기 위한 요소

- ⌚ audio/video 요소의 src 속성의 역할을 대신하는 것
  - 가장 위에서 나열된 source 요소의 파일부터 차례대로 재생 가능 여부를 검사해서 재생

✓ 속성

- ⌚ **src**="URL" → 재생할 미디어 파일의 URL
- ⌚ **type**="MIME형식" → "audio/mp3", "audio/wav", "audio/ogg"
- ⌚ **media**="미디어 쿼리" → 파일을 위한 미디어 쿼리 지정

# 다양한 파일 형식을 지정하는 **source** 요소

오디오 재생: <br/><br/>

```
<audio src="song.ogg" controls autoplay loop>
```

<!-- 브라우저의 미지원을 대비한 대체 텍스트 표시 -->

```
</audio>
```

오디오 재생: <br/><br/>

```
<audio controls autoplay loop>
```

```
<source src="song.ogg" type="audio/ogg">
```

```
<source src="song.wav" type="audio/wav">
```

```
<source src="song.mp3" type="audio/mp3">
```

<!-- 브라우저의 미지원을 대비한 대체 텍스트 표시 -->

```
</audio>
```



# 비디오를 재생하는 video 요소

## ✓ 비디오 재생을 위한 요소

➤ audio 요소의 사용 방법과 거의 동일

```
<video src="비디오 파일 URL" 속성 ... >
```

```
<!-- 브라우저의 미지원을 대비한 대체 텍스트 표시 -->
```

```
</video>
```

비디오 형식

MP4 (\*.mp4)

WebM (\*.webm)

Ogg (\*.ogg)

# 비디오를 재생하는 video 요소

## ✓ 속성

- ⌂ audio 요소의 속성을 모두 그대로 사용
  - autoplay, controls, loop, muted, preload, src
- ⌂ width="픽셀", height="픽셀"
  - 화면에서 비디오 콘텐츠가 표시될 영역의 크기(폭, 높이)
- ⌂ poster="이미지 파일의 URL"
  - 지정한 비디오 데이터가 설정되어 있지 않거나 비디오가 로딩되는 동안에 보여 줄 이미지 지정

# 비디오를 재생하는 video 요소

비디오 재생: <br/><br/>

```
<video controls width="700" height="400" poster="poster.jpg">
```

```
<source src="movie.mp4" type="video/mp4">
```

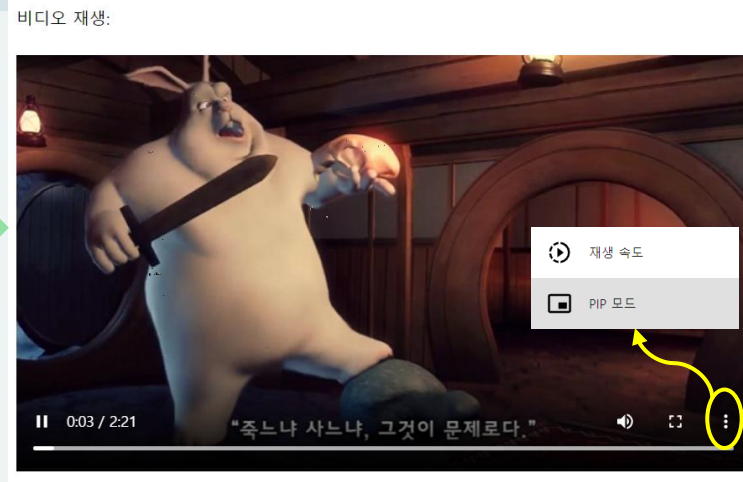
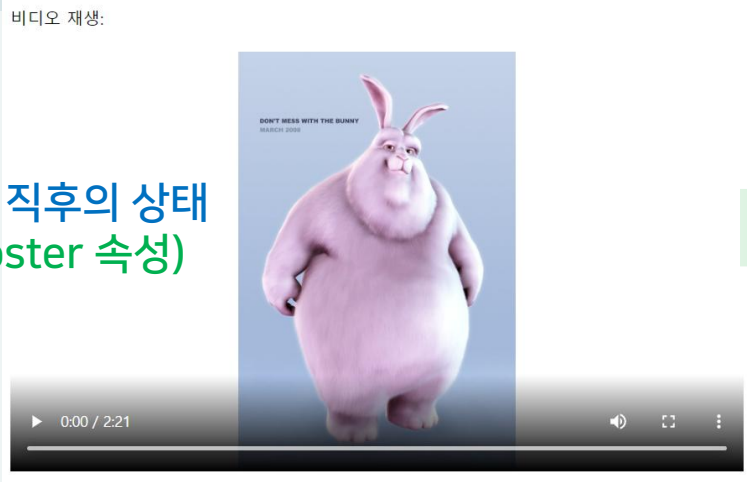
```
<source src="movie.webm" type="video/webm">
```

```
<source src="movie.ogv" type="video/ogg">
```

현재의 브라우저는 video 요소를 지원하지 않습니다.

```
</video>
```

실행 직후의 상태  
(poster 속성)



# 콘텐츠의 재생 구간 지정

✓ 미디어 콘텐츠를 재생할 때 특정 구간을 지정하여 재생

➤ audio/video 또는 source 요소의 **src** 속성 이용해서 지정

• `src="파일명#t=[시작시간][,종료시간]"`

#t=[시:분:초][,시:분:초]	설명
<code>src="a.mp4#t=1:20:32,2:05:00"</code>	1시간20분32초~2시간5분까지 재생
<code>src="a.mp4#t=20,40"</code>	20초~40초까지 재생
<code>src="a.mp4#t=,30"</code>	처음부터 30초까지 재생
<code>src="a.mp4#t=30"</code>	30초부터 재생 시작

# 콘텐츠의 재생 구간 지정

비디오 재생: <br/><br/>

<video src="movie.mp4#t=10,20" controls autoplay>

현재의 브라우저는 video 요소를 지원하지 않습니다.

</video>

비디오 재생:

실행 직후의 상태



재생

비디오 재생:

20초에서 재생을 멈춤



# 02 / HTML5 Web Programming

## 시멘틱 요소



# 시멘틱 요소?

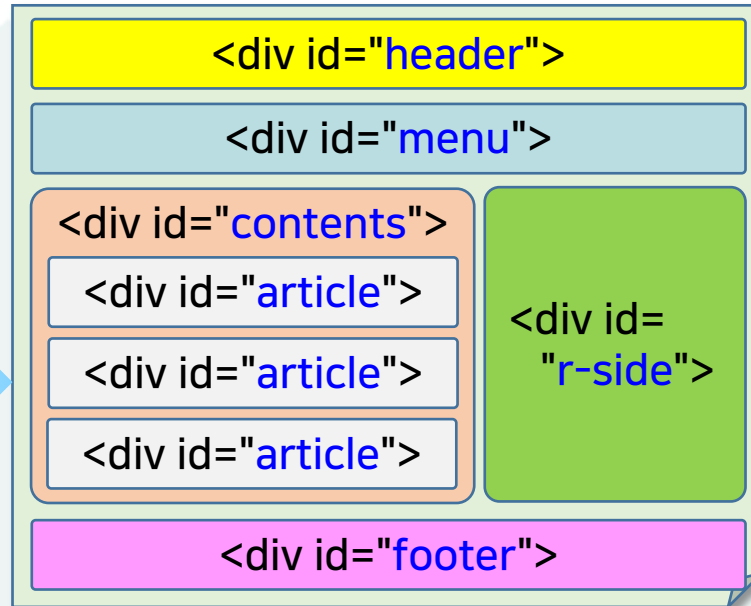
## ✓ semantic → "의미 있는", "의미의"

- 표현 중심이 아닌 의미 있는 구조의 문서 작성을 위한 요소
- HTML 문서에 의미 부여가 가능
  - 사람이 읽고 이해하는 것이 용이
  - 검색 엔진, 문서 해석기 등에 의한 자동적인 문서 처리/해석도 가능
- 문서 구조화 → header, nav, section, article, aside, footer
- 블록 요소 → main, figure ~ figcaption, details ~ summary
- 인라인 요소 → time, meter, progress, ruby ~ rt, wbr

# 기존의 문서 구조화 방식



div 요소



CSS 스타일 적용





# 문서 구조화를 위한 요소

---

header → 머리말(로고, 사이트 이름 등)을 나타내는 요소

nav → 메뉴 영역을 나타내는 요소

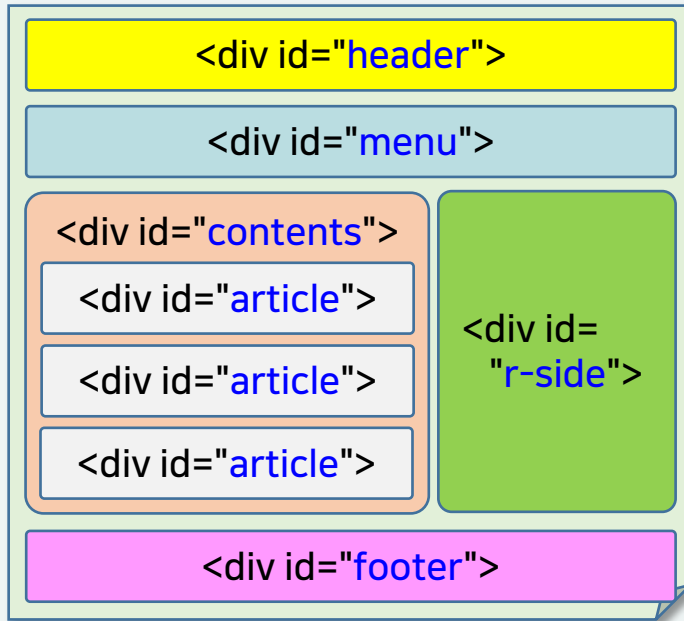
section → 제목/주제별로 나눌 수 있는 문서의 내용 영역을 구성하는 요소

article → 개별 콘텐츠를 나타내는 요소

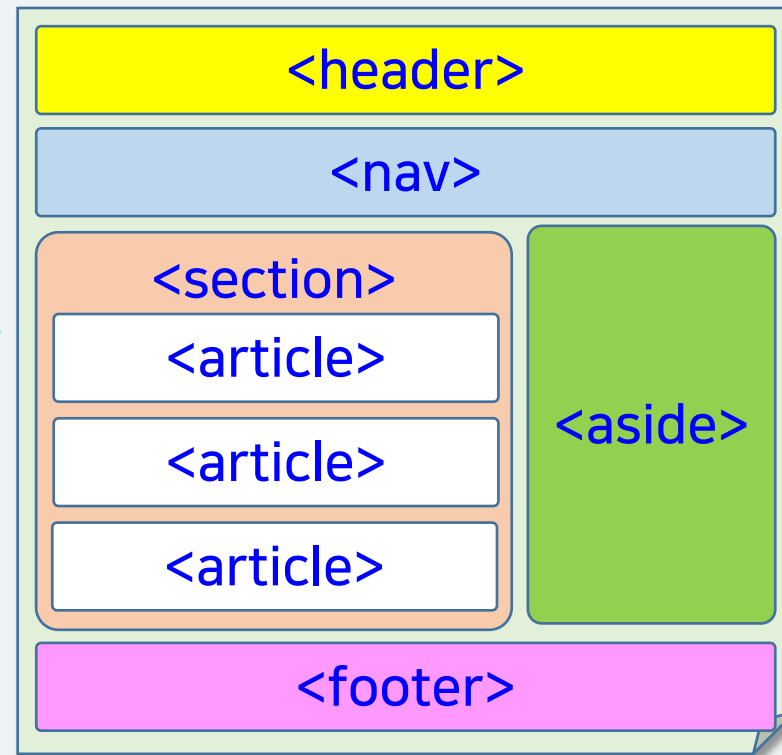
aside → 좌우측의 보조 콘텐츠 영역을 나타내는 요소

footer → 꼬리말(제작자 정보 및 저작권 정보 등)을 나타내는 요소

# 문서 구조화를 위한 요소



기존 HTML 방식의 문서 구조화



HTML5 요소를 이용한 문서 구조화

# 머리말과 꼬리말을 담은 header, footer 요소

## ✓ header 요소

➤ 머리말 → 사이트 이름, 로고와 소개 정보 등

```
<header> 머리말 </header>
```

```
<section>
```

```
  <h1> 제목 </h1>
```

```
  <p> 본문의 내용 </p>
```

```
</section>
```

```
...
```

```
<footer> 꼬리말 </footer>
```

- 위치 무관
- 내용 중간(section/article)에서 사용되는 경우 → 머리글/바닥글 역할
- section, header, footer 요소 포함 불가

## ✓ footer 요소

➤ 꼬리말 → 제작자 정보, 저작권 정보, 연락처 등

# 메뉴와 보조 콘텐츠를 담는 **nav, aside** 요소

## **nav** 요소

### 🔍 navigational links 영역

- 주로 웹 페이지 전체에 적용되는 상단의 메뉴를 표시하기 위한 링크를 모아 놓은 곳 → 페이지의 모든 링크를 포함하는 것은 아님

## **aside** 요소

### 🔍 주요 콘텐츠 이외에 남아 있는 콘텐츠나 참고가 되는 콘텐츠 등을 나타낼 때 사용

- 본문과는 직접 관련이 없지만 링크나 관련 정보를 표현
- 주로 웹 페이지의 왼쪽 또는 오른쪽에 위치

# 메뉴와 보조 콘텐츠를 담은 **nav, aside** 요소



`<header>머리말</header>`

`<nav>메뉴</nav>`

콘텐츠 영역

...

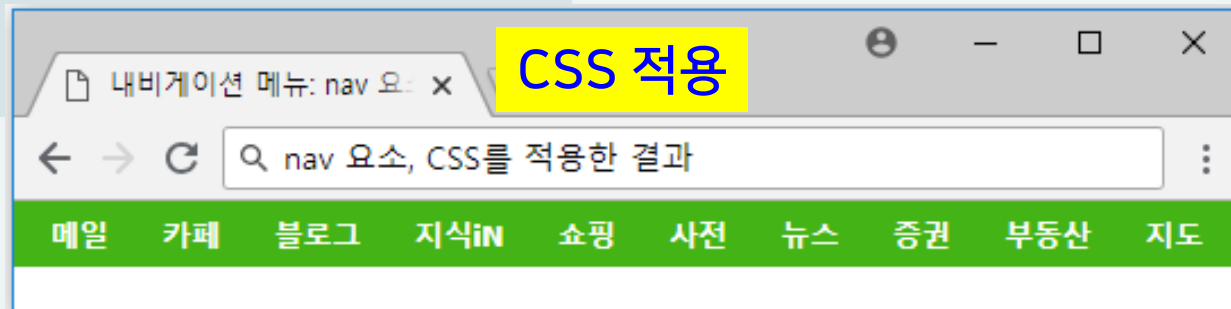
...

`<aside>사이드</aside>`

`<footer>꼬리말</footer>`

# nav 요소

```
<body>
<nav>
  <ol>
    <li><a href="...">메일</a></li>
    <li><a href="...">카페</a></li>
    <li><a href="...">블로그</a></li>
    ... 지식iN, 쇼핑, 사전, 뉴스, 증권, 부동산 ...
    <li><a href="...">지도</a></li>
  </ol>
</nav>
</body>
```



# 내용 영역 section, 개별 콘텐츠 article 요소



<section>

<h1>문화</h1>

<article>

<img ...>

<h3>'웹드' 놀러간 ... </h3>

<p>요즘 10대 스타의 ... </p>

</article>

<article>

<img ...>

<h3>'이상해도 괜찮아' ... </h3>

<p>새로운 상상력의 ... </p>

</article>

...

</section>

# 내용 영역 **section**, 개별 콘텐츠 **article** 요소

## ☑ **section** 요소

- 하나의 주제로 구성된 문서의 내용 영역을 표현
  - 하나의 section 요소에는 오직 하나의 헤딩 요소(h1~h6)만 사용

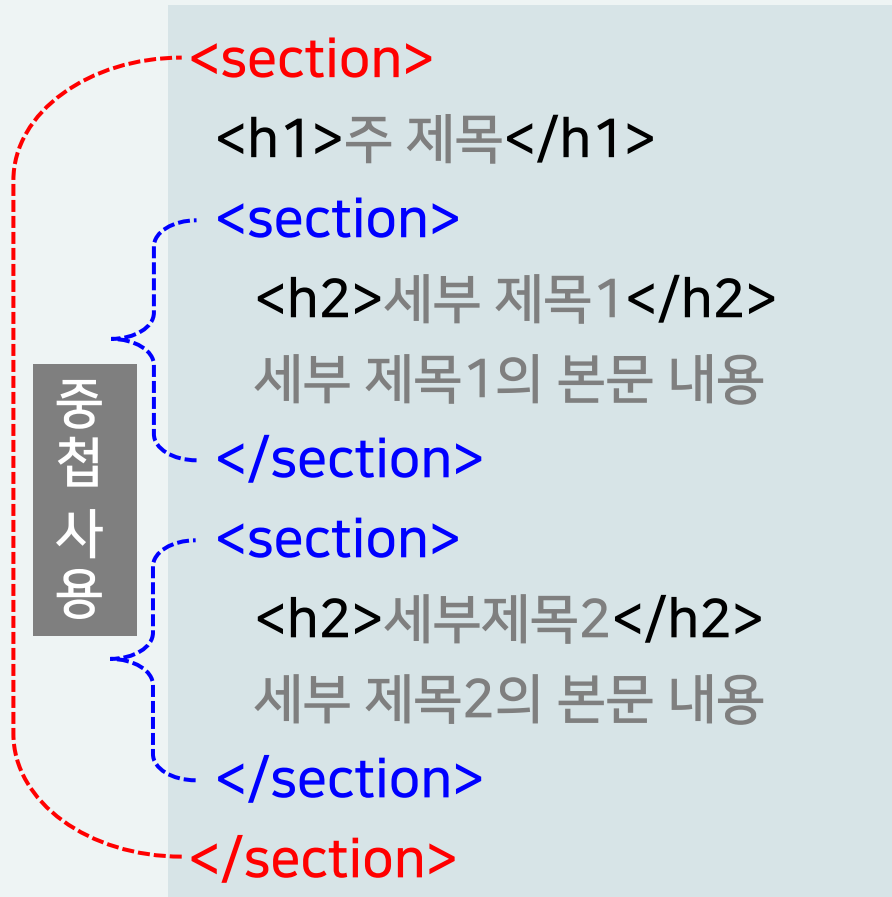
## ☑ **article** 요소

- 독립적으로 분배 및 재사용이 가능한 개별 콘텐츠(뉴스 기사, 블로그 포스트 등)을 나타낼 때 사용
  - section 요소와 분리하여 개별적으로 사용 가능
    - article 요소 안에 section 또는 article 요소 포함 가능



# section 요소

- ☑ 하나의 주제 안에 다른 세부 주제가 존재하는 경우



# section 요소

## ☑ 하나의 요소에 2개 이상의 헤딩 요소가 사용된 경우

→ 헤딩 요소의 레벨에 따라 묵시적으로 결정

### ① 같은 레벨의 헤딩 요소 사용

```
<section>
  <h1>제목 1</h1>
  제목 1의 본문 내용
  <h1>제목 2</h1>
  제목 2의 본문 내용
</section>
```

같은 레벨의  
section 요소로 구분

```
<section>
  <h1>제목 1</h1>
  제목 1의 본문 내용
</section>
<section>
  <h1>제목 2</h1>
  제목 2의 본문 내용
</section>
```

# section 요소

## ☑ 하나의 요소에 2개 이상의 헤딩 요소가 사용된 경우

### ② 다른 레벨의 헤딩 요소 사용

```
<section>
  <h1>제목 1</h1>
  제목 1의 본문 내용
  <h2>제목 2</h2>
  제목 2의 본문 내용
</section>
```

중첩된  
section 요소로 구분

```
<section>
  <h1>제목 1</h1>
  제목 1의 본문 내용
  <section>
    <h1>제목 2</h1>
    제목 2의 본문 내용
  </section>
</section>
```

# 시멘틱 블록 요소

---

- ✓ 주요 콘텐츠 블록을 지정하는 **main** 요소
- ✓ 삽입 그림을 블록화하는 **figure** 요소
  - 하위 요소 → **figcaption**
- ✓ 세부정보를 선택적으로 표시하는 **details** 요소
  - 하위 요소 → **summary**

# 주요 콘텐츠 블록을 지정하는 **main** 요소

- ☑ 문서 body 영역의 주요 콘텐츠 블록을 지정할 때 사용
  - 문서 전체에서 반복적으로 사용되는 콘텐츠는 포함 불가  
→ 사이드바, 내비게이션 링크, 저작권 정보, 사이트 로고, 검색 폼 등
  - 문서에서 유일한 부분을 의미 → 오직 한 번만 사용
  - 문서의 전체적인 레이아웃에는 아무런 영향을 주지 않음
  - article, aside, header, footer, nav 요소의 하위 요소로 사용 불가

# 주요 콘텐츠 블록을 지정하는 **main** 요소

```
<body>
  <main>
    <section>
      <h1>웹페이지 구성 요소</h1>
      <p>HTML, CSS, JavaScript</p>
      <article>
        <h3>HTML</h3> <p>버전 5, 2014</p>
      </article>
      <article>
        <h3>CSS</h3>   <p>모듈 3, 2012</p>
      </article>
      <article>
        <h3>JavaScript</h3> <p>ES6, 2015</p>
      </article>
    </section>
  </main>
</body>
```

## 웹페이지 구성 요소

HTML, CSS, JavaScript

### HTML

버전 5, 2014

### CSS

모듈 3, 2012

### JavaScript

ES6, 2015

# 삽입 그림을 블록화하는 **figure** 요소

- ✓ 본문에서 삽입된 독립적인 콘텐츠를 하나로 묶어  
↳ (그림, 다이어그램, 사진, 소스 코드, 동영상 등)

블록을 형성하고, 캡션을 포함시켜 서로의 관계를  
명확히 구조화시킬 때 사용

- 요소의 위치는 주요 콘텐츠와 독립적  
→ 삭제하더라도 문서의 흐름에는 아무런 영향을 주지 않음

- **figcaption** 요소

- figure 요소에 대한 캡션을 표시하는 자식 요소
- 위치 → figure 요소 바로 다음 또는 맨 마지막

# 삽입 그림을 블록화하는 figure 요소

```
<body>
```

```
<p>figure 요소는 본문에 삽입된 그림,  
다이어그램, 사진, 소스 코드, 동영상 등과  
같은 독립적인 콘텐츠를 블록화할 때 사용.</p>
```

```
<figure>
```

```

```

```
<figcaption>그림 1. 예쁜 노란색 튤립</figcaption>
```

```
</figure>
```

```
</body>
```

figure 요소는 본문에 삽입된 그림, 다이어그램, 사진, 소스 코드, 동영상 등과 같은 독립적인 콘텐츠를 블록화할 때 사용.



그림 1. 예쁜 노란색 튤립



# 세부정보를 선택적으로 표시하는 details 요소

- ✓ 사용자의 요구(핸들(▶) 클릭)에 따라 추가적인 세부정보를 보여주거나 숨기는 형태의 상호작용을 지원하는 요소
  - 세부정보는 어떤 종류의 콘텐츠도 가능
  - open 속성 → 세부정보가 사용자에게 보여지도록 지정
  - summary 요소
    - 세부정보에 대한 캡션을 지정하는 details 요소의 자식 요소
    - 캡션의 클릭을 통해서도 세부정보의 표시 여부의 선택이 가능
    - 미사용 → "세부정보"라는 기본 제목 표시

# 세부정보를 선택적으로 표시하는 details 요소

<details>

<summary>HTML</summary>

<p>웹페이지의 구조와 내용 정의</p>

...

</details>

<details open>

<summary>CSS</summary>

<p>웹페이지의 출력 모양 지정</p>

...

</details>

<details>

<summary>JavaScript</summary>

<p>웹페이지의 동작을 프로그래밍</p>

...

</details>

실행 직후 화면

▶ HTML

▼ CSS

웹페이지의 출력 모양 지정

CSS1 - CSS2 - CSS3 (2012)

다단, 변형, 변환, 애니메이션 등의 다양한 기능이 추가됨

▶ JavaScript

각 핸들을 클릭한 후

▼ HTML

웹페이지의 구조와 내용 정의

HTML 5 표준안(Recommendation) 발표, 2014년 10월

HTML Living Standard로 단일 표준화, 2019년 5월

▶ CSS

▼ JavaScript

웹페이지의 동작을 프로그래밍

HTML5의 공식적인 스크립트 언어

캔버스, 웹 스토리지, 웹 워커 등 다양한 API가 추가됨

# 시멘틱 인라인 요소

- ✓ 시간/날짜를 지정하는 **time** 요소
- ✓ 주어진 값이나 비율을 나타내는 **meter** 요소
- ✓ 작업의 진행률을 나타내는 **progress** 요소
- ✓ 루비 주석을 지정하는 **ruby** 요소
  - 하위 요소 → **rt**
- ✓ 줄바꿈의 허용 위치를 지정하는 **wbr** 요소

# 시간/날짜를 지정하는 **time** 요소

- ✓ 시간/날짜 표현에 의미를 부여하기 위한 요소
  - 🕒 **datetime** 속성 → 시스템이 인식할 수 있도록 정확한 형식의 시간/날짜를 지정

```
<body>
```

```
<p>기상 시간은 <time>07:00</time>이며,
```

```
취침 시간은 <time>23:00</time>입니다.</p>
```

```
<p>HTML5 표준안은 <time datetime="2014-10-28">
```

```
2014년 말</time>에 발표되었다.</p>
```

```
</body>
```

기상 시간은 07:00이며, 취침 시간은 23:00입니다.

HTML5 표준안은 2014년 말에 발표되었다.

# 주어진 값이나 비율을 나타내는 meter 요소

✓ 일정 범위 안의 측정값이나 분포 비율 등을 표시할 때 사용

➤ "측정기" gauge → 현재 디스크 사용량과 같이  
일정 시점에서 주어진 (정적인) 양을 표시

➤ 속성

- value → (필수) 실제로 측정한 현재의 값
- min, max → 요소가 인식하는 최소값(0.0)과 최대값(1.0)
- low, high → 허용되는 범위의 최소값/최대값  
 $\text{min} \leq \text{low} \leq \text{high} \leq \text{max}$
- optimum → 최적의 값

# 주어진 값이나 비율을 나타내는 meter 요소

```
<body>
```

```
<p>디스크 사용률 <meter max=100 value=70></meter> (1TB 기준)</p>
```


```
<p>영화 예매율 <meter min=0 max=100 value=95></meter></p>
```

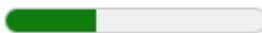
```
<p>지방선거 투표율
```

```
<meter value=0.35 title="오후 3시 현재 투표율"></meter></p>
```

```
</body>
```

디스크 사용률  (1TB 기준)

영화 예매율 

지방선거 투표율 

# 작업의 진행률을 나타내는 progress 요소

✓ 작업의 현재 진행 정도를 나타낼 때 사용

➤ 파일 복사, 다운로드 등과 같이  
동적으로 변하는 작업의 진행 정도를 표현할 때 유용

➤ 속성

- value → 완료된 작업의 양을 나타내는 현재 상태의 값  
( 0.0 ~ 1.0 또는 0.0 ~ max )
- max → 작업에 필요한 작업량의 최대값 (1.0)

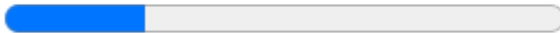
# 작업의 진행률을 나타내는 progress 요소

```
<body>
```

파일 다운로드 상태:

```
<progress value="25" max="100"></progress>
```

```
</body>
```

파일 다운로드 상태: 

- ➔ 현재 진행 중인 상태의 값(value)의 지속적인 업데이트가 필요  
→ 반드시 자바스크립트와의 연동이 필요



# 루비 주석을 지정하는 **ruby** 요소

- ✓ 하나 이상의 구문 콘텐츠에 루비 주석을 표시할 때 사용
  - 루비 주석 → 일본어/한자의 기본 문자열 주변에 발음법이나 의미를 나타내기 위해 추가되는 짧은 텍스트
  - **ruby** 요소
    - 하나 이상의 문자로 구성되는 기본 텍스트를 작성
  - **rt** 요소
    - ruby 요소의 자식 요소
    - rt 요소 바로 앞에 오는 하나 이상의 구문에 대한 루비 주석 표시

# 루비 주석을 지정하는 **ruby** 요소

```
<ruby>韓<rt>かん</rt></ruby>
<ruby>國<rt>こく</rt></ruby>
<ruby>放<rt>ほう</rt></ruby>
<ruby>送<rt>そう</rt></ruby>
<ruby>通信<rt>つうしん</rt></ruby>
<ruby>大學<rt>だいがく</rt></ruby><br><br>
<ruby>HTML5 웹 프로그래밍
  <rt>2학년1학기</rt></ruby><br><br>
<ruby>日本<rt>にほん</rt></ruby><br><br>
```

かん こく ほう そう つう しん だいがく  
韓 國 放 送 通 信 大 學

2학년1학기  
HTML5 웹 프로그래밍

にほん  
日本

# 줄바꿈의 허용 위치를 지정하는 **wbr** 요소

- ✓ 텍스트에서 줄바꿈을 허용할 위치를 지정할 때 사용
  - “허용 위치” → 줄바꿈이 수행되어도 괜찮은 지점
    - 해당 위치에서 실제로 줄바꿈이 무조건 수행되는 것이 아니라 레이아웃의 배치 상황에 따라 결정
  - 긴 URL 또는 단어/텍스트를 표시할 때 유용

# 줄바꿈 허용 위치를 지정하는 wbr 요소

<p>This is a  
veryveryveryveryveryveryveryveryveryveryveryveryveryveryveryvery<wbr>long  
wordthatwillbreakatspecific<wbr>placeswhenthebrowserwindowisresiz  
ed.</p>

This is a  
veryveryveryveryveryveryveryveryveryveryvery

## wbr 요소를 미적용한 경우

This is a veryveryveryveryveryveryveryveryveryveryveryveryveryveryveryvery  
longwordthatwillbreakatspecificplaceswhenthebrowserwindowisresized.

## wbr 요소를 적용한 경우

This is a  
veryveryveryveryveryveryveryveryveryveryveryveryveryveryveryvery  
longwordthatwillbreakatspecific  
placeswhenthebrowserwindowisresized.



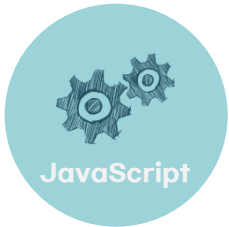
Programming



CSS



HTML5



JavaScript

HTML5 Web Programming

5강

# CSS: 개요

>>>>

- 
- 컴퓨터과학과 이관용 교수

# 학습목차

## 01 / CSS 개요

## 02 /

## 03 /



# 01

/ HTML5 Web Programming

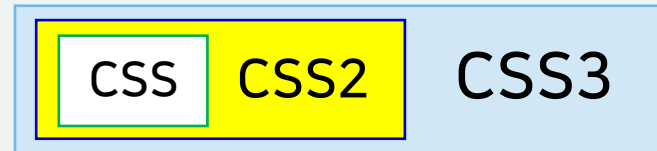
# CSS 개요



## ✓ Cascading Style Sheets

👉 웹 문서에 스타일(글꼴, 색상, 여백 등)을 추가하는 간단한 메커니즘

- 웹 문서의 출력을 위한 외양 지정



h2 Welcome to 웹 프로그래밍

hr

p 문서의 외양 정의 - Cascading Style Sheets

CSS 미적용

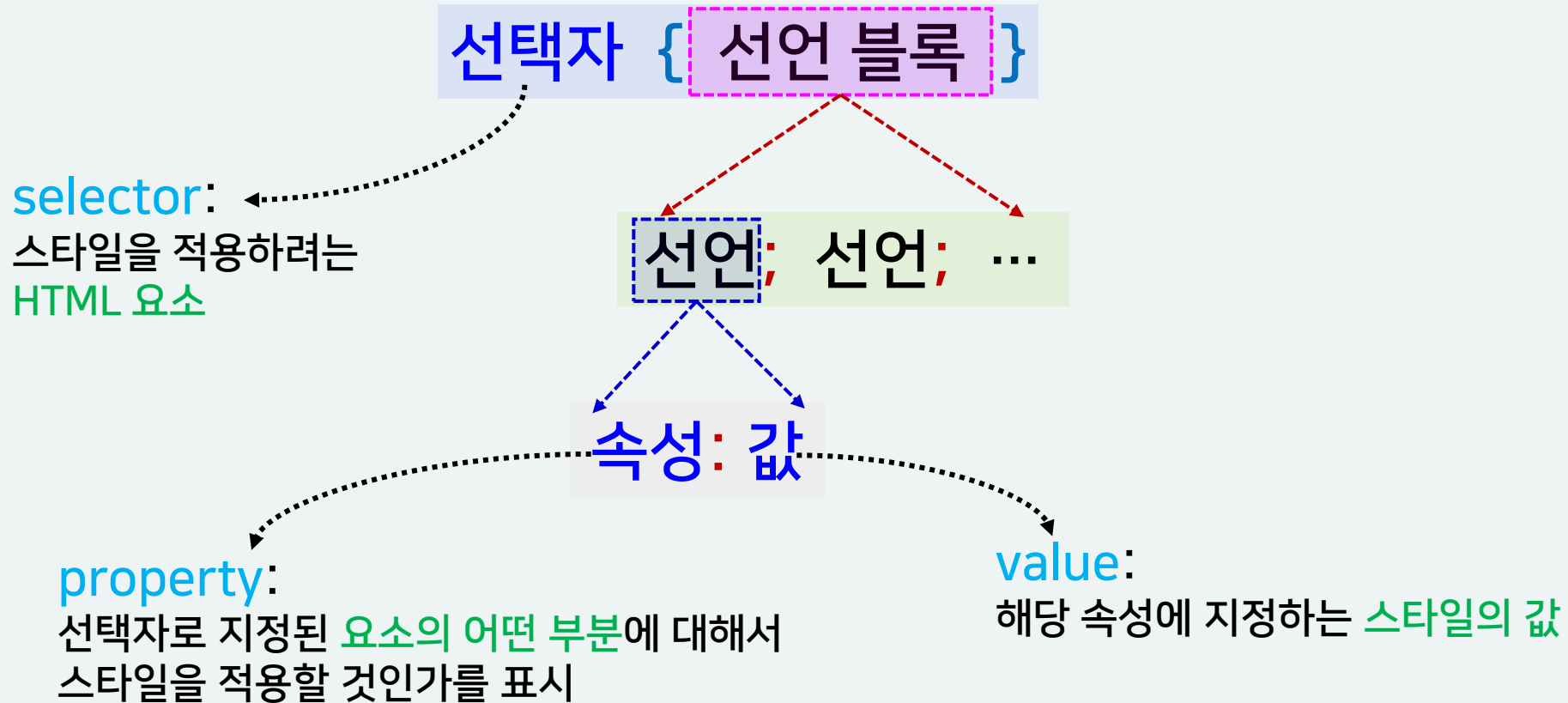
Welcome to 웹 프로그래밍

문서의 외양 정의 - Cascading Style Sheets

CSS 적용



# CSS 형식



```
p { color: red; font-size: 25px; }
```

# CSS 적용 방법

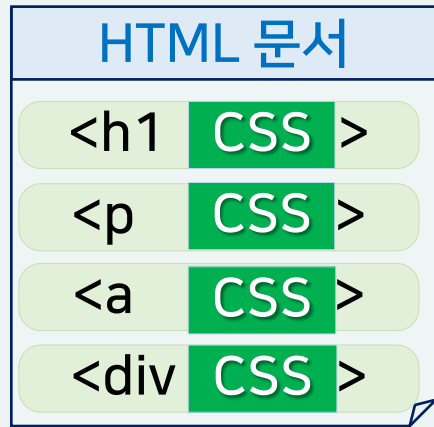
## ✓ HTML 문서에서 스타일이 정의되는 위치에 따른 구분

### style 요소 사용



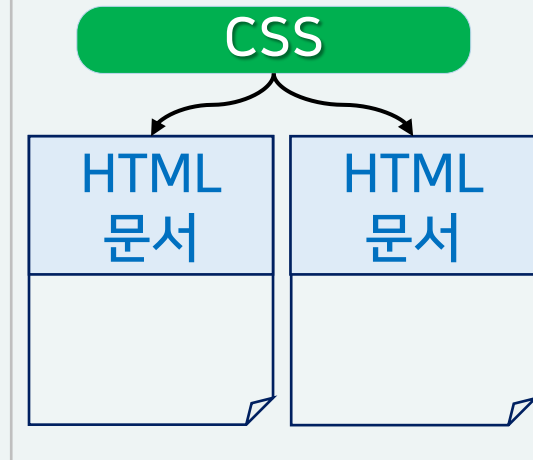
임베디드 방법

### style 속성 사용



인라인 방법

### 외부 파일 사용



링크 방법

# CSS 적용 방법

## style 요소 사용

```
<head>
  <style>
    p { color: red; font-size: 25px; }
  </style>
</head>
```

## style 속성 사용

```
<p style="color: red; font-size: 25px;">CSS</p>
```

## 외부 파일 사용

```
ex.css → p { color: red; font-size: 25px; }
```

```
<head>
  <link rel="stylesheet" href="ex.css">
</head>
```

```
<style>
  @import url(ex.css)
  ...
  @import "ex.css"
</style>
```

# CSS 적용의 우선순위

<head>

<!-- 외부 파일(mystyle.css): body { background-color: yellow; } -->

<link rel="stylesheet" href="mystyle.css">

<!-- style 요소 -->

<style>

body { background-color: skyblue }

</style>

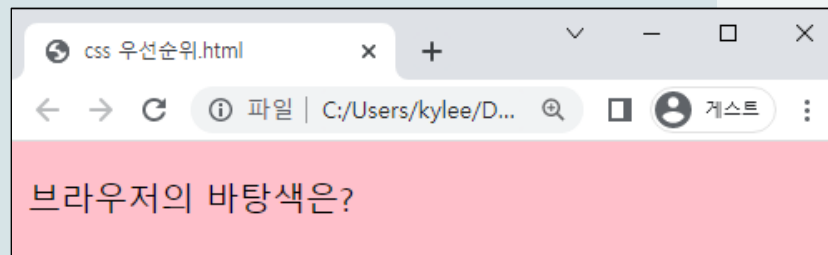
</head>

<!-- style 속성 -->

<body style="background-color: pink">

<p>브라우저의 바탕색은?</p>

</body>



style 속성 > style 요소 > 외부 파일 > 브라우저 디폴트

Thank you!



**출석수업을 마칩니다!**