

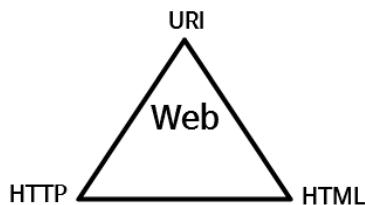
Internet이란

월드 와이드 웹 (www: World Wild Web)의 정의

- 세계 규모의 거미집 또는 거미집 모양의 망

- 하이パーテ스트(hypertext) 기능을 통해 인터넷 상에 분산된 온갖 종류의 정보를 HTTP(Hyper Text Transfer Protocol) 통신규약을 통해 전달되어 사용자에게 전달한다.
- 하이パーテ스트(hypertext) 란
하이퍼링크(hyper link)와 쌍방향성이라는 컴퓨터의 특성을 결합한 것으로 1960년대에 테오도르 넬슨이 만든 컴퓨터 관련용어.

웹의 3대요소



URI : 정보자원의 위치를 표시하기 위한 표기법

HTTP : 위치 표시가 있는 정보자원에 접근하기 위한 통신 규약

HTML : 정보자원과 정보자원 사이를 오가기 위한 하이パーテ스트

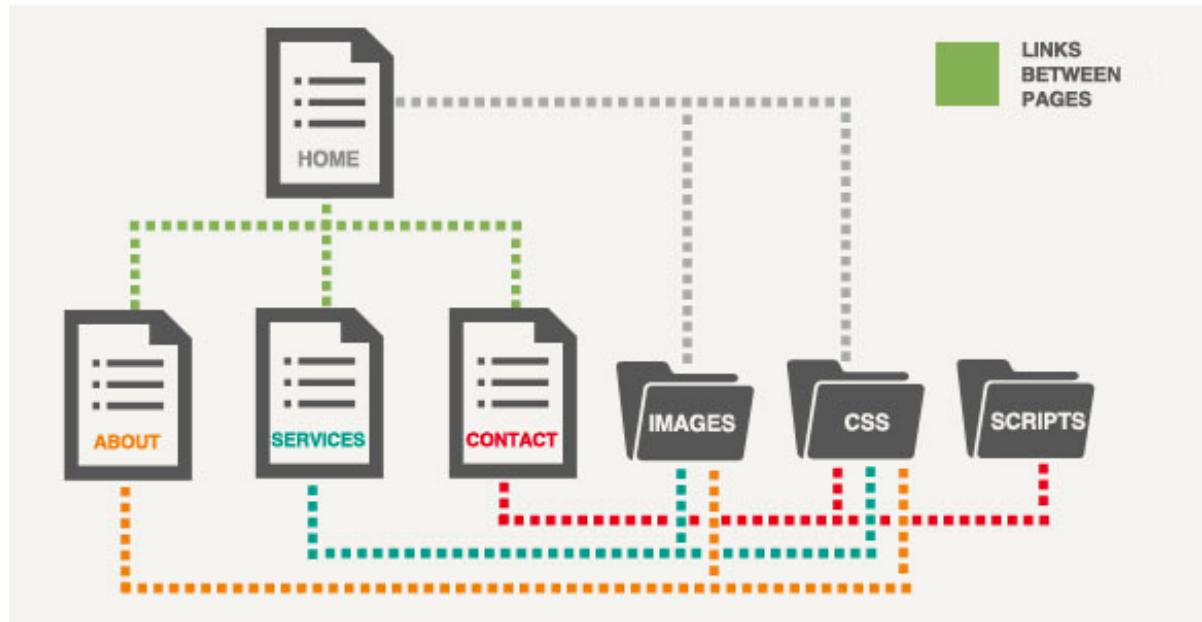
웹 동작 방식



웹의 동작 방식은 서버(server)와 클라이언트(client)의 형태를 띠며, 클라이언트가 웹 브라우저를 통해 특정 콘텐츠를 요청하면 서버는 클라이언트의 요청에 응답하는 형식으로 이루어져 있습니다.

서버(Server)





웹사이트 개발 기술



HTML5 소개

- 큰 의미 : 웹 표준 기술을 총칭
- 작은 의미 : 웹 문서의 문법을 의미

HTML이란?

html은 **Hyper Text Markup Language**의 약자이며, 간단히 말해 웹사이트를 만드는 것은 코딩언어(명령어)의 집단을 구성하는 거다.

* Markup이란?

문서의 일부를 '태그(Tag)'라 불리는 특별한 문자열로 둘러 쌈으로서 문장의 구조(목차, 또는 하이퍼링크 등), 수식정보(문자의 크기, 조판 상태 등)를 문장 안에 기술하는 모든 것을 말한다.

•

CSS3 소개

- CSS Cascading Style Sheets : HTML 문서를 표현하는 방법을 기술하는 언어
- 스타일시트는 현대 웹 페이지에서 매우 중요한 역할임

자바스크립트

- 표준 명칭 : ECMAScript5
- HTML에서 사용자 반응 등을 처리하는 데 사용하는 프로그래밍 언어
- 현대에는 서버는 물론 로봇 개발에도 사용됨

웹브라우저

웹 브라우저(Web Browser, 문화어: 열람기)는 웹 서버에서 쌍방향 통신하는 HTML 문서나 파일과 연동하고 출력하는 응용 소프트웨어이다. 웹 브라우저는 대표적인 HTTP 사용자 에이전트의 하나이기도 하다.

주요 웹 브라우저로는 모질라 파이어폭스, 구글 크롬, 인터넷 익스플로러/마이크로소프트 엣지, 오페라, 사파리가 있다.

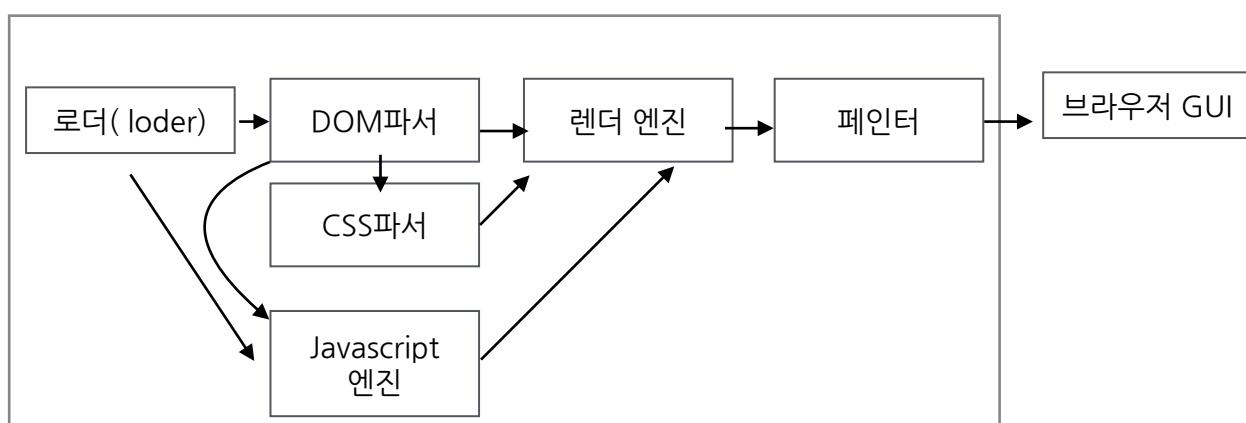
웹브라우저의 핵심은 웹 브라우저엔진으로 종류는 다음과 같다.

게코(Gecko): 파이어폭스, 파이어 폭스 모바일

웹킷(WebKit): 애플 사파리, 사파리 모바일, 구글 크롬

트라이던트(Trident): 마이크로 익스플로러, 익스플로러 모바일

프레스토(Presto): 오페라

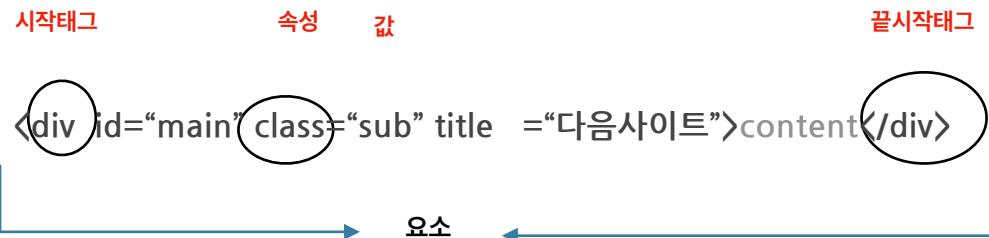


HTML 문서구조와 서식

구성요소

html 문서의 가장 작은 단위를 우리는 요소라 한다. 요소는

<시작태그 속성1="값" 속성2="값">콘텐츠</끝태그>



태그(tag): 태그는 <와 >로 묶어서 표현하는 명령어를 말합니다. 기본 형식은 <Tag>~</Tag>이며, 일부 명령어인 empty element(빈 요소)의 경우 <시작Tag> 형태로 기술합니다.

속성(attribute): 시작 태그는 태그의 의미와 필요에 따라 개별적인 옵션을 가질 수 있는데, 이러한 옵션을 '속성(attribute)'이라고 합니다.

속 여려 개의 속성을 하나의 태그에 지정할 때는 공백으로 구분하여 시작 태그에 지정한다.

속성에는 공통속성, 특별한 의미가 있는 속성으로 구분된다.

값(value): 각 속성이 가지는 값을 의미하며, 속성에 값을 할당할 때는 대입 연산자인 =과 함께 지정하며 “ ” 혹은 ‘ ’로 감싸주어야 한다.

요소(element): 시작 <Tag>~종료 </Tag>까지의 모든 명령어 집합을 '요소(element)'라고 합니다.

HTML 문서에서 콘텐츠는 이러한 요소들로 구성되며 각 요소의 의미에 따라 콘텐츠의 제목이나 본문의 구조를 가집니다.

서식: html의 올바른 코딩 법

1	요소명과 속성명에 대,소문자 사용 가능	
---	-----------------------	-----------------------------

XHTML1.0:

요소 속성명에 소문자만 사용 허용

2	빈 요소 사용 시 <요소명> 형식으로 기술	<link href="../css/reset.css" rel="stylesheet" type="text/css" >
---	-------------------------	--

XHTML1.0:

빈요소 사용시 '/' 사용 해야만 함

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

3	속성에 속성 값을 넣어야 한다. 하지만 논리 속성의 경우 값을 생략 한다(속성명으로 참과 거짓의 값을 가진다)	<option selected></option>
---	---	----------------------------

XHTML1.0:

속성에 무조건 값이 있어야 한다.

<option selected=selected></option>

4	잘못된 중첩의 사용 불가능	<p></p>
---	----------------	---------------------------

잘못 된 경우 <p></p>

6	<, >, & 을 <, &rt;, &로 변환하여 사용	<h1>html - CSS</h1> → <h1>html – CSS</h1>
---	--------------------------------------	---

참고: 특수 기호는 엔티티문자와 엔티티숫자로 사용을 하여야 한다.

<p>Copyright © Company all rights reserved.</p> (좋은 예)

<p>Copyright © Company all rights reserved.</p> (좋은 예)

<p>Copyright ©Company all rights reserved.</p> (잘못된 사용 예)

참고 사이트 <http://entitycode.com/>

Character	Entity Name	Entity Number	Description
&nbsp	 	 	Inserts A Non-Breaking Blank Space
&	&	&	Ampersand
"	"	"	Quotation Mark
©	©	©	Copyright Symbol
®	®	®	Registered Symbol
TM	™	™	Trademark Symbol

7	MINE 타입 생략가능	<script></script>
---	--------------	-------------------

XHTML1.0:

MINE 타입을 꼭 넣어야 한다.

<script type="text/javascript"></script>

주석 사용하기

주석은 문서작성 시 필요한 메모나 설명을 적어 놓는 코드로 브라우저에서는 코드로 인식되지 않는다.

<!-- 설명 -->

<!-- header -->

<!-- nav의 시작 부분 -->

html 문서 구조

구조

<!doctype html>	DTD선언
<html>	Html문서시작
<head>	head시작->
</head>	head끝
<body>	body시작->
</body>	body끝
</html>	html문서 끝

DTD 선언

웹 페이지를 제작할 때 사용할 문서형 정의를 선언합니다.

<html>~</html>

웹 페이지의 시작과 종료를 의미합니다.

<head>~</head>

html 문서의 머리영역으로 문서의 일반적인 정보와 title 등의 제목등 문서의 필요한 부분을 준비하는 역할을 한다.

<body>~</body>

html 문서의 본문 부분에 해당하며, 웹 브라우저 화면에 나타나는 모든 콘텐츠는 <body> 태그와 </body> 태그 사이에 선언해야 합니다.

여러DTD 선언

html 4.0	Strict	<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
	Transitional	<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN " "http://www.w3.org/TR/html4/loose.dtd">
	Frameset	<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN " "http://www.w3.org/TR/html4/frameset.dtd">
Xhtml 1.0	Strict	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
	Transitional	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
	Frameset	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
html 5		<!doctype html>

<html>요소

lang속성 사용하기

```
lang="언어코드"  
lang="ko"  
<html lang="ko">
```

<head>요소

head에 사용할 수 있는 추가 요소

meta	웹 페이지에 추가 정보 전달(빈요소)
title	웹 페이지의 제목 지정
script	웹 페이지에 스크립트 추가
link	웹 페이지에 다른 파일 연결
style	웹 페이지에 스타일 시트 추가
base	웹 페이지의 기본 경로 지정

<title>

- title 요소는 웹문서의 제목을 선언할 때 사용하는 요소로, 문서마다 유일한 내용으로 구성해야 합니다.
- 시각장애인의 경우에는 음성 브라우저를 이용하여 콘텐츠를 탐색하기 때문에 title 요소는 문서를 구별할 수 있는 첫 번째 관문이라고 할 수 있습니다.
- title 요소는 즐겨찾기 및 북마크에 해당 웹 문서를 추가할 경우에 사용되는 북마크 이름으로도 활용 되므로 같은 title을 가진 웹 문서가 여러 개 있어서는 안됩니다.

```
<title>문서 제목</title>
```

<meta>

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

제목요소

제목<hn>:<h1>~<h6>

- 문서 내부의 콘텐츠 제목을 정의하는 요소.
- 1~6까지 여섯 단계로, 순서에 맞게 작성.
- 논리적인 웹 문서를 제작하기 위해 <h1> 다음에 바로 <h3>이 나오지 않도록 주의
- 가급적 h1 요소는 한 문서에 한번만 지정하는 것을 권장

```
<h1> 대제목 </h1>
<h2> 중제목 </h2>          논리성 -> 웹표준 방법 -> 논리성만
<h3> 소제목 </h3>           -> 시맨틱요소 방법 -> 구조 -> outliner (목차)
<h4> 소소제목 </h4>
<h5> 소소소제목 </h5>
<h6> 소소소소제목 </h6>
```

<h1> Heading1</h1>

<h3> Heading3</h3>

잘못된 표현

<h1> Heading1</h1>

<h2> Heading2</h2>

올바른 표현



문단요소

문단<p>

p 요소는 텍스트를 문단으로 정의할 때 사용합니다.

문단 요소 안에는 <a>, 등과 같은 인라인 요소와 텍스트만 포함할 수 있으며, 블록 요소는 사용할 수 없습니다.

문단 안에서 강제로 줄바꿈해야 하는 경우가 있을 때는
 요소를 사용할 수 있지만, 시각적 효과 등을 위해 함부로 사용하지 않아야 합니다.

<p>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus

</p>

<p>

<label for="userPw">패스워드</label>

<input type="password" id="userPw">

</p>

줄 바꿈요소

- 문단에서 텍스트를 강제로 줄바꿈할 때는 'line break'의 의미인 br 요소를 사용합니다.
- 빈 요소 (종료 태그가 필요없음)
- br 요소를 여러 번 사용하여 행 간격을 늘이는 것은 가능하지만, 이와 같은 시각 효과를 위해서 사용하지 않는다.

```
<p>
```

인라인 요소와 텍스트를 포함할 수 있는 블록 요소이지만,

 또 다른 블록 요소를 포함할 수는 없다.
 </p>

구분선:<hr>

문서 내에 주제를 분리하는 역할을 한다.

논리적인 의미의 구분이라기 보다는 콘텐츠의 전후의 내용을 구분할 수 있도록 선으로 나타내기 때문에 웹페이지의 경우 문서의 헤더, 본문, 푸터를 구분하는데 사용하고 본문의 주제가 바뀌는 부분에 적용 CSS가 제공되지 않는 환경에서 콘텐츠의 구조적 구분을 표현할 수 있다.

```
<div id="content"> content 영역 </div>
<hr />
<div id="aside"> aside 영역 </div>
```

목록

- 일반적인 목록을 구성하는 요소:ul, ol, li
- 블록요소

ul, ol : 같은 등급의 목록 묶음
 dl :

비순서형 목록:

li의 순서가 변동이 되어도 됨

항목의 순서를 따지지 않고 목록 나열 (unordered list)

li만 포함 가능

```
<ul>
    <li> 목록 항목 </li>
    <li> 목록 항목 </li>
</ul>
```

순서형 목록:

li의 순서가 변동이 되면 안됨 E) 순위

- 항목의 순서대로 목록 나열 (ordered list)
- li만 포함 가능
- 번호 목록 대신 알파벳이나 로마자 표현 가능, 'type' 속성 정용 적용가능값= a,A,i,l

```
<ol>
    <li> 목록 항목 </li>
    <li> 목록 항목 </li>
</ol>
```

정의형 목록:<dl><dt><dd>

dl 요소는 'Definition List'의 약자로, 용어 정의 리스트를 생성할 때 사용합니다.

dl 요소의 경우 단순히 정의형 목록(용어 제목, 용어 설명 등)만이 아니라 주종 관계가 성립되는 콘텐츠에 광범위하게 사용할 수 있습니다.

dl

- 정의형 항목의 목록 나열 (definition list)

- 블록 요소
- dt 요소와 dd 요소만 포함 가능

dl : 사전을 만들기 위해 생김
 dt -> 제목
 dd -> 본문, 제목 설명, 제목 값

dt

- 정의형 항목의 용어 (definition term)
- 블록 요소

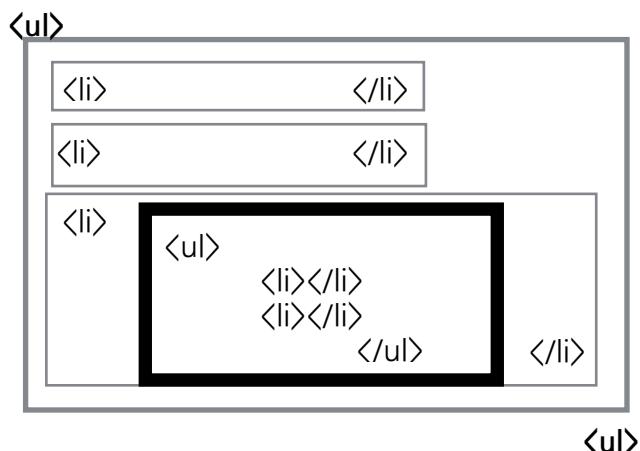
dd

- 정의형 항목의 설명 (definition description)
- 블록 요소
- 인라인 요소와 텍스트, 또 다른 블록 요소를 포함
- 여러 개의 dd도 가능

부모자식 관계
 형제 관계: 같은 부모를 가짐
 종속관계: 큰 부모 안에 자식과 손자를 한번에 지칭

<code><dl></code> <code><dt> 제목</dt></code> <code><dd>설명 </dd></code> <code></dl></code>	<code><dl></code> <code><dt> 제목</dt></code> <code><dd>설명 </dd></code> <code><dd>설명 </dd></code> <code></dl></code>	<code><dl></code> <code><dt> 제목</dt></code> <code><dd>설명 </dd></code> <code><dt> 제목</dt></code> <code><dd>설명 </dd></code> <code></dl></code>
---	--	---

목록안에 하위목록 만들기



- 브라우저
 - A. Chrome
 - B. Firefox
 - C. Opera
- 에디트 프로그램
 - A. Dreamweaver
 - B. Notepad
 - C. Editpuls

목록 문제풀기

```

<ul>
  <li>브라우저
    <ol type="A">
      <li>Chrome</li>
      <li>Firefox</li>
      <li>Opera</li>
    </ol>
  </li>
  <li>에디트 프로그램
    <ol type="A">
      <li>Dreamweaver</li>
      <li>Notepad</li>
      <li>Editpuls</li>
    </ol>
  </li>
</ul>
  
```

브라우저 렌더링 -> 콘텐츠
outline -> 구조

구조를 위한 요소

outline 정확하게

주인정보 header
주요서비스 nav
주내용 main
부가내용 aside
저작권, 서비스 정보 footer

outline에 영향 X ????

1. 시멘틱 웹 이란?

마크업에 의미 부여를 하는 요소이다.

기계적인 검색 엔진은 어떠한 태그가 어떠한 기능을 하는지 분별할 수 없고 웹 페이지에서 데이터를 효율적으로 추출 할 수 없다. 이를 해결하고자 특정한 태그에 의미를 부여해서 웹 페이지를 만드는 시도가 시작되었는데 이를 시멘틱 웹이라고 표현한다.

시멘틱 웹은 유저 에이전트(웹브라우저, 스크린리더, 검색엔진등)에 섹션의 목적을 알려줄 수 있다.

HTML5는 시멘틱 태그를 사용해 시멘틱 웹을 기본적으로 구현해야 한다.

2. 시멘틱 태그 사용 시 주의 사항

CSS 사용 시 문제가 있으면 display 속성을 block로 설정하여 사용한다.

IE8 이하 버전에서는 문제가 생기므로 shiv 파일을 사용해야 된다.

3. 사용 시 좋은 점

- 검색 엔진에서 필요한 부분만 쉽게 찾아서 검색 할 수 있다.
- 스크린 리더 같은 웹 보조기구에서 좀 더 정확하게 웹 문서를 분석하고 전달 할 수 있다.

역할에 따른 시멘틱 요소

<header> </header> 사이트의 제목 h1
• 소개 및 네비게이션의 목적으로 사용되는 그룹을 묶는 컨테이너
• 위치는 꼭 위에 있지 않아도 된다.

<footer> </footer> hn요소 포함 하면 X outline에 포함X nav 요소 포함 X
페이지를 만든 사람, 저작권 정보, 연관된 콘텐츠에 대한 링크, 블로그의 포스트의 작성자, 발행일시 등에 쓰인다.
저자나 편집자의 연락처는 footer 안의 <address>로 표현되며 저작권은 <small>로 표현된다.

<nav> </nav> GNB
• 주로 네비게이션(navigation)에 관한 정보 포함.
• 다른 페이지나 페이지의 다른 부분으로 이동하기 위한 일을 하는 그룹
• 주로 header에 속하지만 위치 보다는 내용이 중요하고 푸터(사용할 수는 있지만) 사용하지 않는 걸로 한다.

독자적 사용도 가능

<aside> </aside>
• 부가 콘텐츠 사용
• 사이드바와 같은 느낌이다.
• 본문의 내용과는 관계 없는 콘텐트를 추가 할 때 사용되며, 장소와는 상관없다.
• 주로 웹사이트의 본문 내용을 구성하고 남는 내용으로 사이드 바를 구성한다.
• 인용문, 광고, 블로그에서 blog roll 같은 내비게이션 요소의 그룹

main = 주내용
main>내용요소 가능
내용요소>main 가능
outline에 포함X -> 목차에 포함 X

내용에 의한 구분 ~main / outline에 영향을 준다

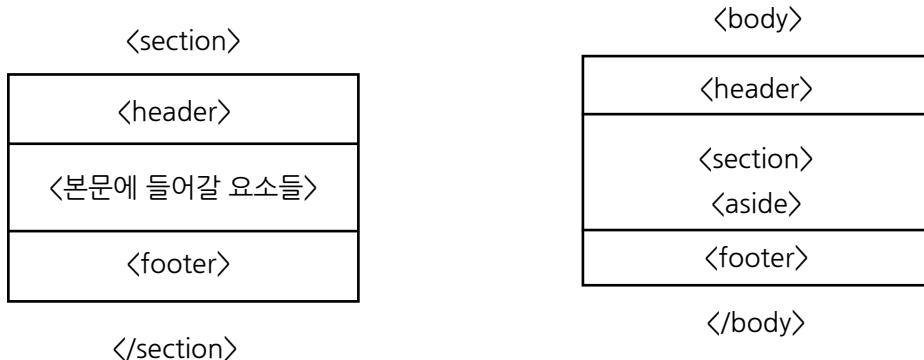
<section> </section> 개별적인 (도용이 불가능한) 내용 / 큰 내용 안에 자잘하게 나눌 때
• 테마 별로 연관된 콘텐츠를 묶는 데 사용되는 컨테이너.
• <section> 안에는 꼭 제목을 넣어야 한다. 왜냐면 연관된 주제의 콘텐츠이기 때문.
• 일반적으로 <section>에 묶여 있는 콘텐츠는 웹상에 배포를 하여서는 안된다.

<article> </article> 포괄적이고 큰 내용 / 자잘한 내용을 하나로 나눌 때
• <section>의 특화된 모습이라고 한다. 콘텐츠의 독립적인 부분으로 보이며, 그 자체로 의미가 있다. 또한 다른 사이트에 붙여 놓아도 그 하나로 의미가 있다. 제목 필요!

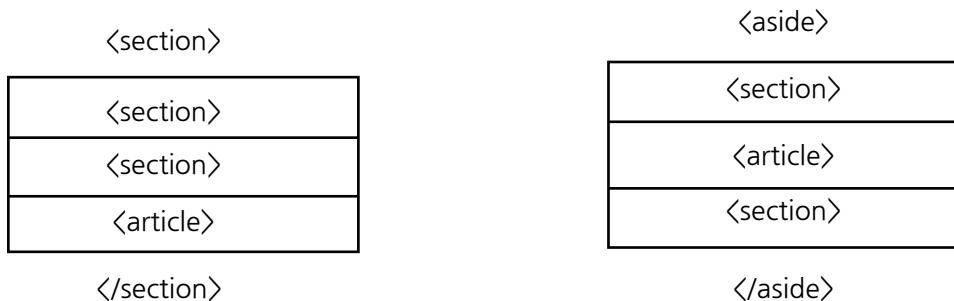
- 블로그 게시글, 뉴스, 짧은 의견, 리뷰, 혹은 포럼 글, 주식 시황, 계산기, 시계, 날씨 위젯등과 같은 위젯등이 포함된다.
- <section>과 다른 점은 <article>로 묶여 있는 콘텐츠는 웹 상에 배포되거나 재 사용되어도 상관없다.
-

확장형 의미

<header>와 <footer>사용법



<sesction>과 <aside>사용법



공간분할 태그

내용이나 기능이 없다
그저 그릇 -> CSS, 스크립트

div	block형식으로 공간 분할	큰 박스 -> 여러 요소
span	inline형식으로 공간 분할	TEXT

• outliner

HTML5에서는 정보 구조를 명확히 할 수 있도록 '아웃라인 알고리즘'(Outline Algorithm)이라는 개념이 도입되었음
아웃라인 알고리즘은 웹 페이지의 정보 구조를 판별할 수 있는 개념으로, 책의 목차와 비슷.

HTML 5에서 추가된 많은 요소들은 대부분 아웃라인 알고리즘과 관련이 있으며 그 중에서도 직접적으로 아웃라인을 구성하는 요소에는 헤딩 콘텐츠, 섹션 콘텐츠 그리고 섹션 루트 요소 등이 있음.

- <https://gsnedders.html5.org/outliner/>

- 1. href -> 연결
- 2. src -> source 가져오기
- 3. poster -> video

경로지정하기

- 1. 서버 경로
- 2. 절대 경로
- 3. 상대 경로

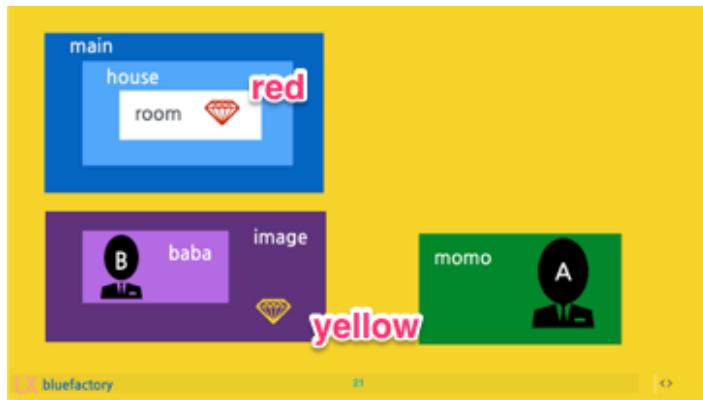
1 절대경로 ('<http://도메인/폴더명/파일명>')

```
<a href="http://sample.com/html/sample.html"> go sample site</a>
```

2 상대경로 ('[폴더명/ 파일명](#)'), 현재 문서의 위치를 기준으로 문서 연결

현재폴더 : ..
상위폴더 : ..

```
<a href="html/sample.html"> go sample site</a>
```



a가 red 다이아몬드를 가져올 수 있는 경로는

b가 red 다이아몬드를 가져올 수 있는 경로는

a가 yellow 다이아몬드를 가져올 수 있는 경로는

하이퍼링크:<a> ~ click 이벤트 -> href 주소 (**클릭을 할 수 있는 콘텐츠가 있어야 함)

- a 요소는 텍스트나 이미지 콘텐츠에 링크를 설정할 때 사용합니다.
- 인라인 요소 (인라인 요소와 텍스트를 포함)(html5에서는 블록요소도 포함 가능하다.)
- a 요소의 href 속성값에 #name(id) 속성을 지정하면, name과 id 속성으로 지정된 문서 내 특정 위치를 이동한다.

속성

href	링크 주소 지정	target : default 속성 _self : default 값	title : 공통 속성
id	앵커 식별자 지정		
title	링크의 보충정보를 표시, 대부분의 브라우저에서 title 속성에 지정한 값이 툴팁으로 표시		
target	링크된 문서를 어떤 창에 열 것인지 지정(_blank / _self)		
-> 연결 렌더링		_blank :연결 문서를 새 창에 연다(default값). _self :연결 문서를 기존의 창에 연다.	

```
<p> <a href="파일명 또는 URL" target="_blank" title="대체설명">텍스트 또는 이미지</a></p>
```

E-mail ('mailto:' 프로토콜로 시작)

- > 사이트 특정 페이지
- > 파일 연결
브라우저 렌더링 O -> Load E jpg, pdf
브라우저 렌더링 X -> 다운로드 E zip, hwp
- > 미디어 연결 -> 미디어 사용 연결
email -> mailto : 이메일 주소
tel -> tel : 전화번호

폼컨트롤 요소???

```
<a href="mailto:web@smapple.com">web@sample.com</a>
```

문서, 파일 (실행가능한 파일은 실행되거나, 다운로드 됨)

```
<a href="http://sample.com/html/sample.gif"> sample 다운로드</a>
<a href="http://sample.com/html/sample.zip"> sample 다운로드</a>
```

요소에 이름 정하기

- 아이디(id)
- 클래스(class)
- name: 특정한 형식의 아이템에 넣기 주요 프로그래머 작업 시 사용

ID(아이디)

- 중요한 곳에 붙인다.
- 1html문서->1의 이름
- #

```
<h1 id="main">text</h1> -> h1#main
```

Class(클래스)

- 여러곳에 같은 이름 사용 할때
- 덜 중요한 곳에
- .

```
<p class="main">text</p> p.main
```

식별자

이름 만들기

- 알파벳과 숫자 가능
 - 첫글자는 무조건 알파벳
 - 공간문자 사용불가
 - 대소문자 구분
 - 특수문자는 '-' '_' '\$'
- | | | |
|--------|---|-------|
| main01 | main | back |
| box01 |  | Box01 |

ID	CLASS
1 html -> 같은 이름 X	1 html -> 같은 이름 허용
1요소에 1개의 아이디	1요소에 여러개 클래스 이름 지정 가능 E) class= "이름1 이름2 이름3"
#	.
	우선권이 더 높다

mainTitle *
main_title *
main-title

이름 가독성, 이름 성격 고려하여 이름 정하기
* 사전 찾지 말기

main 01 01main

책갈피 사용하기(문서내 특정 위치로 이동하기)

특정지정연결:같은 페이지

- 원하는 곳에 이름 지정(id 넣기)
- href속성에 아이디 이름 지정

```
<a href="#아이디명">text</a>
```

클래스 사용 X

a -> 경로 -> 위치

- 위치 지정 : 요소에 이름을 붙임임
- 1. id : 이름 지정
- 2. class : 이름 지정
-
- 3. name : 필요한 값 설정, 요소의 설정

특정지정연결:다른 페이지

- 이동을 원하는 페이지에 이름 지정(id 넣기)
- href속성에 화일경로와 아이디 이름 지정

```
<a href="화일경로.html#아이디명">text</a>
```

```
<a href="..//html/main.html#top">text</a>
```



tabindex 사용하기 -> tab 순서 지정

```
<ul>
<li><a href="http://naver.com/" title="naver">naver</a></li>
<li><a href="http://google.com/" title="google">google</a></li>
<li><a href="http://daum.net/" tabindex="2" title="daum">daum</a></li>
<li><a href="http://nate.com/" tabindex="1" title="nate">nate</a></li>
</ul>
```

- [naver](#) 3
- [google](#) 4
- [daum](#) 1
- [nate](#) 2

1. 가상주소 -> #
``
 -> 필수속성
`cf> 가상 값 -> #`
 2. 웹접근성 : 마우스가 없어도 링크 연결을 할 수 있어야 함 -> `focus` -> 링크 이동

`<style> </style>` -> Internal -> 실무 사용 X

미디어콘텐츠

이미지 -> map
비디오
오디오

이미지:

웹 문서에 텍스트가 아닌 이미지 개체를 삽입할 때 사용
빈 요소(empty element)

속성

생략 불가	src 이미지화일의 경로
	alt 이미지가 보이지 않는 환경에서 대체 텍스트(alternative text)를 제공하기 위한 목적으로 사용하며, 이미지와 동등한 정보를 제공해야 합니다. 설명글 -> 오디오 브라우저 / text 브라우저
	width, height 이미지의 가로/세로 크기 지정(페이지의 로딩 진행시 이미지의 가로/세로 크기만큼 영역을 확보) 최근에는 스타일시트에서 너비와 높이를 입력 하므로 tag에서의 width, height는 잘 사용하지 않는다.

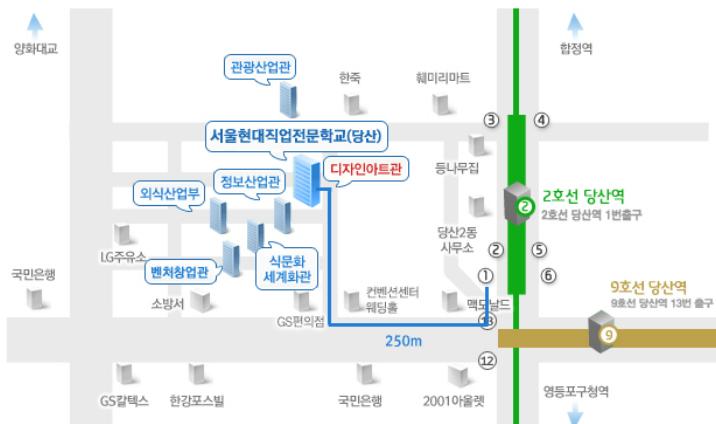
```
<p>

</p>
```

css 설정 불가능
- 반응형 웹 -> 가변
이미지 -> css

alt속성 사용법

이미지로 만든 약도의 표현



이미지 요소 / 비디오
요소 크기
- 기본값 = 이미지 크기
- width : 기준
- height : width에 맞춰서

<<브라우저 -> 표준 이미지>>

```
<p>
```

``

```
</p>
```

이미지의 alt가 길때 사용

<figure>요소 사용하기

+ 인용문구 (blockquote)

이미지, 그래프, 사진, 예제 코드, 비디오, 일반 텍스트 등을 하나로 묶어 블록을 형성하고 캡션을 포함시켜 서로의 관계를 명확히 구조화 시킨다. 반드시 메인이 되는 문서 안에 삽입될 필요는 없으며 페이지 옆이나 참조용 별도 페이지등에 넣어도 크게 상관없는 내용을 지정할 수 있다.

<figure>

data종류요소

<figcaption>data관련 내용 또는 제목</figcaption>

</figure>

figcaption이 있더라도 alt는 기본속성이므로 꼭 써야함

```
<figure>
  
  <figcaption>
    <ol>
      <li>기증신청 및 접수</li>
      <li>기증품 수거</li>
      <li>물품분류 및 판매가격 책정</li>
      <li>매장/행사장 판매</li>
      <li>배분사업을 통한 수익나눔</li>
    </ol>
  </figcaption>
</figure>
```

이미지map사용하기:

- img src="" alt="" usemap = "#이름"
 - map -> name="이름"
 - area -> 모양 위치 연결정보

이미지맵 유튜브
<https://www.image-map.net/>

<map></map>

이미지의 일부 영역에 링크를 두어 사용자에게 해당 정보를 제공할 수 있는 페이지로 이동하도록 만드는 프로그램 기법

Attribute	Value	Description
<u>name</u>	mapname	img요소의 usemap값

<area> 영역표시 빈요소

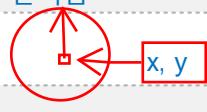
Attribute	Value	Description
<u>alt</u>	text	설명
<u>coords</u>	coordinates	위치지정
<u>download</u>	filename	클릭 시 다운로드 파일 위치 지정
<u>href</u>	URL	경로
<u>hreflang</u>	language_code	언어지정

media	media query	media/device
<u>rel</u>	alternate author bookmark help license next nofollow norereferrer prefetch prev search tag	연결 파일과의 관계
<u>shape</u>	default rect circle poly	모양
<u>target</u>	_blank _self	
<u>type</u>	media_type	media type

```

<map name="Map" id="Map">
    <area shape="rect" coords="40,78,113,148" href="http://www.nate.com" alt="뽐질마크" target="_blank" >
    <area shape="circle" coords="201,169,54" href="#" >
    <area shape="poly" coords="296,17,236,53,254,110,329,109,348,54,296,17" href="http://www.naver.com"
target=_self alt="개발자" >
    <area shape="poly" coords="309,187,252,277,368,286,307,188" href="#" >
    <area shape="poly" coords="84,218,132,214,158,263,133,311,88,310,63,260,86,218" href="#" >
</map>
```

coords 모양 -> 좌표값 <- 그림판

값	설명	x1, y1
x1,y1,x2,y2	(shape="rect")	 x2, y2
x,y,radius	(shape="circle")	 반지름 x, y
x1,y1,x2,y2,...,xn,yn	(shape="poly")	

media 사용

```

<map name="planetmap">
    <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm" media="screen and (min-color-index:256)">
```

</map>

비디오 태그: <video></video>

코덱 -> 파일 생성

- 웹 브라우저에서 영상을 재생할 수 있게 해준다.
- 웹 브라우저마다 지원하는 동영상 파일 형식이 다르다.
- 사용 가능 포맷
 - MP4 = MPEG 4 files with H264 video codec and AAC audio codec
 - WebM = WebM files with VP8 video codec and Vorbis audio codec
 - Ogg = Ogg files with Theora video codec and Vorbis audio codec

플레이시에만 트래픽
???????????????

<code>src</code>	비디오 파일의 경로지정 video태그 or source 태그 중 택 1
<code>poster</code>	비디오 준비 중일 때의 이미지 파일 경로 지정 비디오 준비중 or autoplay가 아닐 때
<code>preload</code>	비디오를 재생하기 전에 데이터 로드에 대한 상황 지정 <code>auto:자동로드</code> <code>metadata: 정보만 로드</code> <code>none: play전에 로드하지 않음</code>
비디오 ->(None) -> 설정/종류/환경 정보 -> (metadata) -> 데이터 -> 렌더링 -> 비 디오 플레이 (auto)	~ 데이터 정보 업로드 문제 시점
<code>autoplay</code>	비디오를 자동 재생할지 지정
<code>loop</code>	비디오를 반복할지 지정
<code>controls</code>	비디오 재생 도구를 출력할지 지정
<code>muted</code>	소리재생하지 않도록 함.
<code>width</code>	비디오의 너비를 지정
<code>height</code>	비디오의 높이를 지정

<video poster="images/puppy.jpg" width="400" height="300" controls src="video/puppy.mp4">

브라우저와 동영상 포맷과의 관계

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES from Firefox 21 from Firefox 30 for Linux	YES	YES
Safari	YES	NO	NO
Opera	YES From Opera 25	YES	YES

브라우저에 맞는 동영상 선택하여 플레이어 하기

<source>

<code>src</code>	비디오 파일의 경로지정
<code>type</code>	MIME-type 지정

포맷의 MIME-type



Format	MIME-type
MP4	video/mp4
WebM	video/webm
Ogv	video/ogg

```
<video poster="images/puppy.jpg" width="400" height="300" controls>
  <source src="video/puppy.mp4" type="video/mp4">
  <source src="video/puppy.webm" type="video/webm">
</video>
```

오디오태그: <audio></audio>

기본: 웹브라우저에서 플러그인 도움 없이 음악을 재생 할 수 있게 만들어 주는 태그

속성	내용
src	음악 파일의 경로 지정 audio 태그 or source 태그 중 택 1
preload	음악을 재생하기 전에 데이터로드에 관한 설정 (none, auto, metadata)
autoplay	음악을 자동 재생할 지 지정(Boolean속성)
loop	음악을 반복할지 지정
controls	음악 재생 도구를 출력할지 지정
muted	소리재생하지 않도록 함.

```
<audio src="audio/test-audio.ogg" controls autoplay></audio>
```

브라우저와 오디오 포맷과의 관계

음원

Browser	MP3	WAV	OGG
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

브라우저에 맞게 로딩하기

<source>

- audio, video에 사용되는 소스 태그

src	비디오 파일의 경로지정
type	MIME-type지정

포맷의 MIME-type

Format	MIME-type
MP3	audio/mp4
WebM	audio/webm
Ogg	audio/ogg

<track>

자막 ~ 웹접근성 <- 사용자가 원하는 컨셉

- video,audio태그에 자막또는 설명 글 넣기

Attribute	Value
<u>default</u>	default
<u>kind</u>	captions chapters descriptions metadata subtitles 자막
<u>label</u>	text
<u>src</u>	URL 자막 파일
<u>srlang</u>	language_code

```
<video width="320" height="240" controls>
  <source src="forrest_gump.mp4" type="video/mp4">
  <source src="forrest_gump.ogv" type="video/ogv">
  <track src="subtitles_en.vtt" kind="subtitles" srclang="en" label="English">
  <track src="subtitles_no.vtt" kind="subtitles" srclang="no" label="Norwegian">
</video>
```

vtt -> 자막 확장자

stylesheet 사용하기

- CSS는 웹 페이지에 “디자인”이라는 시각적 가치를 부여하기 위한 언어입니다.
- HTML 문서의 레이아웃과 스타일을 정의합니다.
- W3C에서 표준안을 만들어 내고 있으며 1996년 CSS Level 1 이후, 현재 CSS Level 2.1에서 CSS Level 3까지 보급되어 있음.
- W3C 표준 : CSS Level 2 Revision 1 (CSS2.1), 모바일 : CSS3 (CSS2.1 기술 포함됨)

서식

```
선택자 {속성      :      값;
         속성      :      값;
         속성      :      값;      }
```

- selector (선택자) : 스타일을 적용하는 대상
- property (속성) : 스타일의 종류
- value (값) : 속성이 가질 수 있는 값

```
body {color:gray;}
div {border:1px solid red;background-color:gray;}
```

적용방식

~~Inline Style Sheet~~

- 요소에 직접 스타일 적용
- 태그와 함께 CSS가 코딩되어 있어 CSS 분리원칙에 위배됨
- 단, 디버깅용(오류 찾기)으로만 사용
 <태그 style="속성:값;">

~~Internal Style Sheet~~

html 문서의 <head></head> 사이에 CSS를 선언하는 방식 (meta 뒤에, script 앞에) 마찬가지로, CSS 분리원칙에 위배됨

```
<head>
  <style type="text/css">
    selector { 속성 : 값; }
  </style>
</head>
```

External Style Sheet:<link> 사용 (가장 권장)

<link> 요소로 외부 파일 사용하기

```
<head>
  <link rel="stylesheet" type="text/css" href="css파일명" />
</head>
```

```
@charset "utf-8";
p{
  background: beige;
  color: gray;
}
p span{
  font-weight: bold;
  color: blue;
}
```

External Style Sheet:<@import> 사용하기

여러개의 CSS파일 연결해서 사용

@import url (경로/파일명.css); 또는 @import “파일명.css”;로 작성

```
<style>
    @import url(경로) <- 세미콜론으로 닫아주기
</style>
```

```
@charset "utf-8";
@import url(sample.css);
```

Basic

CSS 주석

- 작성자의 관리와 편의를 위한 찾아보기나 메모
- “/*”와 “*/” 사용.
- 주석으로 작성은 내용은 스타일에 영향을 미치지 않는다.
/* comment */

환경 영향?????

선택자-1

선택자 {
속성 : 값 ;
속성 : 값 ;
}

기본선택자

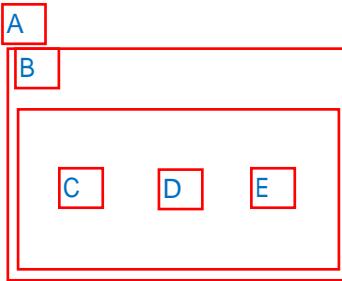
요소마다 이름을 정하지 말고 할때
마다 추가하는 것이 좋음

선택자	기능설명
요소	특정 요소 선택
#id명	특정 id명을 가진 요소 선택
.class명	특정 class명을 가진 요소 선택
요소1 요소2	요소1 안의 있는 모든 요소자손 요소2
부모요소1>자식요소2	부모 -> 자식 : 종속부모요소1에 있는 자식요소2
형제요소1+형제요소2	기준형제 + 인접형제 세로로 아래에 있는 형제간 선택
형제요소1~형제요소2	기준형제 ~ 아래에 있는 형제들 중에 요소2
선택자1, 선택자2	여러 선택자 한꺼번에 선택
*	모두 선택

타입선택자

요소명을 선택자로 사용

```
h1 {font-size:20px;}  
ul {margin:0px;}
```



A > B
A > B > D (우선권)
A D

B+D
B~C

* 우선권 높이기
- 아이디
- 자식선택자

class 선택자 (요소명.class명 : 요소명 생략가능)

- 클래스 이름으로 선택하기, 문서내에서 여러 요소를 한 번에 선택할 수 있다.

```
.div.note { color : blue ;}  
.clr {clear:both;}
```

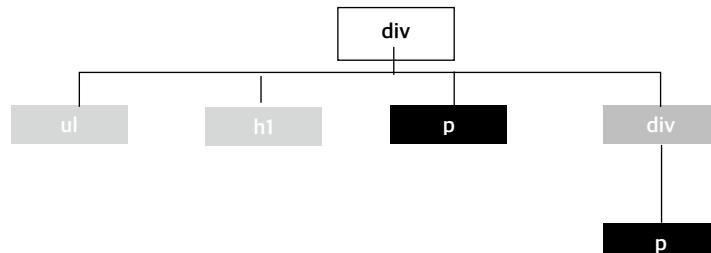
id 선택자 (요소명#id명 : 요소명 생략가능)아이디 이름으로 선택하기

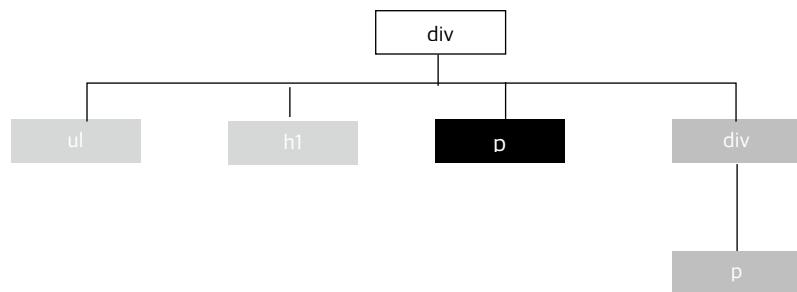
```
ul#gnb {list-style-type : none ;}  
#bodyContent {width:100%; height:500px;}
```

자손,종속선택자

- 특정 엘리먼트 하위의 엘리먼트를 지정할 때 사용
- 공백(space)으로 구분함
- ID, CLASS등 여러 선택자와도 함께 사용 가능

```
div p {height:30px;}  
div>p {color:blue;}
```

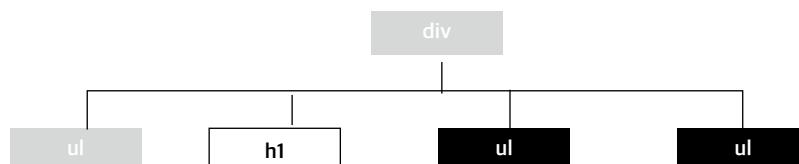
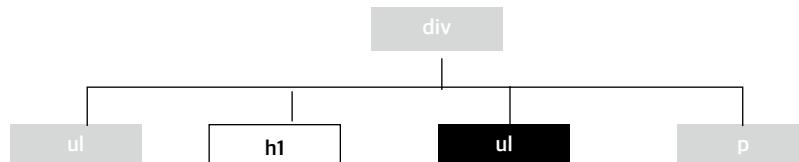




형제 이웃선택자

기준이 되는 요소 와 형제 또는 다음에 오는 요소 선택

```
h1+ul{background-color:red;}
h1~ul{background-color:red;}
```



그룹 선택자

- 선택자와 선택자를 콤마(,)로 구분하여 그룹화 하고 복수의 요소에 같은 스타일을 적용합니다.
- class 선택자나 id 선택자, 속성 선택자, 하위 선택자, 자식 선택자, 인접 선택자 등을 지정할 때도 콤마(,)로 구분하여 그룹화 할 수 있다.

```
#wrap, p.note, blockquote p, h1[title] { color : red ; }
Color
```

css 1, 2
----- html5
css 3 : 추가, 변경

Keyword (색상명, 17가지)

- black, silver, gray, white, maroon, red, purple, teal, fuchsia, green, lime, olive, yellow, navy, blue, aqua
color:lime;
background-color:maroon

샘플참조: https://en.wikipedia.org/wiki/Web_colors(구글에서 webcolor검색)

http://www.w3schools.com/colors/colors_picker.asp(구글에서 webcolor검색)

RGB값 (16진수, 6자리)

color : #ff1f00 ;

RGB값 (16진수, 3자리) 상끼리 같을 때 축약형

color : #ff0 - (#ff1f00 = #ff0)
E) #00aadd = #0ad

RGB값 (10진수)

color : rgb(255,0,0) ; % 방식보다 더 많이 사용됨

RGB값 (%)

color : rgb(100%,0,0) ;

주의할 점은 값이 '0' 이더라고 '%'를 생략해서는 안된다.

0은 일반적으로 단위 생략하지만, 예외 : COLOR

웹 안전컬러 -> 브라우저에 문제가 없는 컬러 : 현재는 관련 X

color

- 키워드 값 (17 -> 100)
- 16진수 (0 ~ 9 a ~ f)
- rgb -> 빛
- 양 UP = white (#FFFFFF) / 양 DOWN = BLACK (#000000) / 세쌍이 같거나 6자리가 다 같으면 Gray
- cymk ->인쇄 : X

white	#ffffff	rgb(255,255,255)	rgb(100%,100%,100%)
black	#000000	rgb(0,0,0)	rgb(0%,0%,0%)

css3 추가된 칼라의 속성 투명도 -> alpha

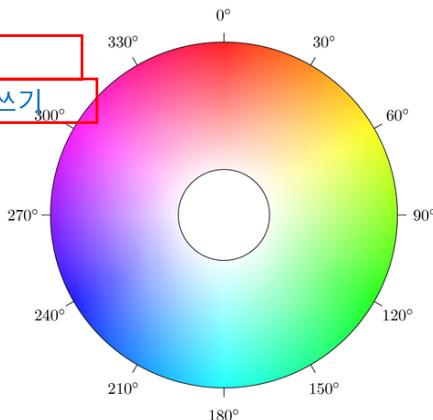
rgba(red,green,blue, alpha)

- red green blue alpha로 칼라 지정
 - alpha: 0~1 (0.5=50%) 0 = 0%, 1 = 100%
- background-color:rgba(0,0,0,0.5)

hsl(hue,saturation,lightness) ~ 색의 삼원색

- hue: 색상환에서 0-360까지의 값으로 표현 deg 단위 꼭 쓰기
- 채도:%로 나타냄 (0:무채색~100%:순색)
- 밝기:%로 나타냄 (0~100%)

background-color:hsl(0, 0%, 78%)



업그레이드 버전 : , 가 없음

rgb (red green blue / %)

hsla(hue saturation lightness / %)

* 하지만 아직 사용X 너무 최신 버전임

투명도 -> opacity : 0 ~ 1

-> 상속 문제가 있다. : 그요소의 자식요소마저 모두 다 투명해짐

단위

1. 절대단위

- cm, in, mm, pt, pc

(pt : 포인트 : 서양에서 많이 사용 E) 1/72

(pc : 포인트를 기준으로 가이드를 잡을 때 많이 사용)

(12pt -> 1pc)

2. 상대단위 : 환경이나 미디어 크기에 따라 변화

- px, %, em, rem, vh, vw

(px : 해상도에 따라서도 달라진다)

- ppi = pixel per inch E) 300ppi = 인쇄 / 72ppi = 우리쪽 -> 적절한 해상도와 크기의 이미지를 사용해야 함 / 아무리 해상도가 높아도 쓸모 X)

(% : 나의 크기 / 기본 크기 * 100 E) 250px / 500px = 50%)

(em -> font-size : 기본크기가 부모크기 E) body = 16px / 여백, line-height ... : 선택자에서 사용되는 것들 : 자신의 폰트 사이즈가 기준 // 나의 크기 / 기본크기 E) 250px / 500px = 0.5)

html 기준 : rem

head 기준 :

body 기준 : em -> body = 1em = 16px

* hn 요소 = 2em

타이포그래피

글자의 종류 선택

- 기본 : 사용자의 폰트 (Windows -> font) 사용
용 (대표 페밀리체 사용)

font-family - 외부 폰트 사용용 : 트래픽 조심

font - 글자정의

text - 단락이나 text그룹의 성격을
말할 때 사용됨

- 글자의 모양을 설정한다.
- 하나 이상의 글꼴을 선언할 수 있으며 글꼴은 콤마(,)로 구분한다.
- 주의사항
 1. 한글글꼴은 “ ” 반드시 필요하다. 한글 글꼴 뒤에는 꼭 글꼴의 영문이름을 추가한다.
 2. 영문글꼴이더라도 글꼴이름에 공백이 있는 경우 “ ” 필요하다.(ex:PT mono, Malgum Gothic)
- 해당 글꼴이 없을 경우 제일 마지막에 대표 페밀리체가 렌더링되며, 만약에 폰트 페밀리를 지정하지 않으면, 사용자 컴퓨터에 설정된 브라우저 글꼴로 렌더링 된다.

형식: **font-family: 1순위글꼴페밀리명, 2순위글꼴페밀리명, 3순위글꼴페밀리명, …, 대표페밀리명**

font-family : "나눔고딕", "돋움", Dotum, sans-serif;

한글폰트, 공간문자 들어간 것은 따옴표 생략 X

대표 페밀리란?

serif(인쇄물), sans-serif(온라인) cursive, fantasy, monospace

<https://fonts.google.com/>

serif: 문자 끝부분에 장식용 가시와 고리모양고상하고 고전적인 느낌을 준다.

Times, Times New Roman, Georgia, 바탕, 궁서

sans-serif: 고리모양이 없다. 깔끔하고 읽기쉽다.

Verdana, Arial black, Arial, 굴림, 돋움

Cursive(흘림체): 손으로 쓴것같은 폰트 (필기체)

Calligraphic script. Sanvito

Fantasy 화려한 장식

Impact

Monospace 일정한 폭의 문자 폰트로 이루어져 타자기로 타이핑한 것 같은 느낌

Courier, Prestige

font-size 1 웹 문서의 글자 크기를 지정한다.

- 방식:

기본 16px

키워드: xx-small, x-small, small, normal, large, x-large, xx-large
절대, 상대크기: pt, px, %, em

font-size: 키워드 또는 길이(상대, 절대, inherit)

```
font-size:12px;  
font-size:1.5em;  
font-size:150%
```

font-weight

- 글꼴의 굵기를 표현
- 방식:

default = normal

하지만 hn 요소는 bolder가 기본

- 키워드: normal, bold, bolder, light, lighter
- 숫자: 100(lighter)-400(normal)-600(bold)-900(bolder)

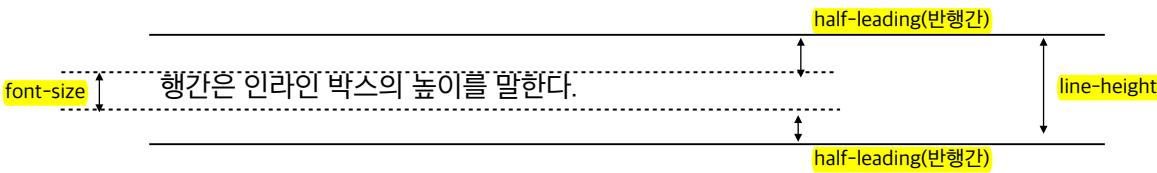
font-weight: 키워드, 또는 100-900, inherit

```
font-weight: normal;
font-weight: bold;
font-weight: 900;
```

~ 가독성

line-height

- 문단의 행간을 지정 할 때 사용한다.
- line-height는 font-size로 지정된 글자 크기를 포함하여 글자의 위쪽과 아래쪽 여백까지를 말한다.



- font-size보다 작게 지정할 경우에는 글자가 겹칠 수 있다.
- 보통은 120%로 신문등 글자가 많은 경우(신문)는 150%로 사용되는 경우가 많다.
- 수치는 짹수로 사용하는 것이 좋다.

테스트가 많을 경우 1.5이상
해상도가 높은 디바이스들을 위해 1.6 ~ 1.8

line-height: normal, 실수, 길이, %, em, inherit

line-height:	normal;	기본값 = 1.2
line-height:	1.6;	normal의 1.6배
line-height:	16px;	px, %, em, 실수 (em 단위 생략)
line-height:	200%	- 자신의 폰트 사이즈 기준 (Default = 1.2 : 폰트사이즈의 120% = 1.2em)

font-style

기울기

- 글꼴의 스타일을 지정할 때 사용합니다.

두개의 차이는 없음

font-style: normal, italic, oblique;

```
font-style : normal ;
font-style : italic ;
font-style : oblique ;
```

default = normal
하지만 address 요소는 기울여지는 게 기본

font-variant

- 영문 글꼴의 소문자를 대문자로 변형할 때 사용합니다.
- 값: normal | small-caps 소문자의 크기에 맞춰 글자만 대문자로 변경

font-variant: normal, small-caps;

```
font-variant : normal ;
font-variant : small-caps ;
```

if I had thought about it I might have been scared.
IF I HAD THOUGHT ABOUT IT I MIGHT HAVE BEEN SCARED.

font 단축형

- font-family, font-size, line-height, font-weight, font-style, font-variant 의 6가지 속성을 한번에 선언할 때 사용하는 대표 속성이다.
- 선언순서를 지켜야 하고, **font-size와 font-family**는 꼭 선언해야 한다.

default 값 생략 가능

```
font:[ font-weight | font-style | font-variant ] | [ font-size ] | [ /line-height ] | [ font-family ] ;
```

font : bold italic small-caps 12px /1.6 “돋움”,Datum,sans-serif;

text-indent**px, %, +, -**

- 문단의 첫 줄 들여쓰기를 적용(값 적용 가능값만큼 첫줄만 앞당겨짐)
- 블록요소에만 적용 가능 (인라인요소인 a, span태그에는 안먹힘)
- 음수도 가능하다. 하지만 음수인 경우에는 같은 값인 안의 여백이 필요하다.

text-indent:길이 | 퍼센트 | inherit ;

```
text-indent:50px;
text-indent:2em;
p{text-indent:-50px; padding-left:50px;}
```

text-align**-> block / 텍스트 외에도 모든 인라인 요소 정렬가능**

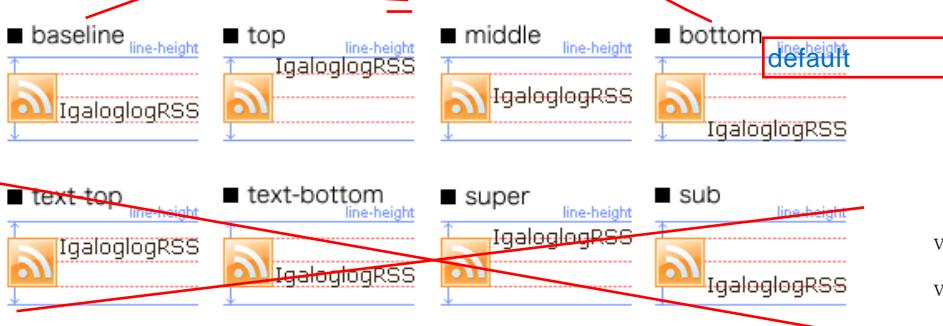
- 단락의 텍스트를 가로 기준으로 정렬할 때 사용 합니다.

text-align:left | center | right | justify | inherit ;**박스를
다채움**

```
text-align : left;
text-align : center;
text-align : justify;
```

vertical-align**-> inline****기준이 되는 요소에 넣기**

- 인라인 요소끼리의 세로 위치를 정렬할 때 사용된다.
- 인라인요소(이미지, 인풋, 스패n 등), 일부 블록요소(테이블의 th, td)에 사용가능
- 지정한 인라인요소 뒤에 오는 요소에 대한 세로정렬 가능

vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom | 길이 | 퍼센트 | inherit;

vertical-align:top;
vertical-align:middle;

text-decoration

- 텍스트에 밑줄을 긋거나 취소선 등의 효과를 추가할 때 사용한다.

디폴트**text-decoration:none | underline | overline | line-through | blink | inherit****ins del**

text-decoration : underline

`text-decoration: underline;`

text-decoration : line-through none;

`text-decoration: line-through;`

text-decoration :overline;

`text-decoration: overline;`

text-decoration :none;

`text-decoration: none;`

letter-spacing

문자 간격을 조절할 때 사용된다. 음수 양수 모두 사용가능.

가독성을 위해 0.5px,
0.02px로 줄이는 경우
우가 있다

`letter-spacing:길이 | normal | inherit`

`letter-spacing:1em;` (글자크기에 비례하여 조절)

`letter-spacing:-3px;`

word-spacing

letter보다 word를 조절하는 경우가 많음
- letter는 티가 많이 남

단어와 단어 사이의 간격을 조절

`word-spacing:길이 | normal | inherit`

`word-spacing:1em;`

`word-spacing:-3px;`

text-transform

영문 대소문자를 변환할 때 사용 합니다.

`text-transform : none | uppercase | lowercase | capitalize | inherit`

text-transform : none;

This is a paragraph(none)

text-transform : capitalize;

This Is A Paragraph(capitalize)

text-transform : uppercase;

THIS IS A PARAGRAPH(UPPERCASE)

text-transform : lowercase;

this is a paragraph(lowercase)

white-space

공백 문자를 처리하는 방법

pre:입력한 형태 그대로 브라우저에 렌더링

nowrap:무조건 한 줄로 출력됨. 엔터키는 들어가지만

`white-space : normal | nowrap | pre`

공간, 공백문자

- 규칙이 있는 것 E) 코드 -> html <pre></pre>

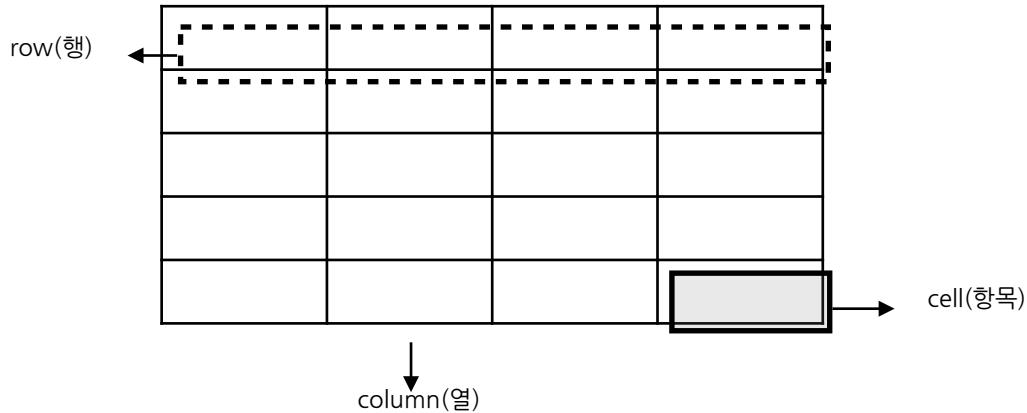
- 장식 -> white-space

테이블 사용하기

table -> 같은 내용 그룹 정리

1. 전체 레이아웃 (과거O - XHTML1.0 / 현재X) -> 저가 쇼핑몰에 남아있음

- 2차원의 격자 형태로 data를 정리 관리 할 때 사용한다.
- 테이블(table)은 크게 열(column)과 행(row), 그리고 셀(cell)로 구성된다.



셀(cell) : 표 안의 항목
 행(row) : 흉렬의 셀
 열(column) : 종렬의 셀

기본요소와 속성: <table>, <tr>, <th>, <td>, <caption>, summary

1. 기본형식

```
<table>
    <caption> 표의 제목</caption>
    <tr> <th> </th> <th> </th> </tr>
    <tr> <td> </td> <td> </td> </tr>
    <tr> <td> </td> <td> </td> </tr>

```

```
<table summary="테이블 요약설명">
    <caption>제목</caption>
    <tr> <th>제목 셀(header cell)</th> <th>제목 셀(header cell)</th> </tr>
        <tr> <td>내용 셀(data cell)</td> <td>내용 셀(data cell)</td> </tr>
        <tr> <td>내용 셀(data cell)</td> <td>내용 셀(data cell)</td> </tr>
    </table>
```

2. 기본요소와 속성 tr 먼저 다 만들고 나서 th, td 추가

tr

- 행을 정의하는 요소이다.
- 블록 요소이다.
- th, td 요소만 포함 가능하다.

th

th랑 td는 인라인 요소
는 아니지만 vertical
align이 들어감

- 제목 셀을 정의하는 요소이다.
- 굵은 글꼴로 중앙 정렬되어 표시된다.

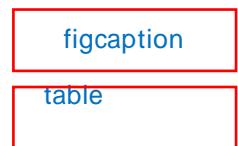
td

- 데이터 셀을 정의하는 요소이다.
- 일반적인 글꼴로 왼쪽 정렬되어 표시된다.
- 블록 요소이다.
- 인라인 요소와 텍스트, 또 다른 블록 요소 포함 가능하다.

caption(html5에서 삭제)

대체

- 표 제목을 지정하는 요소이다.
- table 요소의 시작 태그 바로 뒤에 기술한다.
- 표 위에 중앙 정렬로 표시된다.

figure**figure****summary**

- <table>요소 안에 사용되는 속성이다.
- 테이블 데이터의 내용을 요약한다.

주의:

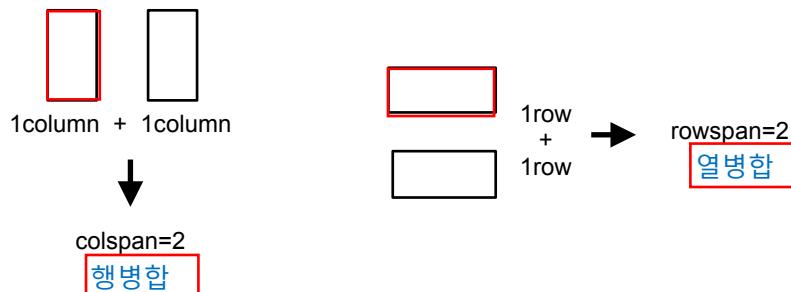
접근성 높은 테이블 데이터를 마크업할 때 테이블의 제목 및 테이블 데이터의 내용을 요약해서 알려줄 수 있는 caption 요소와 summary 속성을 사용하는 것이 좋습니다.

caption 요소에는 테이블의 내용을 대표할 수 있는 제목을 삽입하고, summary 속성에는 간단한 제목보다 테이블의 내용이 무엇인지를 설명해서 시각적으로 테이블 접근성이 쉽지 않은 사용자가 summary의 내용을 듣고 해당 테이블의 콘텐츠를 파악한 후 건너뛸 것인지, 탐색할 것인지를 선택할 수 있도록 해야 합니다.

cell 변형하기**rowspan, colspan**

- 셀 결합 속성
- rowspan - 수직방향으로 셀 결합
- colspan - 수평방향으로 셀 결합

	<td colspan="2">		</td>	
			<td rowspan="2">	

원리**행의 그룹:<thead>, <tfoot>, <tbody>**

** tbody는 생략 불가

thead - 머리 파트

tbody - 데이터 파트

tfoot - 데이터 파트 정리

thead : 표의 머리글
tfoot : 표의 바닥
tbody : 표의 본문

- 인쇄할 때 표가 여러 페이지에 걸친 경우 모든 페이지에 헤더행과 푸터행이 반복해서 출력된다.
- 행 그룹화 요소의 선언 순서는 thead, tfoot, tbody 순이고 thead 요소와 tfoot 요소는 테이블에서 한 번만 사용할 수 있습니다.
- 제목 행으로 지정된 thead 요소에는 반드시 제목 셀(th 요소)을 이용해서 해당 영역이 테이블의 헤더 영역임을 지정해야만 합니다.

옛날 방식 : thead -> tfoot -> tbody

```
<table>
  1 <thead><tr>
  3 <tbody><tr>
  2 <tfoot> <tr>
```

2010년 교통비		
월	일	금액
1월	1일	1,000
	2일	2,000
합계		3,000

```
</tr> </thead>
</tr>
</tr> </tbody>
</tr> </tfoot>
</table>
```

열의 그룹:<colgroup>, <col>

- 표의 열을 그룹화하기 위한 요소
- table 혹은 caption 요소의 바로 뒤 (tr, thead, tfoot, tbody 요소보다 앞)에 지정

colgroup: 열의 구조적인 그룹화하기 위해 사용

col: 열을 그룹화하여 공통 속성 및 스타일을 적용하기 사용

colgroup : css 적용 X

col : css 적용 O

<colgroup>

2010년 교통비		
월	일	금액
1월	1일	1,000
	2일	2,000
합계		3,000

↓ ↓ ↓

col col col

</colgroup>

<table rules="groups">

<colgroup span="2">

<col />

<col />

<col />

</colgroup>

<tr></tr>

</table>

col을 2개를 합쳤다는
것을 보여줌

scope

전세계 웹 브라우저 점유율

웹브라우저	점유율
익스플로러	38.79%
파이어폭스	30.13%
크롬	21.45%
그외	9.63%
합계	100%

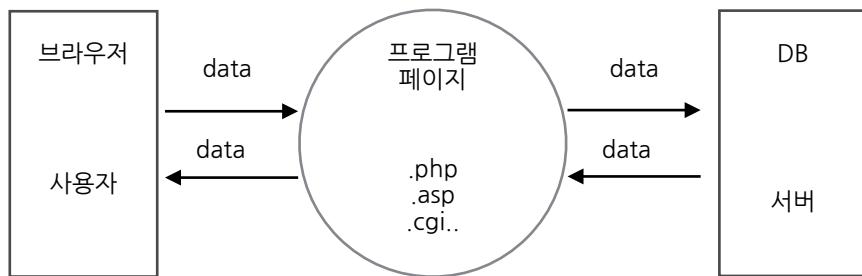
col

row

form

폼 요소는 웹 문서가 서로 상호작용을 할 수 있도록 하는 역할을 담당합니다.

사용자가 자료를 검색하기 위한 검색어 입력 상자나 여러 가지 항목에서 원하는 값을 선택할 수 있도록 제공하는 라디오 버튼 또는 체크 박스, 목록 형태로 펼쳐지는 목록 상자(combo box), 서버로 자료를 전송하기 위한 전송 버튼 등이 이에 해당됩니다.



<form></form> 속성

action : 폼을 전송할 URL 지정

method : 폼의 데이터를 전송하는 방법(get 또는 post를 사용한다.)

get

- URL 뒤에 파라미터 값을 붙여 전송
- 간단한 데이터 전송시 사용되지만 데이터가 주소 입력란에 표시되므로 보안유지 안됨.

post

- http 헤더에 숨겨져 서버로 전송
- 일정 크기 이상의 데이터 전송시 사용된다. 데이터가 주소입력란에 표시되지 않아 보안성이 우수하다.

폼 요소 그룹화 및 그룹 제목: <fieldset>, <legend>

fieldset: 여러 개의 폼 요소를 그룹화하여 좀 더 구조적으로 만들때 사용한다.

legned : fieldset의 컨트롤들이 어떠한 성격의 콘텐츠인지 제목 형식으로 알려준다. fieldset 요소의 바로 뒤에 한 번만 작성되며 필수 요소이다.

```

<form action="서버URI" method=""get 또는 post ">
    <fieldset>
        <legend>필수조건</legend>
        폼 컨트롤 요소
    </fieldset>
    <fieldset>
        <legend>선택조건</legend>
        폼 컨트롤 요소
    </fieldset>
</form>

```

레이블:<label></label>

- 폼을 구조화하고 접근성을 높일 수 있는 요소로 각 폼 컨트롤의 연관 관계와 설명을 추가하는 역할을 담당합니다.
- 웹표준을 지원하는 대부분의 최신 브라우저의 경우 label만 선택해도 폼 컨트롤을 선택할 수 있으며, 음성 브라우저의 경우 폼 컨트롤이 label과 인접하여 있지 않는 경우에도 인식할 수 있도록 지원합니다.
- 접근성을 높이기 위해 모든 폼 컨트롤 등에 label 요소를 사용하는 것이 좋습니다.
- 연결방법에는 명시적인 레이블과 암묵적인 레이블이 있다.

명시적인 레이블 (id와 for 속성 연결하기)

- label의 for 값 = 폼 컨트롤 요소의 id값

<p>

```
<label for="userName">이름</label>
<input type="text" id="userName" name="name" value="value" />
</p>
```

암묵적인 레이블

- label 요소에 폼 컨트롤 포함시키기

```
<p>
<label>이름
<input type="text" id="userName" name="name" value="value" />
</label>
</p>
```

text :	<input type="text"/>
password :	<input type="password"/>
checkbox :	<input type="checkbox"/> 체크박스 1 <input type="checkbox"/> 체크박스 2
radio :	예 <input checked="" type="radio"/> 아니오 <input type="radio"/>
button :	<input type="button" value="버튼"/>
submit :	<input type="submit" value="전송하기"/>
reset :	<input type="reset" value="취소"/>

입력요소:<input>

- 폼 안에 기본적인 컨트롤을 생성
- 인라인 요소, 빈 요소
- input 요소의 경우 type 속성값에 따라 폼의 종류가 결정된다.

```
<input type="컨트를 값" value="초기값" size="크기" id="식별자" name="변수명" />
```

<input>요소의 type

text	일반 텍스트 입력필드
password	비밀번호 입력필드 대부분의 브라우저에서 입력한 텍스트를 '*' 등으로 감추어 표시 일반 텍스트로 전송되기에 실제 데이터 전송 시에는 암호화 기술 필요
checkbox	복수 선택 가능한 체크
radio	한 개만 선택 가능한 라디오버튼
submit	송신버튼
reset	리셋버튼 (입력 내용 전부를 초기화)
button	범용버튼
image	송신 이미지버튼 (src 속성과 alt 속성 지정)
file	송신파일 선택필드
hidden	감춰진 필드 (화면에서는 표시되지 않으나 서버로 전송할데이터를 지정할 수 있음)

Form 2.0

text	search	검색어 입력
개인 주소	tel	전화번호 입력
	url	인터넷 주소의 입력
	email	e-mail 주소 입력
시간	date	날짜 입력
	week	주 차의 입력
	month	달의 입력
	time	시간의 입력

	datetime-local	지역 시간의 입력
숫자	number	숫자의 입력
	range	슬라이더 형식의 숫자 입력
기타	color	색상의 입력

name	컨트롤의 이름 (서버에서 처리할 데이터의 컨트롤에는 name 속성 필수)
value	컨트롤의 값
size	text, password 컨트롤의 가로 크기 size를 지정하지 않으면 컨트롤의 폭은 브라우저에 따라 달라진다. 하지만 size를 지정하여도 브라우저의 설정이나 글꼴 크기에 따라 폭이 달라질 수 있음 보통은 스타일에서 설정을 많이 한다.
maxlength	text, password 컨트롤의 최대 입력 문자수 지정
checked	checked : 체크박스, 라디오버튼의 초기 선택상
disabled	컨트롤을 포커스, 선택, 변경 등의 조작이 불가능하게 하고 데이터는 서버로 전송하지 않음
readonly	컨트롤의 내용을 변경되지 않게 하지만 데이터는 서버로 전송

```
<input type="submit" name="submit1" value="전송하기" >
<input type="password" name="password1" size="10" maxlength="10" >
<input type="radio" name="radio1" value="radio_value1" checked >
```

선택요소:<select>

- 셀렉트메뉴 전체를 정의하는 요소
- 여러 줄로 된 텍스트 필드를 생성하는 인라인 요소이다.
- 셀렉트메뉴는 드롭다운 메뉴와 리스트 박스로 나눌 수 있다.

```
<select name="변수명" id="식별자">
    <option value="초기값">항목</option>
</select>
```

select 요소의 주요 속성

size	표시 줄 수 지정하고 목록을 박스로 표시 (기본값 1) 지정된 size를 넘는 항목의 선택을 위해 스크롤바가 자동으로 제공됨
name	셀렉트메뉴의 이름 지정
option	셀렉트메뉴의 항목을 정의하는 요소 select 요소에는 한 개 이상의 option 요소가 포함되어야 한다.

option 요소의 주요 속성

selected	해당 항목이 선택된 상태 (선택된 option 요소가 없을 경우 첫번째 option 요소가 기본으로 선택됨)
value	선택된 항목의 값

<optgroup>: select에서 그룹을 만들 때 사용한다.
label속성: optgroup의 이름 설정

```
<select name="변수명" id="식별자">
  <optgroup label="그룹이름">
    <option value="초기값">항목</option>
  </optgroup>
</select>
```

<textarea></textarea>

- 여러 줄로 된 텍스트 필드를 생성하는 인라인 요소이다.

```
<textarea id="cmtCont" name="cmtCont" cols="가로크기" rows="세로크기"></textarea>
```

textarea 요소의 주요 속성

name	컨트롤의 이름 (서버에서 처리할 데이터의 컨트롤에는 name 속성 필수)
disabled	컨트롤을 포커스, 선택, 변경 등의 조작이 불가능하게 하고 데이터는 서버로 전송하지 않음
readonly	컨트롤의 내용을 변경되지 않게 하지만 데이터는 서버로 전송
rows	표시 줄 수를 지정(입력 가능한 줄 수를 제한하는 것이 아니라 브라우저에서 보이는 줄 수)
cols	표시폭을 문자수로 지정(한 문자의 폭이 어느 정도일지는 브라우저의 설정이나 글꼴 크기에 영향을 받기 때문에 브라우저마다 차이가 발생할 수 있음)

<button>

- 버튼을 생성하는 인라인 요소
- 기능적으로는 input 요소로 생성하는 버튼과 같다. 요즘에는 폼태그 안에서 사용보다는 폼태그 이외의 버튼이 필요시 많이 사용한다.

```
<button type="button">버튼의 이름</button>
```

button 요소의 주요 속성

type	기본값이 브라우저에 따라 다를수 있기 때문에 type을 기입하는것을 권장 submit : 송신버튼 reset : 리셋버튼 button : 범용버튼
------	--

<datalist> </datalist>

- 서식요소에 입력 추천값을 제시해 주는 요소
- 검색에 많이 사용된다.

```
<input type="text" list="dataList">
<datalist id="dataList">
  <option value="html5"></option>
  <option value="html5 video"></option>
  <option value="html5 강좌"></option>
  <option value="html5 canvas"></option>
  <option value="html5 demo"></option>
</datalist>
```

2. Form2.0 속성

placeholder : 입력내용의 안내

```
<input type="tel" id="placeholder" name="placeholder" placeholder="숫자로만 입력해 주십시오" >
```

autofocus: 자동으로 포커스를 위치시킴

- 페이지가 로딩되면 autofocus 속성이 지정된 서식 요소로 포커스가 이동된다.

```
<input type="text" id="input02" name="input02" autofocus>
```

autocomplete: 자동완성 기능의 컨트롤

```
<input type="text" id="autocompleteOff" name="autocompleteOff" autocomplete="off">
<input type="text" id="autocompleteOff" name="autocompleteOff" autocomplete="on">
```

required: 필수 입력사항

```
<input type="text" id="required" name="required" required>
```

min , max: 수치 입력의 최대 값과 최소 값

```
<input type="number" id="minMax" name="minMax" min="50" max="100" title="50부터 100까지 사이값으로 입력해주세요" />
```

기타 폼과 비슷한 수치를 넣을 수 있는 요소

<progress> </progress>

- 진행률의 표시
- 최소값이 0으로 고정되어 있으므로 min값이 없다.

```
<progress id="progress" name="progress" max="1"*** 진행중</progress>
```

<meter> </meter>

- 정도의 표시
 - <progress>가 동적으로 변화하는 상황을 나타내면, <meter>는 정적인 수치를 나타냅니다.
- <meter id="meter" name="meter" min="0" max="1" value="0.5">50% 사용중</meter> input 요소의 주요 속성

내용을 나타내는 요소

강조:,

- 문단의 내용 중에서 강조하려는 콘텐츠를 마크업할 때는 strong와 em을 사용할 수 있다.
- strong은 객관적으로 누구나 중요하다고 생각되는 내용
- em은 객관적이지 않아도 주관적으로 강조하고자 할 때 사용된다.(emphasis)
- 일부 음성 브라우저의 경우 강조 관련 요소로 마크업한 콘텐츠는 좀 더 크게 읽어 주거나 두 번 반복하여 읽어 주기도 합니다.

```
<p> 텍스트 <em> 세상에서 가장 예쁜 탤런트이다</em> 텍스트 </p>
```

```
<p> 텍스트 <strong> 별금 300만원</strong> 텍스트 </p>
```

인용문 <blockquote> & <q>

- 블록 단위의 인용문이나 문장 형태의 짧은 인용문의 경우 blockquote와 q 요소를 이용할 수 있습니다.
- blockquote 요소는 블록 요소만 포함할 수 있고, q는 인라인 요소와 텍스트를 포함할 수 있습니다.
- 인용문의 경우 출처를 명시하려고 할 때 cite 속성으로 인용한 곳을 지정해야만 합니다.

```
<blockquote cite="출처">
  <p>블록 인용구</p>
</blockquote>
<p>텍스트 <q cite="출처">인라인 인용구</q> 텍스트 </p>
```

<cite></cite>

결과물의 인용 및 참조를 정의할 때 사용.

속성도 가능하고 요소도 가능하다

축약어 및 두문자어:<abbr>,<acronym>

- abbr:축약어(abbreviation)
- 두문자어인 acronym 요소는 여러 개의 단어들로 이루어진 문장 또는 복합 단어의 앞글자만 따서 줄여 사용하


```
<p> <abbr title="World Wide Web Consortium">W3C</abbr></p>
<p> <acronym title="Rich Internet Application">RIA</acronym></p>
```

추가글 및 삭제글:<ins>,

추가글이나 삭제글의 변경된 콘텐츠를 마크업할 때 사용하는 요소로, ins 요소와 del 요소를 사용할 수 있습니다.
~~datetime~~ 속성을 사용하여 정확한 시간과 cite속성으로 출처를 제시해야 한다.

```
<p><ins>취소 콘텐츠</ins><p> 또는 <ins><p> 취소 콘텐츠</p><ins>
<p><del> 추가 콘텐츠 </del><p> 또는 <del><p>추가 콘텐츠</p><del>
```

첨자:<sup>,<sub>

윗첨자나 아랫첨자 형식으로 콘텐츠를 표현해야 할 경우 sup 요소와 sub 요소를 사용할 수 있습니다.

```
<p>X<sup>2</sup> </p> – 윗첨자
<p>H<sub>2</sub>O </p> – 아랫첨자
```

주소:<address>

- address 요소를 이용하여 웹 문서의 아래쪽에 연락처(이메일 주소 포함) 및 제작자, 저작권(copyrights) 정보 등을 표시할 수 있습니다.
- address 안에는 인라인 요소와 텍스트를 포함할 수 있지만, 블록 요소는 포함할 수 없습니다. (호환형 제외)

```
<address> 작성자 정보 콘텐츠 </address>
```

mark

형광펜으로 특정단어들을 칠해서 강조하는 효과와 비슷하며 문장내의 중요한 부분에 강조효과를 넣는 , 과는 다른 개념이다. 시각적 주목효과 만을 노리며 스타일이 정의되어 있지 않으므로 직접 효과를 작성한다.

```
<p>지금 공부하고 있는 것은 <mark>HTML5</mark> 입니다.</p>
```

time

날짜와 시간을 기계가 이해할 수 있게 명확하게 인코딩하면서 사람도 이해할 수 있게 노출시킨다. datetime 속성으로 YYYY-MM-DD(날짜) / HH:MM:SS(24시의 형태)로 표기하고 동시에 나타낼 경우, 2013-01-30T12:00:00+09:00 와 같이 T문자로 구분하고 문자열 끝에 타임존 정보를 더하면 된다.

만약에 추

```
<p>도서관은 <time>22:00</time> 에 문을 닫습니다.</p>
<p>다음주 <time datetime="2013-01-30">수요일</time>은 쉽니다.</p>
<p>Published on <time datetime="2008-02-14 20:00">2008년 2월 14일</time></p>
```

details

HTML 문서에서 추가적인 설명을 붙일 때 사용하며 워드프로세서의 '각주' 역할과 비슷하다. <summary> 태그와 함께 사용되며 기본적으로 <details>는 '감춤' 상태이다. summary 부분만 보이도록 되어 있고 그 외 콘텐츠는 기본적으로 감추어져 있는 것이다. details 콘텐츠를 모두 펼쳐 놓기 위한 속성이 있는데 'open' 속성이다.

```
<details open="open">
  <summary>HTML5 란?</summary>
  <p>HTML5는 어찌고 저찌고</p>
</details>
```

dialog

대화를 의미하는 콘텐츠로 만들고자 할 때 사용하며 <dt>에 말하는 사람 <dd>에 내용을 작성한다.

```
<dialog>
  <dt>나</dt>
  <dd>아!, 이런 오늘 작업한 걸 저장하지 않고 종료했네~</dd>
  <dt>수정</dt>
  <dd>이런, stupid!</dd>
</dialog>
```

글자의 의미

<i>	HTML5에서 의미가 변한 요소로, 단순히 이탤릭체로 나타내기 위한 요소가 아니라 분위기를 전환 하는 의미의 텍스트를 나타내며, 전문 용어, 관용구, 생각 또는 선박 이름 등에 사용.
	텍스트를 볼드체로 표현하여 다른 콘텐츠와 구분 하고자 할 때 사용.
<small>	저작권 정보 등과 같이 작은 크기의 텍스트 콘텐츠에 사용.

<ruby></ruby> 루비 문자

루비 문자는 문장 내의 임의의 문자에 대해 읽는 법을 알려주는 글자

태그	설명
ruby	루비문자 선언태그
rt	위에 위치하는 설명하는 문자
rp	ruby 태그를 지원 할 경우 출력되지 않는 태그

```
<ruby>
  北<rt>běi</rt>
  京<rt>jīng</rt>
</ruby>
```

```
<ruby>HTML N° 5
<rp>(</rp><rt>Web Standard</rt><rp>)</rp>
</ruby>
```



컴퓨터용어

<code>	소스 코드를 정의할 때 사용.
<samp>	시스템의 상태 메시지를 정의할 때 사용.
<kbd>	키보드 입력 값을 정의할 때 사용.
<code>	컴퓨터코드를 정의할 때
<var>	변수값을 정의할 때

기타

<dfn>	정의형 목록의 용어를 의미. <p><dfn>HTML</dfn> 은 standard markup language 이고 웹페이지를 생성한다.</p>
<bdo>	문단 내 일부 텍스트의 방향을 지정할 때 사용 속성의 값으로는 ltr, rtl이 있다.
<wbr>	특정 단어 단위로 줄바꿈을 적용하고자 할 때 사용.

미디어 오브젝트 연결 요소

인라인 프레임 <iframe>

- 인라인 프레임을 이용하여 콘텐츠를 삽입할 경우에는 iframe 요소를 사용합니다.
- 인라인 프레임은 인터넷 익스플로러의 독자적인 요소였으나 HTML 4.0과 HTML 4.01 이후에 정식으로 지원한다.
- 인라인 프레임을 사용할 경우에는 접근성을 고려하여 iframe에 title 속성을 함께 제공하는 것이 바람직합니다.

```
<iframe src="https://www.naver.com">
<p>Your browser does not support iframes.</p></iframe>
```

Attribute	Value
<u>height</u>	pixels
<u>name</u>	text
<u>sandbox</u>	allow-forms allow-pointer-lock allow-popups allow-same-origin allow-scripts allow-top-navigation
<u>src</u>	URL
<u>srcdoc</u>	HTML_code
<u>width</u>	pixels

```
<iframe srcdoc="<p>Hello world!</p>" src="demo_iframe_srcdoc.htm"></iframe>
```

<embed>

Element	CHROM	IE	Firefox	Safari	Opera
<embed>	Yes	Yes	Yes	Yes	Yes

속성	설명
height	크기
src	URL
type	media_type
width	크기

```
<embed src="helloworld.swf">
```

<object></object>요소

Element	CHROM	IE	Firefox	Safari	Opera
<object>	Yes	Yes	Yes	Yes	Yes

속성	값
data	경로
form	form_id
height	크기
name	name
type	media_type
usemap	#mapname
width	크기

```
<object width="400" height="400" data="helloworld.swf"></object>
```

<picture></picture>요소

Element	CHROM	IE	Firefox	Safari	Opera
<picture>	38.0	13.0	38.0	9.1	25.0

```
<picture>
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
  
</picture>
```

〈meta〉

- 웹 문서의 관련된 정의 또는 설정을 담당한다.

name	metadata의 이름을 지정하는데 사용
http-equiv	문자의 인코딩의 설정 또는 몇초 후 페이지를 다시 로드할 것인지 등에 사용
content	name, http-equiv 속성과 연관된 설정 값을 지정한다.
viewport	

인코딩설정

많이 사용하는 코드 셋트

- euc-kr : 한글인코딩
- utf-8 : 다국어인코딩

html5

<meta charset="utf-8">

<meta http-equiv="Content-type" content="text/html+xml; charset=utf-8"/>:/xhtml1.0

다양한 문서 정보 지정하기

<meta name="author" content="" >	제작자
<meta name="description" content="" >	요약설명
<meta name="keywords" content="" >	키워드
<meta name="filename" content="" >	파일이름
<meta name="generator" content="" >	제작도구
<meta name="location" content="" >	위치
<meta name="other Agent" content="" >	웹책임자
<meta name="publisher" content="" >	제작사
<meta name="subject" content="" >	홈페이지 주제
<meta name="title" content="" >	제목

HTML5에서는 reply-to, date, copyright을 적용하면 오류로 표시.

해당내용은 문서의 본문에서 <address> 나<footer>요소로 표현하거나 주석으로 작성 권장. (HTML5문서는 아직 완성된 언어가 아니므로 완성 될때는 오류가 사라 질수도 있음)

검색엔진설정

robots설정: 검색엔진에 문서의 정보를 알수 있도록 안내하는 역할을 한다.

index	해당 페이지 정보를 검색해서 등록하는 것을 허락한다
noindex	해당 페이지 정보를 검색해서 등록하는 것을 허락하지 않는다.
follow	해당 페이지에 있는 링크를 추적하는 것을 허락한다.
nofollow	해당 페이지에 있는 링크를 추적하는 것을 허락하지 않는다.
all	해당 페이지의 정보를 검색해서 등록하고 해당 페이지에 있는 링크를 추적하는 것을 허락한다.
none	해당 페이지의 정보를 검색해서 등록하거나 해당 페이지에 있는 링크를 추적하는 것을 허락하지 않는다.

<meta name="goolebot" content="noindex">	구글로봇 색인차단
<meta name="Robots" content="ALL" />	색인 링크 모두 허용
<meta name="Robots" content="index,follow" />	색인허용, 링크 차단
<meta name="Robots" content="noindex,follow" />	로봇 색인차단, 링크허용
<meta name="Robots" content="index,nofollow" />	색인허용, 링크차단
<meta name="Robots" content="noindex,nofollow" />	색인, 링크 모두 차단

refresh설정

http-equiv="refresh" content="시간;URL=주소"
<meta http-equiv="refresh" content="7;url=http://www.greenart.co.kr/" />viewport

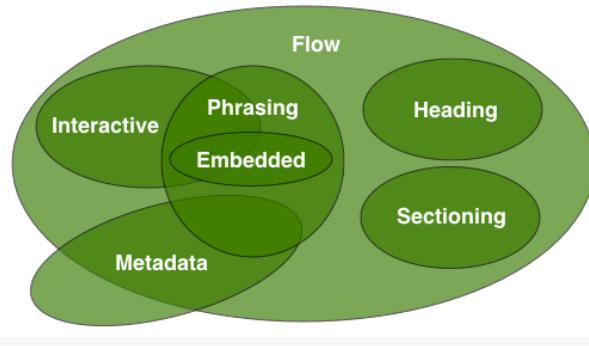
블록요소와 인라인 요소

요소의 분류:

- 의미와 역할
- 성격

콘텐츠로 분리

HTML 문서에 의해 제공되는 콘텐츠는 크게 7개의 카테고리로 분류하고 있으며, 각 요소는 하나 이상의 카테고리에 포함될 수 있습니다(일부 요소는 해당 카테고리가 없다).



카테고리	설명	해당 요소
Metadata content	콘텐츠의 표현이나 동작 또는 다른 문서와의 관계를 설정하기 위한 콘텐츠	base, command, link, meta, noscript, script, title
Flow content	문서의 본문 영역에 나타나는 콘텐츠	a, abbr, address, area, article, aside, audio, b, bdi, bdo, blockquote, br, button, canvas, cite, code, data, datalist, del, details, dfn, dialog, div, dl, em, embed, fieldset, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hr, i, iframe, img, input, ins, kbd, keygen, label, link, main, map, mark, math, menu, meta, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style, sub, sup, svg, table, template, textarea, time, u, ul, var, video, wbr, text
Sectioning content	영역을 정의하기 위한 콘텐츠	article, aside, nav, section
Heading content	제목을 정의하기 위한 콘텐츠	h1, h2, h3, h4, h5, h6
Phrasing content	문단을 이루는 콘텐츠	a, abbr, area, audio, b, bdi, bdo, br, button, canvas, cite, code, data, datalist, del, dfn, em, embed, i, iframe, img, input, ins, kbd, keygen, label, link, map, mark, math, meta, meter, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, template, textarea, time, u, var, video, wbr, text Embedded content 외부 리소스로 삽입되는 콘텐츠 audio, canvas, embed, iframe, img, math, object, svg, video
Interactive content	사용자와의 상호 작용을 위한 콘텐츠	a, audio, button, details, embed, iframe, img, input, keygen, label, object, select, textarea, video

Inline과 block로 분류하기

inline과 block요소의 확인

같은 요소를 두번 이상 입력할 때 세로로 렌더링 되면 블록요소이고, 같은 요소를 두번 이상 입력할 때 가로로 렌더링 되면 인라인요소이다.

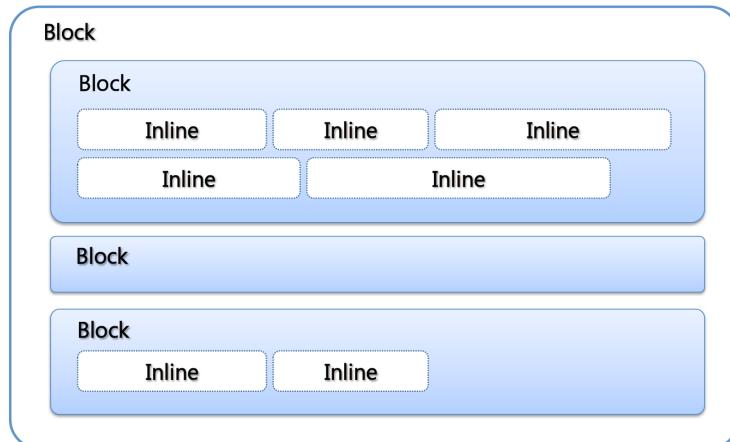


1. 블록 요소 (Block Elements)

- 줄을 바꿔 각각 독립된 줄에 표시
- 자신만의 크기와 여백을 가지고 있다.(내부여백, 외부여백)
- 인라인 요소와 텍스트 혹은 또 다른 블록 요소를 포함할수 있다.
- h1~h6, p, div, ul, ol, li 등

2. 인라인 요소 (Inline Elements)

- 다른 인라인 엘리먼트와 같은 줄에 표시
- 거의 모든 인라인 요소는 크기 개념이 적용이 안되고 외부 여백이 적용 안된다.
- 또 다른 인라인 요소와 텍스트를 포함하지만 블록 요소를 포함할 수는 없다.
- 인라인 요소와 텍스트는 반드시 블록 레벨 요소에 포함시켜 나타내야만 합니다. 즉 BODY 요소의 바로 하위에는 블록 레벨 요소만 자식 요소로 올 수 있고, 인라인 요소를 BODY 요소의 직접적인 하위 내용으로 하는 것은 옳지 않다.
- a, img, strong, span 등



부모 요소와 자식 요소

- 요소는 경우에 따라 하나 이상의 다른 요소를 포함할 수 있다. 부모 요소(parent element)는 다른 요소를 포함하고 있는 요소이며, 자식 요소(child element)는 부모 요소에 포함된 요소를 말합니다.
- 위계적으로 보면 부모 요소는 상위 레벨의 요소, 자식 요소는 하위 레벨의 요소라고 볼 수 있습니다.
- HTML 문서에서 최상위 레벨의 요소를 루트 요소(root element)라고 부르는데 <html>요소가 이에 해당합니다.
- 루트 요소는 자식 요소만 가질 수 있으며, 모든 요소의 부모 요소가 됩니다.
- 포함 관계는 부모-자식 관계
- 병렬 관계는 형제 관계를 이루게 됩니다.

- 부모 요소에 적용된 속성은 자식 요소로 상속되며 자식 요소에 적용된 속성은 부모 요소로 역상속 되지는 않습니다.

