**HTML Guidelines**

# HTML 네이밍 및 기본 규칙

## 파일 네이밍

파일 네이밍은 파일 네이밍 규칙을 따릅니다.

## 들여쓰기

### 기본규칙

* HTML에서 들여쓰기는 다른 언어와 마찬가지로 스페이스 2칸으로 들여쓰기 합니다.
* 부모 요소의 자식 요소는 부모 요소보다 한 단계 더 들여쓰기를 함으로써 문서의 구조를 명확히 알아볼 수 있도록 합니다.
* 요소의 형제 요소는 들여쓰기를 동일하게 처리하여 같은 부모 요소를 가지고 있다는 것을 명확하게 표현합니다.

### 들여쓰기 제외 규칙

* head, body는 들여쓰기를 하지 않습니다.
* body요소의 한 단계 아래 자식요소인 큰 레이아웃을 담당하는 요소들(header, body, aside, footer, layer 등)은 들여쓰기를 하지 않습니다.

## 줄 바꿈

### 기본 규칙

* block타입 tag는 자식의 내용을 적기 전에 줄 바꿈을 하는 것을 원칙으로 합니다.

<div>

 내용

</div>

* inline, inline-block 타입안에 텍스트만 있는 경우 줄 바꿈을 하지 않습니다.

하지만 구조에 따라 줄 바꿈이 필요하다면, 이를 허용합니다.

* li나 tr, dd, dt등 리스트 형식의 경우 내용에 따라 inline과 동일하게 한 줄로 하도록 합니다.

<ul>

<li>아이템</li>

<li>아이템</li>

<li>아이템</li>

</ul>

### 빈 줄 규칙

* body안에서 의미 있는 레이아웃간 구분에는 사이에 빈 줄을 삽입하여 구분합니다.

<body>

<section>

 내용

</section>

 빈줄 삽입

<section>

 내용

</section>

 빈줄 삽입

<aside>

 내용

</aside>

</body>

* 큰 레이아웃의 구분에는 주석을 넣어 어느 레이아웃인지 표기해 줍니다. 자세한 사항은 “주석” 부분을 참고하세요.

<!—- 본문시작 -->

<section>

 내용

</section>

**<!-- //본문 끝 -->**

## tag와 attribute 표기

### 기본 표기 규칙

* tag명과 attribute는 소문자로 표기합니다.
* attribute의 값은 큰 따옴표(“)로 묶습니다.
* 속성 이름만 적지 말고 값도 항상 함께 표기합니다.

disabled=”disabled”

* 표준에 정의되지 않는 사용자 커스텀 속성은 반드시 앞에 “data-”를 붙여줍니다.

<input type=”text” name=”userid” data-text=”아이디” />

* 속성 값 중 특수문자는 entity문자로 변경합니다.

entity문자 예제

& → &amp;

> → &gt;

공백문자 → &nbsp;

* 단순 띄어쓰기 공백은 &nbsp;를 넣지 않습니다.
* 속성에 자바스크립트 이벤트 바인딩 및 스크립트 코드를 삽입하는 것은 지양합니다.

자바스크립트 코드 사용금지 예제.

아래와 같은 코드는 사용하지 않습니다.

<body onload=”……”>

<input type=”button” onclick=”alert(‘test’); return false;” />

<input type=”text” onfocus=”……” />

<form onsubmit=”……”>

<a href=”javascript:……”>click here</a>

## id, class, name attribute의 naming 규칙

### id, class, name 속성의 값은 목적에 맞고 의미에 맞게 정하여야 합니다.

* **id**는 한 요소의 고유한 이름으로 문서 내에 오직 하나만 정의되어야 합니다
* **class**는 어떤 요소의 집합체나 종류를 나타내는 이름으로, 하나 또는 그 이상의 요소들이 같은 클래스를 가질 수 있습니다. 한 요소가 여러 개의 클래스를 가질 수도 있으며 여러 값들을 정의할 때에는 스페이스로 구분합니다.
(출처: <http://www.w3.org/TR/1999/REC-html401-19991224/struct/global.html#h-7.5>.2)
* **name**은 form객체내에 있는 필드를 위해 사용하는 이름이며 form을 서버에 전송할 때 서버가 참조하는 이름입니다.

### 기본 규칙

* id와 class, name의 이름은 소문자로 작성하는 것을 원칙으로 합니다.
* id가 여러 단어로 이루어져야 할 경우, 단어 사이의 구분은 “-”으로 합니다.
* class는 되도록 한 단어로 하되, 필요하면 두 단어 이상으로 표현하려면 역시 “-”으로 구분합니다.

id, class네이밍 예제

<p id=”important-hint” class=”red hint lower-case”>…</p>

<nav id=”top-nav” class=”drop-down”>
 <ul id=”menu-list”>
 …

# 2. 문서 구조

## DTD(Document Type Definition) 선언

* 앞으로 진행하는 신규 프로젝트는 HTML5 형식인 <!DOCTYPE html> 을 따르는 것으로 합니다. 다만, 진행하는 프로젝트의 성격에 따라 HTML4.01이나 XHTML1.1도 사용합니다. 이때에는 호환모드를 사용합니다.

HTML5

<!DOCTYPE html>

HTML 4.01 Transitional(HTML 4.01 호환모드)

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

XHTML 1.0 Transitional(XHTML 1.0 호환모드)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

* 반드시 문서의 첫 줄에 DTD만 표기되도록 합니다. dtd를 선언하기 전 다른 문자가 오면 안됩니다.

## <html>

html에 올 수 있는 attribute는 필요에 따라 lang을 일반적으로 사용합니다.

<html lang=”ko”>

* 기타 attribute는 필요에 따라 추가하여 사용합니다.

## <head>

### <meta>

* 필요한 항목을 적절히 사용합니다.
* meta중 문서의 유니코드선언인 utf-8선언은 반드시 해줍니다.

<meta charset="utf-8" />

* SEO(Search Engine Optimization)을 위하여 사용되는 meta정보는 아래의 항목을 참고하여 정확히 작성합니다.

<meta http-equiv=”description” content=”문서의 설명이 들어갑니다.” />

<meta http-equiv=”keywords” content=”문서에 등장하는 핵심적인 단어들이 들어갑니다.” />

<meta http-equiv=”author” content=”문서를 작성한 사람의 정보가 들어갑니다.” />

* IE8부터 지원하는 하위 브라우저 호환모드를 사용할 수 있습니다. 이를 위한 meta태그 사용은 “IE버전별 분기법 및 버전별 상황 대처법”항목을 참고하세요.

### <title>

* title은 문서의 의미에 맞는 제목을 사용하는 것을 원칙으로 합니다.
* title에서 depth를 표시하는 경우 그 페이지의 정보를 맨 앞으로 가져오고, 홈페이지의 타이틀을 뒤로 오게 합니다.

<title>사용자페이지 -RemoteCall 5X</title>

### <link>

html문서에 CSS나 favicon등 필요한 외부 파일들을 불러올 때 사용합니다. 반드시 <head>안에서 정의합니다.

* CSS

CSS를 불러 올 때 media를 정의하여 어느 곳에 적용될 스타일시트인지 정의합니다. media query를 정의하기 위해서라도 반드시 정의합니다.

<link rel="stylesheet" media="screen" type="text/css" href="style.css" />

* Favicon

파비콘은 크로스 브라우징을 고려하여 아래와 같은 방법을 사용합니다.

<link rel="icon" type="image/png" href="favicon.png" />

<!--[if ie]>

<link rel="shortcut icon" type="image/x-icon" href="favicon.ico" />

<![endif]-->

* IE에서는 ico파비콘만 사용 가능하며 rel="icon"은 적용되지 않으므로 위의 코드를 사용합니다. 리퀘스트를 줄이기 위해 비IE에서는 최대한 용량을 줄이고 크기도 조절가능한 png를 사용을 권장합니다. (ico, png두개의 이미지 준비)

### <style>

style요소는 되도록 link엘리먼트를 이용하여 css파일을 외부로 따로 빼는 방법을 권장하나 여의치 않는 경우라면 반드시 header안에 선언하여야 합니다.

<style type=”text/css”>

 html{ font-size: 16px; font-family: sans-serif; }

 p{ color:#333; line-height: 1.6;}

</style>

### <script>

script요소는 body의 가장 하단에 넣는 것이 좋으나, 문서가 onload또는 onready되기 전에 script가 미리 선언되어야 레이아웃이나 성능에 문제가 없는 경우같이 header에 사용해야 하는 경우라면 아래의 방법대로 사용합니다.

<head>

//IE9 이하에서 html5태그를 지원하기 위해 html5.js가 필요한 경우

<!--[if lt ie 9]>

<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

//body안에 있는 jquery를 이용하여 작성된 스크립트들을 해석해야 할 경우

 <script type="text/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script>

</head>

### HTML5 지원

* IE8버전 이하는 HTML5를 지원하지 않으므로 반드시 <head>안에 아래의 코드를 붙여 넣어 HTML5의 요소들을 지원할 수 있도록 하여야 합니다.

<!--[if lt IE 9]>

<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

* 위의 경로에 있는 주소를 그대로 써도 되지만 되도록 js파일을 프로젝트에 올려서 사용하는 것이 좋습니다.
* frameset이나 iframe안에 들어갈 문서라면 부모파일에 DTD가 선언이 되지 않을 경우, 호환성문제로 인해 깨져보이는 경우가 있습니다. 그럴땐 <!—[if IE]>를 사용하십시오.
* html5shiv 프로젝트 홈페이지 및 최신 빌드 받는 곳: <https://github.com/aFarkas/html5shiv>

## <body>

### 기본 레이아웃 구조 (html5)

html5에 새로 추가된 레이아웃 BOX 요소들을 사용하여 의미에 맞게 레이아웃을 구성합니다.

* **<header>**
문서의 헤더부분으로 문서의 소개와 네비게이션, 목차등이 들어갑니다.
* **<nav>**문서 내 네비게이션요소를 나타낼 때 사용합니다.
* **<hgroup>**문서의 제목(H1~H6)을 그룹화 할 때 사용합니다.

<hgroup>
  <h1>Title goes here</h1>
  <h2>Subtitle of article</h2>
</hgroup>

* **<section>**
범용적인 섹션을 나타냅니다. 일반적인 문서의 애플리케이션이나 특정 영역을 표시할 때 사용합니다. h1~h6등의 제목과 함께 사용할 수 있습니다. 의미가 있는 섹션에 사용하고 의미가 없는 단순 범위지정은 <div>를 사용합니다.
* **<article>**
문서내의 독립적인 글을 표시할 때 사용합니다. 블로그 글이나 뉴스기사를 표시하는 것이 그 예입니다.
* **<aside>**
문서의 주요부분을 표시하고 남은 양 옆의 공간을 표시할 때 사용합니다. 문서의 본문내용과 분리되어도 문제없는 섹션에 사용합니다.
* **<footer>**문서 하단의 꼬릿말 부분이나 문서의 저자, 저작권 정보 등을 나타낼 때 사용합니다.

****

html5 문서구조의 예

html5 레이아웃 예제

<!DOCTYPE html>

<html lang="ko">

<head>

 <meta charset="utf-8" />

 <title>html5 레이아웃 예제</title>

<!--[if lt ie 9]>

<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

</head>

<body>

<header>

 <h1>document header</h1>

 <nav id="top-nav">

 <ul>

 <li><a href="#">menu1</a></li>

 <li><a href="#">menu2</a></li>

 </ul>

 </nav>

</header>

<section id="contents">

<h2>contents header</h2>

 <article>

 <h3>article header1</h3>

 <p>article</p>

 </article>

 <article>

 <h3>article header2</h3>

 <p>article</p>

 </article>

 <article>

 <h3>article header3</h3>

 <p>article</p>

 </article>

</section>

<aside>

 <nav id="side-nav">

 <ul>

 <li><a href="#">menu1</a></li>

 <li><a href="#">menu2</a></li>

 </ul>

 </nav>

</aside>

<footer>

 footer

 <address>

 address

 </address>

</footer>

</body>

</html>

### block요소와 inline요소

레이아웃을 구성하는 block요소는 div가 대표적이고 inline은 span이 대표적입니다. inline요소안에 block요소가 들어가지 않도록 합니다.

Inline-block요소는 inline요소와 속성이 같지만 margin과 padding을 설정 할 수 있습니다.

* **block요소 tag**div, h1~h6, p, ul, ol, dl, blockquote, form ….
* **inline요소 tag**
span, a, strong, em, img, br, input, abbr ….

### table 엘리먼트

table은 표를 작성하는데 사용합니다. 레이아웃을 구성하는 용도로 사용하는 것을 금지합니다. 게시판의 리스트에는 table을 사용합니다.

* **table**
테이블을 정의합니다
* **thead**
테이블의 헤더부분을 묶어서 정의합니다.
* **tbody**
테이블의 본문부분을 묶어서 정의합니다.
* **tfoot**
테이블의 푸터부분을 묶어서 정의합니다.
* **caption**
테이블의 제목을 나타내는데 쓰입니다.
* **colgroup**
한 개 혹은 여러 개의 열을 묶어 테이블의 포멧을 명시할 때 쓰입니다.
* **col**
각각 열의 속성을 명시합니다. colgroup으로 묶을 수 있습니다.
* **tr**
테이블의 행을 정의합니다.
* **th**
테이블의 제목셀을 정의합니다.
* **td**
테이블의 셀을 정의합니다.

### 리스트 엘리먼트

* Ul
* Ol
* Dl

### form 엘리먼트

* form
* input
* select
* textarea
* button

### 기타 엘리먼트

## script태그

### script의 사용

script는 문서의 필요에 따라 head부분 혹은 body의 맨 하단에 사용하는 것을 원칙으로 합니다.

* **head부분에 올 script** : DOM이 정의되기 전에 필요한 코드
* **body하단에 올 script**: DOM이 모두 정의 된 후 작동할 코드
* 필요에 따라 중간에 정의하는 경우가 최대한 없도록 합니다.
* 상단에 위치한 자바스크립트더라도 DOM이 준비된 후 작동하게 하려면 document.ready혹은 window.onload를 사용하세요.

### noscript의 사용

스크립트를 지원하지 않는 브라우저를 위해 noscript를 정의합니다. 반드시 해줄 필요는 없으나 자바스크립트 안내문을 보여줄 필요가 있을 때나 기타 필요한 경우 작성합니다.

//script를 해석하지 못할 시 안내문

<noscript>

 이 페이지는 자바스크립트가 필요합니다.

</noscript>

//script를 지원하지 않을시 다른페이지로 이동시키는 예제

<head>

<noscript>

 <meta http-equiv="refresh" content="0; url=noscript.html">

</noscript>

</head>

## 기타 참고사항

### 스타일의 사용

스타일 시트는 항상 css파일을 분리하여 사용하는 것을 원칙으로 합니다.

태그에 style속성으로 넣어야 할 경우 핵을 쓰지 않고 최소한으로 마크업 하여야 합니다.

### 주석

주석은 레이아웃 구조를 설명할 때나 기타 필요할 때 작성합니다. 항상 같은 줄에 유효한 텍스트나 태그가 오지 않게 작성합니다. 또한 주석을 한꺼번에 연속해서 작성하지 않게 합니다.

잘못된 사용 예

태그나 글자와 같은 줄에 주석 넣기

<a href="#">링크</a><!-- 링크 -->

본문 글 <!-- 본문글 -->

주석 뒤에 바로 다른 주석넣기

<!-- 주석 -->

<!-- 다른 주석 -->

* 레이아웃의 시작과 끝 알림

레이아웃을 구성하는 요소의 앞과 뒤에 다음과 같이 주석을 달아 어느 부분인지 표시합니다.

<!-- contents -->

<section id="contents">

 ...

</section>

<!-- //contents -->

* 테스트 코드 숨김 처리

주석을 코드를 숨기는데 사용한다면 아래와 같이 작성합니다.

<!-- contents 숨김

<section id="contents">

 ...

</section>

// -->

* 개발에 필요한 주석

개발에 필요한 주석을 넣되 위의 규칙에 따라서 넣습니다.

* IE6에서는 float되는 요소 근처에 주석이 있으면 버그가 발생하니 주의해서 사용합니다.

### IE버전별 분기법 및 버전별 상황 대처법

IE에서 브라우저간 호환을 맞추기 위해 조건부 주석을 사용하거나 meta태그의 "X-UA-Compatible"을 이용하여 IE브라우저에서 지원하는 하위버전호환 모드를 사용할 수 있습니다.

* 조건부 주석 사용

IE에서는 주석을 이용하여 조건식에 따라 버전별로 마크업을 달리할 수 있습니다.

* 조건부 주석의 조건식

|  |  |  |
| --- | --- | --- |
| **조건** | **예** | **설명** |
| **IE** | [If IE] | 모든 버전의 IE |
| **IE Version** | [If IE 7] | 지정한 버전의 IE |
| **true/false** | [If true] | 항상 참/거짓 |
| **!** | [If !IE] | IE가 아닐 때 |
| **lt** | [If lt IE 9] | IE 지정버전의 하위 버전들 |
| **lte** | [If lte IE 9] | IE 지정버전을 포함한 이전 버전들 |
| **gt** | [If gt IE7] | IE 지정버전의 상위 버전들 |
| **gte** | [If gte IE7] | IE 지정버전을 포함한 상위 버전들 |
| **()** | [If ! (IE7)] | 조건식에서 ()역할과 같음 |
| **&** | [If (gt IE 6) & (lt IE 8)] | 조건식에서 AND연산자와 같음 |
| **|** | [If (gt IE 6) | (lt IE 8)] | 조건식에서 OR연산자와 같음 |

* 조건부 주석 사용법 예제

//IE9 이전 버전에서 html5태그를 지원하기 위해 조건부 주석을 사용하는 예

<!--[if lt IE 9]>

<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

* 조건부 주석 사용시 주의사항
* IE버전별로 body안의 마크업을 다르게 나타나기 위해 사용하는 것을 지양합니다.
* CSS를 수정함에 있어서 수정해야 할 부분이 많지 않을 경우 되도록 핵을 사용하고 IE버전별로 CSS파일을 따로 가져오는 방법은 될 수 있는 한 사용하지 않습니다.
* Meta 태그의 "X-UA-Compatible"사용

IE8부터 지원하는 기능으로 하위 IE버전 호환모드가 제공 되며, 각 버전별로 IE7까지의 하위 버전을 호환하여 지원하는 기능입니다. 문서의 호환 모드는 아래와 같이 간단히 meta태그의 속성으로 설정할 수 있습니다.

* Meta 태그 "X-UA-Compatible"사용 예제 (문서모드를 IE7로 설정)

<meta http-equiv="X-UA-Compatible" content="IE=7" />

### 브라우저별 tag지원 상황 및 대처방법

IE8이하의 브라우저에서는 공식적으로 HTML5를 지원하지 않습니다.

# 3. 기타 사항

## 의미있는 태그 사용법

Webpage에 표시되는 정보는 의미에 맞는 형식으로 마크업 되어야 합니다.

시멘틱(Semantic)한 문서를 위해서 각 영역에 적절한 tag를 넣어서 문서를 마크업합니다.

HTML5에서는 문서의 영역에 대해 의미를 부여하는 태그들이 많이 추가 되었습니다. 자세한 사항은 [기본 레이아웃 구조(html5)](#_기본_레이아웃_구조) 항목을 참고하세요.

## 웹페이지 최적화 방법

* 포토샵에서 이미지 저장시 save for web을 사용하여 저장.
* jpg이미지는 높은 품질 대비 용량을 위해 quality를 100보다 조금 낮게 설정
* 이미지는 되도록 png를 사용
(IE6에서 투명 배경이 있는 이미지를 사용해야 할 경우 png, gif 두개의 이미지를 준비하여 css핵이나 조건부 주석으로 처리)
* png를 최대한 압축하여 사용. 용량이 절반이하로 줄어듬. (png압축툴 pngOptimizer추천)
* 파비콘도 png,ico 두가지를 준비 (ico변환툴 : iconBox추천)
* 조그마한 이미지들은 하나의 이미지 파일로 모아서 사용
* 이미지를 만들 때 사용하는 색상을 되도록 적게 하면 용량측면에서 많은 이득을 볼 수 있음
* Js,css는 가능하면 빌드시 압축하여 배포(이클립스에서 js,css자동 minify할 수 있도록 툴 제공 예정)
* 웹페이지는 가능하면 gzip압축을 사용하여 압축전송(서버측 설정 필요함)