**FUZ Coding Convention**

**for Markup Languages (HTML/CSS) Version 0.5**

**Prepared by**

**개정이력**

|  |  |  |
| --- | --- | --- |
| **버전** | **내역** | **일자** |
| 0.5 | 재정 | 2017.02.21 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**1. 개발환경설정**

**1.1.1. 해상도**

프로젝트 진행 시 기획파트에서 지정.

**1.1.2. 크로스브라우징**

IE 최신버전을 기준으로 프로젝트 특성에 따라 IE 하위버전 지원 시, IE8이하는 지원하지 않으며  
IE 하위버전은 레이아웃이 크게 깨지지 않는 선까지 수용한다.  
그 외에 Chrome, Firefox, Safari를 기준으로 한다.  
HTML 표준 유효성 검사는 http://validator.w3.org 또는 Chrome, Firefox 확장프로그램 Validator를 사용한다.

**1.1.3. 웹접근성(WA)**

한국형 웹 콘텐츠 접근성 지침 2.1을 기준으로 범위는 WA가이드 라인 참고하며,  
웹접근성 테스트는 Chrome, Firefox 확장 프로그램 Open-WAX, N-WAX를 사용한다.

**1.1.4. HTML 버전 관리**

보통은 SVN을 사용하나, 특별한 경우 개발파트에서 지정한 S/W를 사용한다.

**2. 디렉토리 구조**

**A. 파일 및 폴더 네이밍**

• 표 2-8 파일 및 폴더 네이밍 규칙을 따른다.

**<파일 및 폴더 네이밍 규칙 예>**

|  |  |  |
| --- | --- | --- |
| **분류** | **예제** | **설명** |
| HTML | fuz0001001.html | '페이지ID.html' 로 사용 |
| CSS | news.css | '서비스영문이름.css'로 사용한다 |
| m.news.css | 모바일일 경우 m.으로 사용한다. |
| Folder | 프로젝트이름 | 신규 프로젝트 작업 시 사용 |
| images, css, js | image, css, javascript 폴더 사용 |
| 프로젝트명\html\메뉴명 | HTML 파일의 폴더 분류가 필요한 경우 사용 |

**B. 파일 및 폴더 네이밍 조합**

• HTML파일은 '페이지ID \_의미\_상태' 순서로 조합한다.  
• CSS 파일은 '서비스이름'을 맨 앞으로 하여 조합한다.

**<HTML/CSS 조합 예>**

|  |  |  |
| --- | --- | --- |
| **분류** | **조합 예** | **설명** |
| HTML | fuz0001001\_board\_view.html | '페이지ID\_의미\_상태' 순서로 조합한다. |
| CSS | news\_home.css news\_admin.css news\_pop.css, news\_nanum.css | '서비스이름'을 맨 앞으로 하여 조합한다 |

**3. 네이밍 규칙**

**3.1 기본 규칙**

**A. 일반 규칙**

이름은 영문 소문자, 숫자, 언더스코어( \_ )로 작성한다.

**B. 시작 이름**

이름은 영문 소문자로만 시작할 수 있다.  
단, css 주석문은 영문 대문자를 허용한다. (예: /\* FUZ WSD1Team KDW 120101 \*/)

**<공통 네이밍 규칙 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| Btn | btn |
| 2btn | btn2 |
| \_btn | btn |
| btn\_ | btn |

**C. 약속어**

• 약속어는 레이아웃 약속어, 객체 약속어, 이미지 약속어에 근거하여 작성한다.( 표 2-3 레이아웃 약속어 범위,표 2-4 팝업 레이아웃 약속어 범위,표 B 2-1 객체 약속어,표 B 2-2 이미지 약속어에 근거하여 작성한다.)  
• 약속어가 없으면, 종류와 특성을 나타내도록 네이밍하거나 공통 네이밍 약속어 Guide를 참고하여 작성한다.   
• 공통요소 약속어(.u\_)는 공통업무 담당자에 한에서만 사용 한다.

**D. 영문 소문자, 숫자, 언더스코어 조합**

• 영문 소문자, 숫자, 언더스코어(\_)만 사용할 수 있다.  
• 언더스코어(\_)는 단어와 단어 조합할 때만 사용한다.  
• 언더스코어(\_)가 포함된 약속어는 숫자, 영문 소문자와 조합하여 사용할 수 있다.

**<네이밍 조합 예>**

|  |  |  |
| --- | --- | --- |
| **기본형** | **잘못된 예** | **올바른 예** |
| section | sectionList | section\_list |
| no | no\_1, no-1 | no1, no2, no3 …. no10 |

**3.2. CSS 네이밍**

**3.2.1 id 및 class 네이밍 규칙**

**<HTML의 기본 템플릿 구조>**

#header: 머리글

#wrap: 감싸기

그림 2-1 PC HTML 기본 템플릿 구조

.gnb: Global Navigation Bar

2. 네이밍 규칙

.content: 본문표시

.aside: 곁가지

.ad: 광고

.section: 그루핑

.path: 문서경로 표시

.pgtit: 페이지 타이틀

.section: 그루핑

#container : 컨테이너

.snb: Side Navigation Bar

.nav: Navigation

#visual : 페이지 비쥬얼

.lnb : Local Navigation Bar

.section: 그루핑

#footer: 바닥글

**A. id, class**

• id는 문서 전체의 고유 식별자 이므로 한 문서에서 동일한 id를 여러 번 사용하지 않는다.  
• 레이아웃을 제외한 id는 스타일을 지정하지 않는다.  
• class는 문서에서 여러 번 사용할 수 있다.

**B. 레이아웃 약속어**

레이아웃에는 다음 표에 예약된 id만 사용한다.

**<레이아웃 약속어 범위>**

|  |  |
| --- | --- |
| **약속어** | **범위** |
| #wrap | 페이지 전체 영역 |
| #header | GNB 영역 |
| #visual | 비쥬얼 영역 |
| #container | 본문 영역 |
| #content | 콘텐츠 영역 |
| #footer | 푸터 영역 |

**C. 윈도우 팝업 레이아웃 약속어**

• 팝업 문서의 레이아웃 지정 범위는 동일하다. • 레이아웃 약속어 앞에 'pop\_'를 조합하여 사용한다.

|  |  |
| --- | --- |
| **약속어** | **범위** |
| #pop\_wrap | 팝업 페이지 전체 영역 |
| #pop\_container | 팝업 본문 영역 |
| #pop\_content | 팝업 콘텐츠 영역 |
| #pop\_footer | 팝업 푸터 영역 |

**D. 레이어 팝업 레이아웃 약속어**

• 레이어 팝업 문서의 레이아웃 지정 범위는 동일하다. • 레이아웃 약속어 앞에 'layer\_'를 조합하여 사용한다.

|  |  |
| --- | --- |
| **약속어** | **범위** |
| #layer\_wrap | 팝업 페이지 전체 영역 |
| #layer\_container | 팝업 본문 영역 |
| #layer\_content | 팝업 콘텐츠 영역 |
| #layer\_footer | 팝업 푸터 영역 |

**E. CSS 약속어**

|  |  |  |
| --- | --- | --- |
| **분류** | **약속어** | **부가설명** |
| 네비게이션 | gnb | 전체 메뉴 |
|  | lnb | 2Depth 메뉴 |
|  | snb | 3Depth 메뉴 |
|  | top\_menu | 탑 유틸메뉴 |
| 영역 | section, area | 영역구분 |
|  | inner | 부모 영역이 존재하며 자식 영역이 필요한 경우 |
| 타이틀 | pgtit | 페이지 타이틀 |
|  | tit | 일반 타이틀 |
|  | sub\_tit | 서브 타이틀 |
| 탭 | tab |  |
| 테이블 | tbl |  |
| 리스트 | list | 일반 목록(ul, ol, 리스트 형식의 dl) |
| 버튼 | btn |  |
| 아이콘 | ico |  |
| **분류** | **약속어** | **부가설명** |
| 배경 | bg |  |
| 폼 | input | input box |
|  | select | select box |
|  | radio | radio button |
|  | check | check button |
|  | date | <input type=”date”> |
| 박스 | box |  |
| 선 | line |  |
| 썸네일 이미지 | thum |  |
| 배너 | banner |  |
| 상세내용/부가설명 | decs |  |
| 페이징 | paging |  |
| 번호 | num |  |
| 댓글 | comment |  |
| 화살표 | arrow |  |
| 카테고리 | cate |  |
| 상태 | on / off |  |
| 위치 | top / bttm  left / right |  |
| 이전 / 다음 | prev / next |  |
| 임시사용 | .@ | 임시 사용시 앞에 @을 부친다. |

[조합 기호: "\_"]

**4. HTML 코드 작성 규칙**

**4.1 기본 규칙**

**A. W3C Validation**

HTML은 해당 DTD의 명세에 맞게 작성하며, W3C validation을 통과해야 한다. 단, HTML5 DTD 선언 시 다음 오류 내용은 허용한다.  
• <iframe>의 frameborder, marginwidth, marginheight, scrolling 애트리뷰트

**B. 영문 소문자 사용**

DTD를 제외한 모든 요소와 애트리뷰트는 소문자로 작성한다.

**<소문자 작성 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <SPAN Class="desc">간단한 설명</SPAN> | <span class="desc">간단한 설명</span> |

**C. 애트리뷰트값 표기**

애트리뷰트값은 큰따옴표(" ")로 묶는다.

**<애트리뷰트값 표기 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <img src='test.gif' width='100' height='100' alt=' 테스트'> | <img src="test.gif" width="100" height="100" alt="테스트"> |

**D. Character entity references 사용**

특수 기호는 문자 엔티티 참조를 사용하여 코드로 변환한다. 자세한 사항은 아래 경로에서 확인할 수 있다. 단, 아포스트로피(‘)는 &#39;로 선언한다.  
HTML 4.01의 Character entity references: <http://www.w3.org/TR/html4/sgml/entities.html>  
HTML 5의 Character references: http://www.w3.org/TR/html5/syntax.html#character-references

**<Character entity references 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <a href="...&nid=2">Q&A | <a href="... &nid=2">Q&amp;A |

**E. 스크립트 선언 지양**

Javascript를 사용하되 태그에 inline방식으로 사용하는 이벤트 속성 및 스크립트는 HTML 산출물에 선언하지 않는다.(<head>태그 사이 또는 <body>태그 최하단에 선언할 수 있다)

**<스크립트 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <input type="text" id="user\_id" name="user\_id" onfocus="this.className='focus'"> | <!-- [D] 입력 박스에 포커싱되었을 때, class="focus" 추가 --> <input type="text" id="user\_id" name="user\_id" class="focus"> |
|  | <script> document.getElementById("user\_id").addEventListener('focus',function(e){e.target.className="focus ";},false); </script> |

**4.2 들여쓰기 규칙**

들여쓰기를 하면 코드의 가독성이 높아지고 전체 HTML 구조를 쉽게 파악할 수 있다. 다음과 같은 들여쓰기 규칙을 준수한다.  
마크업의 중첩이 깊어질 때마다 자식 요소는 1탭 들여 쓰고, 1탭의 크기는 공백 4칸으로 설정한다. 문서 내에서 반드시 탭을 이용하여 들여쓰기를 하며, 탭을 대신하여 공백으로 띄어 들여쓰지 않는다. 또한 다음의 경우에는 중첩 단계의 파악이 용이하도록 들여쓰지 않는다.   
• <html>의 자식 요소인 <head>, <body>  
• <head>의 자식 요소  
• <body>의 자식 요소  
• <table>의 자식 요소인 <caption>, <colgroup>, <col>, <thead>, <tbody>, <tfoot>, <tr>,<th>, <td> • <ul>, <ol>, <dl>의 자식 요소인 <li>, <dt>, <dd> 요소  
• <fieldset>의 자식 요소인 <legend> • <map>의 자식 요소인 <area>

**<들여쓰기 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <table>  <caption>...</caption>  <colgroup>  <col>…</col>  </colgroup>  <thead>  <tr>  <th>...</th>  <th>...</th>  </tr>  </thead>  <tbody>  <tr>  <td>...</td>  <td>...</td>  </tr>  </tbody>  </table> | <table>  <caption>...</caption>  <colgroup>  <col>…</col>  </colgroup>  <thead>  <tr>  <th>...</th>  <th>...</th>  </tr>  </thead>  <tbody>  <tr>  <td>...</td>  <td>...</td>  </tr>  </tbody>  </table> |
| <ul>  <li>...</li>  <li>...</li>  </ul> | <ul>  <li>...</li>  <li>...</li>  </ul> |
| <ol>  <li>...</li>  <li>...</li>  </ol> | <ol>  <li>...</li>  <li>...</li>  </ol> |
| <ul>  <li>...</li>  <li>...</li>  </ul> | <ul>  <li>...</li>  <li>...</li>  </ul> |
| <div>  <h3>…</h3>  <p>…</p>  <div>  <span>...</span>  </div>  </div> | <div>  <h3>…</h3>  <p>…</p>  <div>  <span>...</span>  </div>  </div> |

**4.3 빈 줄**

빈 줄을 사용하려면 다음과 같은 사용 규칙을 준수한다. 의미 있는 객체를 구분하기 위하여 코드 그룹 간 1줄씩 빈 줄을 만드는 것은 허용한다. 빈 줄의 간격은 1줄을 초과하지 않는다.

**<빈 줄 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <head>  내용...  </head>  빈 줄  빈 줄  <body>  내용...  </body> | <head>  내용...  </head>  빈 줄  <body>  내용...  </body> |

**4.4 DTD 및 인코딩   
4.4.1 DTD 선언**

**A. 기본 DTD**

신규 HTML 문서를 작성할 때 'HTML5'를 사용한다. 작성 예는 다음과 같다.

|  |
| --- |
| <!DOCTYPE html> |

**참고**  
HTML5는 SGML에 기본으로 하지 않기 때문에, DTD의 참조를 요하지 않는다.  
<!DOCTYPE>은 웹 브라우저가 HTML이 무슨 버전인지 알 수 있게 해준다.

**B. 기타 DTD**

사용 중인 서비스를 부분 개편하거나 완전히 개편하더라도 기존의 HTML/CSS 데이터 의존도가 높고 개발에서 특별한 요청이 있을 경우, 기존과 동일한 DTD를 사용할 수 있다.  
작성 예는 다음과 같다.

HTML 4.01 Transitional

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

XHTML 1.0 Transitional

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

Quirks Mode

|  |
| --- |
| <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> |

**주의**

**아래와 같은 경우 DTD 설정별 표준 문법으로 마크업 하더라도 브라우저에서 Quirks Mode로 인식하여 바르게 해석되지 않으므로 주의한다.**

**4.4.2 인코딩 선언**

HTML 문서는 반드시 인코딩 정보를 선언한다. 인코딩 설정은 DB의 인코딩 방식과도 관련이 있으므로 반드시 담당 개발자와 협의하여 정해야 한다.

**A. 기본 인코딩**

신규 HTML 문서를 작성할 때 기본 인코딩은 utf-8을 원칙으로 한다.  
다음은 인코딩 방식으로 utf-8을 선언한 예이다.

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

HTML4.01 DTD 사용 시에는 아래와 같이 선언한다.

|  |
| --- |
| <meta http-equiv="Content-Type" content="text/html;charset=utf-8"> |

**참고**

**utf-8은 다국어 지원이 가능하며, euc-kr보다 표현 가능한 한글(고어, 음절 등)이 더 많다.  
한글은 utf-8에서 3바이트, euc-kr에서 2바이트를 차지하기 때문에, utf-8이 euc-kr에 비하여 DB 저장 용량이나 트래픽이 많을 수 있다.  
Internet Explorer 7 이상의 버전에서 링크의 밑줄이 글자와 붙어 보이는 현상이 있다.**

**B. 기타 인코딩**

utf-8 인코딩을 사용할 수 없으면 euc-kr을 사용한다. 다음은 인코딩 방식으로 euc-kr을 선언한 예이다.

|  |
| --- |
| <meta http-equiv="Content-Type" content="text/html;charset=euc-kr"> |

HTML, CSS 파일을 저장할 때 반드시 설정한 인코딩을 선택하여 저장한다.

**4.5 주석**

**A. 기본 형식**

HTML 주석의 시작과 종료는 아래와 같이 표기하며, 기본 형식에 맞게 작성한다.

|  |
| --- |
| 시작 주석 <!-- 주석 내용 -->  종료 주석 <!-- //주석 내용 --> |

• 주석 기호와 주석 내용 사이에는 반드시 공백 한 칸이 있어야 한다.   
• 시작과 종료 주석의 주석 내용은 동일해야 한다.

**<주석 기본 형식 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| <!--GNB--> | <!-- GNB --> … <!--// GNB --> |
| <!--\* GNB \*--> |
| <!-- GNB 시작-->  …  <!--// GNB 끝 --> |

**B. 레이아웃 및 콘텐츠 영역의 주석 표기**

wrap을 제외한 레이아웃과 독립된 콘텐츠 영역의 시작과 끝에 주석을 표기하며, 독립된 콘텐츠 영역의 주석 표기는 선택 사항이다.  
HTML 코드와 주석은 줄로 분리해야 한다.

|  |
| --- |
| <!-- content -->  <div id="content">  <!-- 네임카드 -->  <div class="namecard"> ... </div>  <!--// 네임카드 -->  </div>  <!--// content --> |

**4.6 초기 파일**

신규 HTML 문서를 작성할 때 아래 파일을 참고한다.

|  |
| --- |
| <!DOCTYPE html> <html lang="ko"> <head> <meta charset="utf-8"> <title>메뉴 : 서비스명</title> <link rel="stylesheet" type="text/css" href="sevice\_name.css"> </head> <body> <div id="wrap">  <!-- skip 네비게이션-->  <div id="skipnavi">본문바로가기</div>  <!--// skip -->  <!-- header -->  <div id="header">  </div>  <!--// header -->  <!-- container -->  <div id="container">  <!-- content -->  <div id="content">  </div>  <!--// content -->  </div>  <!-- //container -->  <!-- footer -->  <div id="footer">  </div>  <!-- //footer --> </div> </body> </html> |

모바일 신규 HTML 문서를 작성할 때 아래 파일을 기본으로 한다.

|  |
| --- |
| <!doctype html> <html lang="ko"> <head> <meta charset="utf-8"> <meta name="viewport" content=".."> <title>상단: 헤더명</title> <link rel="stylesheet" type="text/css" href="css/.."> </head> <body> <!-- header --> <div id=”header”> </div> <!--// header --> <!-- content -->  <div id="content" role="main"> </div> <!--// content --> <!-- footer --> <div id="footer"> </div> <!--// footer --> </body> </html> |

**참고**

**모바일일 경우 html5 시맨틱 태그를 사용해도 된다.**

**5. CSS 코드 작성 규칙**

**5.1 기본 규칙**

**A. W3C Validation**

CSS는 사용 가능한 Hack과 CSS3 속성을 제외하고 W3C Validation을 통과해야 한다.

**B. 영문 소문자 사용**

모든 속성은 영문 소문자로만 작성한다.  
**<소문자 작성 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class{Font-Family:AppleGothic} | .class{font-family:AppleGothic} |

**C. 따옴표 사용 범위**

공백이 포함된 폰트명, 한글 폰트명, 문자열 데이터 타입, filter 속성의 파라미터 값은 작은따옴표(' ')로 감싸며, @charset 선언 시에는 속성 값을 큰따옴표(" ")로 감싼다.  
그 외의 경우에는 따옴표는 (“ ”) 사용한다.   
**<따옴표 사용 예>**

|  |
| --- |
| font-family:'돋움' filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='bg.png', sizingMethod='scale') background:url(“bg.gif”) no-repeat content:”Chapter” charset "utf-8"; |

**D. 마지막 세미콜론 사용 지양**

마지막 속성의 세미콜론(;)은 삭제한다.  
**<마지막 세미콜론 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class{font-size:12px;color:#000;} | .class{font-size:12px;color:#000} |

**5.2 들여쓰기**

CSS 코드를 작성할 때는 들여쓰기를 하지 않는다.단, 중괄호”{}”가 중첩되는 경우 예외

**<들여쓰기 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| #content{border:1px solid #000} #content .class{color:#000} | #content{border:1px solid #000} #content .class{color:#000} |
| @media all and (min-  width:480px){.vod\_wrp2{height:278px}} | @media all and (min-width:480px){  .vod\_wrp2{height:278px}  } |

**5.3 공백**

**A. 선택자 간 공백 제거**

쉼표로 구분되는 선택자 간 공백은 제거한다.

**<선택자 간 공백 제거 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| a:hover,^a:active,^a:focus{text-decoration:underline} | a:hover,a:active,a:focus{text-decoration:underline} |

**B. 속성 간 공백 제거**

속성 간 공백 및 속성 값 사이 공백은 제거한다.

**<속성 간 공백 제거 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class p{color:#000;^line-height:18px}  .class p{color:#000^ ;line-height:18px} | .class p{color:#000;line-height:18px} |
| font-family:'돋움',^dotum; | font-family:'돋움',dotum; |

**C. 중괄호 좌우 공백 제거**

중괄호 좌, 우 공백은 제거한다.

**<중괄호 좌, 우 공백 제거 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class p^{color:#000} | .class p{color:#000} |
| .class p{^color:#000; line-height:18px^} | .class p{color:#000;line-height:18px} |

**5.4 빈 줄**

의미 있는 객체를 구분하기 위하여 코드 그룹 간 1줄씩 빈 줄을 넣을 수 있다. 단, 빈 줄의 간격은 1줄을 초과하지 않게 한다.

**<빈 줄 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| /\* 의미 있는 그룹 1 \*/  .class1{border:1px solid #000}  .class1 img{border:1px solid #000}  빈 줄  빈 줄  /\* 의미 있는 그룹 2 \*/  .class2{border:1px solid #000}  .class2 img{border:1px solid #000} | /\* 의미 있는 그룹 1 \*/  .class1{border:1px solid #000}  .class1 img{border:1px solid #000}  빈 줄  /\* 의미 있는 그룹 2 \*/  .class2{border:1px solid #000}  .class2 img{border:1px solid #000} |

**5.5 줄 바꿈**

속성, 속성 값 사이 줄 바꿈은 허용하지 않는다.단, 선택자와 중괄호”{}”가 중첩되는 경우 예외

**<줄 바꿈 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class1,  .class2,  .class3{  width:  100px;  color:  #000  } | .class1,.class2,.class3{width:100px;color:#000}  .class1,  .class2, .  class3{width:100px;color:#000} |
| @media print{#header{display:none}} | @media print{  #header{display:none}  } |

**5.6 인코딩**

폰트 이름이 영문이 아닐 때 이를 브라우저에서 바르게 표현하고, HTML에서 불러온 스타일을 제대로 렌더링하려면 반드시 CSS 인코딩을 선언해야 한다. HTML과 동일한 인코딩을 문서 첫 줄에 공백 없이 선언한다. 작성 예는 다음과 같다.

|  |
| --- |
| @charset "utf-8"; |

파일을 저장할 때는 반드시 선언한 인코딩과 동일한 인코딩을 선택한다.

**5.7 선택자**

**A. 공통 선택자 사용 지양**

최 상위 공통 선택자 '\*'는 웹 페이지의 성능을 떨어뜨리고, Internet Explorer에서는 주석까지 영향을 받을 수 있으므로 사용하지 않는다.

**B. id 선택자 범위**

id 선택자는 2.2 id 및 class 네이밍 규칙에서 정의한 레이아웃(wrap, header, container, content, footer)에만 사용한다.

**C. 다중선택자 사용**

대응 범위에 인터넷 익스플로러 6.0 브라우저 환경이 포함될 경우 다중선택자 오류가 발생하므로 id와 class, class와 class 간의 다중선택자를 사용하지 않는다.

**<다중선택자 사용 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class{width:980px}  .class.bg{background:yellow}  .class.bg\_v1{background:green} | .class{width:980px}  .bg{background:yellow}  .bg\_v1{background:green} |

|  |
| --- |
| <p class=”class bg”>다중선택자</p> <p class=”class bg\_v1”>다중선택자</p> |

다중선택자란 두 선택자의 조합에 의해 속성이 부여되는 경우를 의미한다.  
예에서 background:yellow 속성은 .class와 .bg가 동시에 선언될 때만 의미가 있으며, background:green 속성은 .class와 .bg\_v1이 동시에 선언될 때만 의미가 있다.  
다중선택자를 사용하는 이유는 스타일 작성을 최소화 할 수 있으며, 여러 조합이 가능하기 때문이다.

**5.8 속성**

**5.8.1 속성 선언 순서**

속성을 선언할 때는 그 쓰임새가 레이아웃과 관련이 큰 것에서 시작하여 레이아웃과 무관한 것 순서로 선언한다. 관련 속성은 대표되는 속성 다음으로 선언하며, 표 5-11에 표기된 의미 순서대로 선언한다.

**<속성 선언 순서>**

|  |  |  |
| --- | --- | --- |
| **순서** | **의미** | **대표되는 속성** |
| 1 | 레이아웃 | display, visibility, overflow, float, clear, position, top, right, bottom, left |
| **2** | BOX | width, height, margin, padding, border |
| **3** | 배경 | background |
| **4** | 폰트 | font, color, letter-spacing, text-align, text-decoration, text-indent, vertical-align, white-space |
| **5** | 기타 | 위에 언급되지 않은 나머지 속성들로 폰트의 관련 속성 이후에 선언하며, 기타 속성 내의 선언 순서는 무관함 |

[속성 선언 순서 기준: 1: 레이아웃, 2: BOX, 3: 배경, 4: 폰트, 5: 기타]  
밴더속성과 핵속성은 해당 속성 뒤에 선언한다.

|  |
| --- |
| .btn{border:1px solid #f60;\*border:2px solid #f60;border-radius:2px;-webkit-boderradius:2px} |

**5.8.2 속성 값 축약**

CSS 최적화를 위해 다음과 같이 속성 값을 축약한다.

**<속성 값 축약 예>**

|  |  |  |
| --- | --- | --- |
| **축약 전** | **축약 후** | **설명** |
| #555555 | #555 | 16 진수 컬러 코드값 동일한 값으로 이루어진 16 진수 컬러 코드값은 세 자릿 수로 축약하여 사용한다. 단 인터넷 익스플로러 전용 속 성인 CSS filter 를 사용했을 경우 축약 속성을 인식하지 못하는 오류가 있기 때문에 속성 값을 축약하지 않는다. |
| #ff4400 | #f40 |
| backgroundposition:left center | background-position:0 50% | 위치값 문자로 표현한 위치값은 숫자로 변경한다 |
| top:0px | top:0 | 0 의 단위 속성 값이 0 일 경우 단위를 표시하지 않는다. |
| padding:20px 20px 20px 20px | padding:20px | 동일한 속성 값 상, 우, 하, 좌의 속성 값이 동일하면 축약한다. |
| margin:0 auto 0 auto | margin:0 auto |
| padding:20px 30px 50px 30px | padding:20px 30px 50px |
| border-color:#ccc #eee #ccc #eee | border-color:#ccc #eee |

**참고**

**문자로 표현된 위치 값을 숫자로 변환할 때 다음과 같이 한다.**

**top, left → 0**

**right, bottom → 100%**

**5.8.3 약식 속성 사용 범위**

border 와 background 는 약식 속성을 우선적으로 사용하며, font 약식 속성은 사용하지 않는다.

**A. border**

속성 값은 border-width, border-style, border-color 순서로 선언한다.  
테두리 스타일 속성을 초기 선언할 때는 반드시 border 단일 속성을 사용하고, 이후 테두리의 부분적인 속성이 변경될 경우 border 관련 속성(border-width, border-style, border-color)을 선언한다.

작성 예는 다음과 같다.

|  |
| --- |
| .class{**border**:1px solid #ccc} .class\_v1{**border-color**:#666} .class\_v2{**border-style**:dotted}  .class2 {**border-top**:1px solid #ccc} .class2\_v1{**border-top-color**:#666} .class2\_v2{**border-top-style**:dotted} |

테두리의 상, 우, 하, 좌 스타일이 2개 이상 다르면, 공통 스타일을 약식 속성으로 선언한 후 다른 부분은 관련 속성으로 선언한다.

<**테두리 스타일이 2개 이상 다를 경우 약식 속성 선언의 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| .class{border:1px solid #ddd;border-bottom:1px solid #eee;border-left:1px solid #eee} | .class{border:1px solid;border-color:#ddd #ddd #eee #eee} |
| .class{border-top:1px solid #ddd;border-right:1px dotted #ddd;border-bottom:1px solid #eee;borderleft:1px dotted #eee} | .class{border:1px;border-style:solid dotted;bordercolor:#ddd #ddd #eee #eee} |

**B. background**

속성 값은 background-color, background-image, background-repeat, background-position, background-size, background-attachment, background-origin, background-clip 순서로 선언한다.  
배경 스타일 속성을 초기 선언할 때는 반드시 background 단일 속성을 사용하며, 이후 배경의 부분적인 속성이 변경될 경우 background 관련 속성(background-color, background-image, background-repeat, background-position, background-size, background-attachment, background-origin, backgroundclip)을 선언한다.

작성 예는 다음과 같다.

|  |
| --- |
| .class{background:#ccc url(bg.gif) no-repeat}  .class\_v1{background-position:0 -50px}  .class\_v2{background-position:0 -100px} |

**C. font**

폰트 스타일은 약식 속성으로 선언하지 않는다.  
폰트 스타일을 약식 속성으로 선언하면 다음과 같은 문제가 발생한다.  
아래와 같이 선언하면, font-weight:bold는 상속되지 않고 font 속성의 디폴트 값인 font-weight:normal로 변경되기 때문에 불필요한 속성 값을 선언해야 하는 문제가 발생한다.

|  |
| --- |
| .class{font-weight:bold;font-size:12px;font-family:dotum} .class p{font:15px gulim} |

결국, class p의 폰트 스타일은 아래와 같아진다.

|  |
| --- |
| .class p{font-family:gulim;font-style:normal;font-variant:normal;font-weight:normal;fontsize:15px;line-height:normal} |

**5.8.4 한글 폰트 선언**

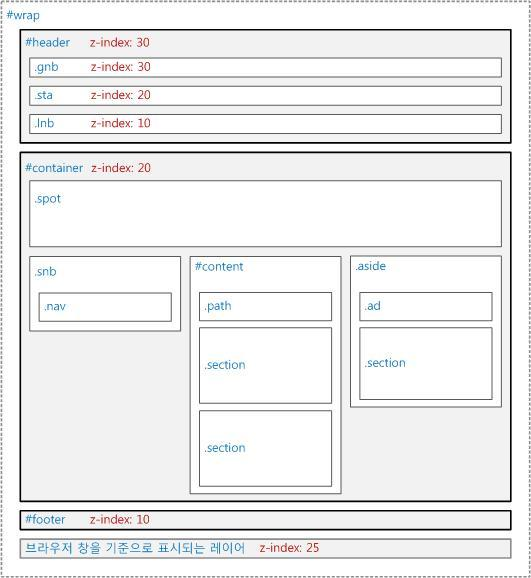
영문폰트만 선언할 경우 특정 브라우저에서 폰트를 올바르게 출력하지 못하는 경우가 있으므로 한글 폰트 선언 시 한글, 영문 폰트를 모두 선언한다.

**<폰트 선언 예>**

|  |  |
| --- | --- |
| **잘못된 예** | **올바른 예** |
| font-family:'돋움' | font-family:'돋움',dotum |

**5.9 z-index**

z-index 속성의 속성 값의 범위는 최소 10, 최고 1000이며, 10 단위로 증감한다. 단, 10 단위 사이의 예외 변수가 발생하면 1 단위 값을 지정할 수 있다.  
다음은 기본 가이드로, 각 서비스의 z-index값은 상황에 맞게 선택적으로 선언한다.

**<z-index 권장 선언값 예>**

**5.10 핵(hack)**

핵(hack)은 가능한 한 사용하지 않는다. 불가피하게 사용해야 할 때는 표 5-15에 제시된 핵만 사용하는 것을 원칙으로 한다.

**<사용 가능한 핵>**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **브라우저** | **표준 DTD** | | | **Quirks Mode** | | |
| IE6 | S{\*P:V} | S{\_P:V} |  | S{\_P:V} |  |  |
| IE7 | S,x:-moz-any-link, x:default {P:V} | \*+html S{P:V} | S,x:-moz-any-link, x:default{P:V} |
| IE8 | S{P:V\0} |  |  |
| IE9 | :root S{P:V\0 !importa nt} | :root S{P:V} |  | :root S{P:V} |
| Firefox |  | S,x:-moz-any-link, x:default{P:V} |  |
| Safari/ Chrome |  |  |
| Opera |

[약어 표기 – S(Selector): 선택자, P(Property): 속성, V(Value): 속성 값]

**5.11 미디어 타입**

미디어 타입 선언을 위한 명령문(중괄호 포함)은 다음과 같이 세부 스타일을 지정하는 명령문과 줄로 구분한다.  
미디어 타입은 기본 CSS 파일의 가장 아랫부분에 선언한다

|  |
| --- |
| /\* 반응형 \*/  @-ms-viewport{width: device-width} @viewport{width: device-width} @media screen and (min-width:420px){  #header,.snb,.aside{display:none} } |

**참고**

**미디어 타입은 각종 미디어(프린터, 모바일, 보조공학기기 등)에 대응하는 별도의 CSS 코드를 작성할 수 있게 한다. CSS 2.1 Specification에 따라 대응 가능한 미디어 타입은 아래와 같으며, 가장 범용적으로 사용하는 타입은 print 타입이다.**

**- all (모든 출력 장치)**

**- aural (음성 출력)**

**- braille (점자 출력)**

**- handheld (포켓, 모바일)**

**- print (인쇄)**

**- projection (투사 장치)**

**- screen(컴퓨터회면)**

**- tty(고정폭 글자를 사용하기위한 미디어타입)**

**- tv(텔레비전 타입)**

**5.12 CSS 선언 타입**

CSS 선언 타입은 크게 세 가지로 분류하며, 용도에 맞게 사용한다.

**A. External Type**

CSS를 선언하는 가장 기본적인 방식으로, CSS 파일이 별도로 존재하는 형태이다. link 요소를 통해 HTML과 CSS 파일을 연결한다.

작성 예는 다음과 같다.

|  |
| --- |
| <link rel="stylesheet" type="text/css" href="../css/service\_name.css"> |

**B. Internal Type**

HTML 파일의 head 안에 스타일을 선언하는 방식으로, 단발성 페이지의 CSS 분량이 작을 경우 사용한다.

작성 예는 다음과 같다.

|  |
| --- |
| <head>  …  **<style type="text/css">  … </style>** </head> |

**C. Inline Type**

HTML의 개별 요소에 style 속성을 이용하여 스타일을 선언하는 방식으로, 속성 값이 동적으로 변경되어야 하는 경우 사용한다. 속성 값에 사용되는 %와 같은 특수기호는 Character entity references로 변환하지 않는다.

작성 예는 다음과 같다.

|  |
| --- |
| <div **style="top:0;left:50%"**> …  </div> |

**5.13 주석**

**5.13.1 기본 형식**

CSS 주석은 아래와 같이 표기하며, 기본 형식에 맞게 작성한다.

|  |
| --- |
| /\* 주석 내용 \*/ |

• 주석 기호와 주석 내용 사이에는 반드시 공백 한 칸이 있어야 한다.  
• 주석 기호와 주석 내용 사이의 줄 바꿈은 허용하지 않는다.  
• 단, 주석 내용 간 줄 바꿈은 허용한다.  
• 종료 주석은 사용하지 않는다.

**5.13.2 작성자 정보 표기**

작성자 표기는 css인코딩 선언 다음줄에 1회만 작성하며, 작성자 정보에는 소속 부서, 영문 이름 이니셜, CSS 파일 생성 날짜(YYMMDD 형식)를 작성한다. 유지보수만 담당하는 경우라도 모두 기입한다. 작성자 정보 밑에는 빈 줄(한 줄)을 두어 스타일을 선언하는 문장과 구분되도록 한다. 동일 파일을 유지보수 하는 경우, 작성자 바로 아래 줄에 동일한 형식으로 수정내용을 포함하여 작성한다.

|  |
| --- |
| @charset "utf-8"; /\* FUZ UI.DEV JJS 170101, FUZ UI.DEV JJJ 170201 \*/ /\* FUZ UI.DEV KDW 170301 - 버튼수정 \*/ **빈 줄** |

**5.13.3 의미 있는 그룹 영역의 주석 표기**

의미 있는 객체를 구분하기 위한 주석은 영역의 윗부분에 표기한다. 초기화와 레이아웃 스타일 그룹을 제외한 의미 있는 그룹 영역의 주석 표기는 선택 사항이다.

작성 예는 다음과 같다.

|  |
| --- |
| **/\* 마이페이지 SNB \*/** .my\_snb{width:182px} .my\_snb li .num{padding-left:4px;color:#919190;font-size:11px;letter-spacing:0} .my\_snb li.on a,.my\_snb li.on .num{color:#259e0b} .my\_snb li a{color:#424242} |

**A. 초기화 스타일 그룹**

CSS 초기 파일에 따라 초기화 속성은 /\* Common \*/으로 그루핑한다.

|  |
| --- |
| **/\* Common \*/** body,p,h1,h2,h3,h4,h5,h6,ul,ol,li,dl,dt,dd,table,th,td,form,fieldset,legend,input,textare a,button,select{margin:0;padding:0} body,input,textarea,select,table{font-family:’돋움’,Dotum,AppleGothic,sans-serif;fontsize:12px} |

**B. 레이아웃 스타일 그룹**

레이아웃을 위한 스타일 선언 시 /\* Layout \*/으로 그루핑한다

|  |
| --- |
| /\* Layout \*/ #wrap{…} #header{…} #container{…} #footer{…} |

**5.14 초기 파일**

CSS를 새로 작성할 때는 아래 파일을 기본으로 한다. 초기 파일에는 스타일 속성 초기화 문장이 포함되어 있으며, 초기화 문장은 변경할 수 없다.

|  |
| --- |
| @charset "utf-8"; /\* fuz UI.DEV. JJS 090707, JJJ 090815 \*/  /\* Common \*/ body,p,h1,h2,h3,h4,h5,h6,ul,ol,li,dl,dt,dd,table,th,td,form,fieldset,legend,input,textare a,button,select{margin:0;padding:0} body,input,textarea,select,button,table{font-family:'돋움',Dotum, Helvetica,sansserif;font-size:12px} img,fieldset{border:0} ul,ol{list-style:none} em,address{font-style:normal} a{text-decoration:none} a:hover,a:active,a:focus{text-decoration:underline} .blind{visibility:hidden;overflow:hidden;position:absolute;top:0;left:0;width:1px;height: 1px;font-size:0;line-height:0} |