

Platform Development. | 최경환

# Ndoors Web UI Guideline v1.0

**2011. 01. 18** (Last Update 2011. 01. 18)



# 개정 이력



No	버전	변경일	변경 내용	작성자

# 목 차

1. Ndoors Web UI 정책
2. Document Type
3. Meta Data
4. HTML
5. Cascading Style Sheet
6. Javascript
7. Images name
8. Flash
9. Web Accessibility (웹 접근성)



# Ndoors Web UI 정책



1. 모든 Property의 UI는 Windows (98, ME, 2000, XP, Seven)의 IE 6.0 이상을 기준으로 한다.

- IE 8.0 – Win XP, Win vista, Win 7 (필수지원)
- IE 7.0 – Win XP, Win vista, Win 7 (필수지원)
- IE 6.0 – Win98, Win2000, Win XP (필수지원)
- Mozilla Firefox 3.6 (필수지원)

- Safari 5.0 (권장지원)
- Crome 5.0 (권장지원)
- Opera 10.0 (권장지원)

2. 필수지원 OS, 브라우저는 Function 100% 구현

3. 예외 / 특별한 사안에 대해서는 기술지원팀과 협의 한다.

# Document Type



Ndoors Document Type은 xhtml 1.0 Transitional을 기본으로 하여 작성한다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>TITLE</title>  
<link rel="stylesheet" href="marster.css" type="text/css" media="all" />  
</head>  
<body>  
<div id="wrapper">
```

Cording...

```
</div>  
</body>  
</html>
```

※ 다른 Document Type을 사용할 경우 기술지원팀과 협의한다.

※ 문서의 기본은 위와 같은 구조이고 구조와 표현의 분리를 위해 CSS는 External 방식으로 작성한다.

# Meta data



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> 문자셋 선언
<meta name="author" content="작성자" /> 작성자
<meta name="description" content="웹의 구조를 설명" /> 설명문
<meta name="keywords" content="web, www, 구조, html" /> 키워드

<meta name="copyright" content="저작권" />
<meta name="reply-to" content="연락처나 메일주소" />
<meta name="date" content="날짜(작성자)" />

<meta name="robots" content="noindex"> 어떤 로봇도 사이트의 페이지에 대해 색인 생성불가
<meta name="googlebot" content="noindex"> 구글을 제외한 다른 모든 로봇이 사이트의 페이지에 대해 색인 생성허용

<title>메타데이터제어</title>

</head>
```

■ 필수입력    ■ 권장입력    ■ 외부 색인 제어

1. 메타데이터는 title 위에 위치한다.
2. 필수입력 메타데이터는 반드시 입력하고 권장입력 메타데이터는 필요에 따라서 입력한다.
3. admin, 인트라넷 외부에 노출을 제한하는 용도의 사이트에 대해선 검색로봇을 제어하는 메타데이터를 사용한다.

# HTML



## 1. Html 목적에 맞는 tag를 사용한다.

- Element Spotlight tag : <em> <label> <dfn>
- 제목 : <h1> ~ <h6>
- text : <a> <p> <pre> <em> <strong> <cite> <address> <dfn> <br> <code>..
- Containers : <div> <span>
- 내용 : <p>
- List : <ol> <ul> <li> <dl> <dt> <dd>
- 특정 부분 참조 : <span>
- 양식 : <form> <input> <select> <option> <textarea> <button> <fieldset> <label>
- 표 형식의 데이터 표현 : <table> <tr> <th> <td> <thead> <tbody> <tfoot>
- <b> <u> <center> <font> xhtml로 넘어가면서 폐기된 태그는 사용금지

## 2. 주의사항

- 단위는 px로 지정한다. (예외사항 : 유동성을 고려해야하는 페이지)
- 모든 태그와 속성은 소문자로 작성한다.
- 모든 태그의 속성값은 "(따옴표)" 안에 작성한다.
- 공백 사용을 최대한 줄인다.
- 불필요한 tag를 줄인다.
- 각 요소들의 속성을 규정할 수 있도록 CSS를 적극 활용한다.
- 칼라 값 '#000000' 처럼 #을 붙여 표시한다.
- Indentation Tab 크기는 space 4로 설정하여 코딩 한다.
- 닫는 tag는 항상 존재한다.
- forms tag는 개발을 위한 요소이므로 디자인 구현을 위해 사용하지 않는다.
- 부모요소와 자식요소의 관계를 적절히 지정  
ex) <p><a href="http://www.web.co.kr/">웹</a></p>

# CSS



1. 웹페이지의 속도를 향상 시키기 위해 Markup과 CSS는 최소화 한다.

2. 외부 스타일시트를 권장한다.

ex) `<link rel="stylesheet" href="/css/master.css" type="text/css" media="all" />`

외부스타일 시트 방식을 이용하면 캐시의 영향으로 User 입장에서는 내부 스타일방식 보다는 웹페이지의 로딩속도가 빠르게 된다.

3. CSS를 페이지 상단에 놓는다.

- 브라우저들은 모든 css 파일들을 완전히 다운로드 하기 전까지는 그 페이지에서 아무것도 읽지 않는다.
- css 파일을 빨리 다운로드 할 수록, 페이지를 더 빨리 읽기 시작한다. 젤 위에 넣어서 텍스트나 이미지들이 css 요소들이 다운로드 되면서 읽히도록 한다.

4. 페이지의 http request 를 최소화 한다.

- 사용자가 경험하게 되는 전체 반응시간의 10-20%가 요구한 HTML 페이지를 받는데 소요되고 나머지 80-90%는 그 html 페이지의 모든 요소들(이미지, JS 파일, CSS등)을 가져 오기 위한 HTTP request 를 위해 쓰인다고 한다.
- 이러한 http request 수를 줄이는 것이 반응시간(response time)을 줄이는데 가장 큰 영향을 줄 것이고 또한 쉬운 방법이다.

5. 화이트 스페이스를 제거한다.

6. 단축형 속성을 사용한다

7. 속성은 레이아웃 관련 프로퍼티(position, display, clear, float)를 우선으로 하고 이미지, 폰트 스타일 등을 차선으로 한다.

8. 되도록 단축형 16진 칼라 값을 사용한다

다음페이지 계속...



# CSS



7. 아래와 같이 축약형을 사용한다.

```
#nddoors{background:#fff;color:#ccc;font:bold 12px 굴림;} (o)
```

```
#nddoors{  
    background-color:#ffffff;  
    color:#cccccc;  
    font-family:굴림;  
    font-size:12px;  
    font-weight:bold;  
} (x)
```

8. 상속을 이용한다.

9. 스타일 선언을 그룹화 하는 다중의 class 를 사용한다.

```
<style type="text/css">  
.cl1{color:#900;}  
.cl2{color:#900;}  
.itl{font-family:Dotum;}  
.non{font-weight:bold;}  
</style>
```

```
<span class="cl1 non">TEST</span>
```

```
<span class="cl2 itl">TEST</span>
```

10. 외부 스타일시트 작성법

```
<link rel="stylesheet" href="/css/master.css" type="text/css" media="all" />
```

- master.css의 안에는 용도별로 css를 정리해 놓는다.

@charset="utf-8";	← 문자셋 선언
@import url("common.css");	← 공통적으로 들어갈 CSS
@import url("layout.css");	← 레이아웃만을 모아놓은 CSS
@import url("main.css");	← 메인페이지 작업만을 모아놓은 CSS
@import url("contents.css");	← 서브페이지 작업만을 모아놓은 CSS
@import url("board.css");	← 게시판 작업만을 모아놓은 CSS
@import url("popup.css");	← 팝업에 적용될 CSS

# Javascript



1. JS 부분은 가능한 한 페이지 아래에 넣는다.

- Js 파일들은 그 파일 뒤에 있는 페이지 읽기와 다운로드를 중단 시키므로 최대한 아래쪽에 넣는다.

2. JS 코드를 최소화 하거나 까다롭게 만든다.

- Javascript 코드를 까다롭게 만든다는 것은 대부분 요소들과 빈칸을 제거하는 것과 긴 심볼네임을 짧은 스트링으로 대체하는 둘 다를 포함한다. 퍼포먼스 측면에서 봤을 때 유저들이 다운로드 할 JS 파일의 사이즈를 줄이는 것이 중요하다.

3. 같은 JS component를 한 번 이상 include 하지 않는다.

4. IE 6.0, IE 7.0 , IE 8.0, Firefox, Crome, Safari, Opera에서 100% 구현 가능해야 한다.

5. Css를 제거 한 상태에서도 기능이 정상적으로 작동해야 한다.



# Image Name

1. 이미지 파일명은 파일의 의미 1자 파일의 성격 액션.gif 의 형식으로 한다.

ex) b\_write.gif , b\_write\_on.gif , b\_write\_off.gif

2. 파일의 성격이 같고 디자인이 다를 경우는 01, 02의 형식으로 추가한다.

ex) b\_write01.gif , b\_write02.gif ...

3. 이미지 파일을 수정해서 올릴 경우 이미지명\_v2.gif 의 형식으로 버전관리를 하도록 한다.

ex) v\_banner\_v2.gif

4. 이미지 종류 접두어

이미지종류	접두어	비고
Background	k_	배경이미지
Button	b_	버튼이미지
Menu	m_	메뉴이미지, 탭...
Bullet	u_	Dot, arrow, bar.. 등 의미를 갖지 않고 텍스트와 함께 사용되는 이미지
Banner	r_	배너이미지
Line	l_	CSS로 표현 불가능한 line 이미지
Icon	i_	이미지 마다 각각의 의미를 갖는 텍스트와 함께 사용되는 이미지
Title	t_	페이지나 모듈상에 타이틀로 표현되는 이미지
Visual	v_	컨텐츠에 고정으로 박히는 이미지
Sample	x_	개발상에서 뿌려지는 이미지를 html에 표현할 때 임의로 사용할 공갈이미지

# Flash



## 1. 자바스크립트와 제이쿼리를 사용하여 삽입하는 방법

- <head>에 엔도어즈에서 기본적으로 쓰이는 common.js와 제이쿼리를 삽입한다.

```
<script src="/Common/javascript/common.js" type="text/javascript"></script>
<script src="/Common/javascript/jquery-1.4.2.min.js" type="text/javascript"></script>
```

- 플래시를 호출하는 스크립트는 </body> 바로 위에 넣어준다.

```
<script type="text/javascript">
  //
    $(document).ready(function() {
      $("#nddoors").html(returnFlashContents("nddoors.swf", "nddoors", "900", "550", "transparent", "&lt;param name='menu' value='false' /&gt;&lt;param name='allowScriptAccess' value='always' /&gt;"));
    });
  //]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="50 509 797 552" data-label="List-Group"><ul><li>- 지정된 id값을 구조화된 마크업에 지정해 준다.</li><li>- 스크립트가 작동하지 않는 환경을 고려하여 &lt;noscript&gt;안에 플래시와 관련된 이미지나 링크, 대체 텍스트를 넣어준다.</li></ul></div><div data-bbox="50 569 490 629" data-label="Text"><pre>&lt;div id="nddoors"&gt;
  &lt;noscript&gt;&lt;img src="movie.jpg" alt="영상 이미지" /&gt;&lt;/noscript&gt;
&lt;/div&gt;</pre></div><div data-bbox="50 668 337 693" data-label="Section-Header"><h2>2. Object 태그를 사용하여 삽입하는 방법</h2></div><div data-bbox="50 712 787 755" data-label="List-Group"><ul><li>- data의 swf 경로와 param name="movie"의 value 경로를 동일하게 한다.</li><li>- &lt;p&gt;태그 안에는 스크립트가 작동하지 않았을 때의 상황을 고려하여 관련된 이미지나 링크, 대체 텍스트를 넣어준다.</li></ul></div><div data-bbox="50 773 850 916" data-label="Text"><pre>&lt;object type="application/x-shockwave-flash" data="nddoors.swf" width="500" height="300"&gt;
  &lt;param name="movie" value="nddoors.swf" /&gt;
  &lt;param name="wmode" value="transparent" /&gt;
  &lt;param name="menu" value="false" /&gt;
  &lt;p&gt;해당 콘텐츠를 보려면 &lt;a href="http://www.adobe.com/kr/products/flashplayer/"&gt;Flash Player&lt;/a&gt;가 필요합니다.&lt;/p&gt;
  &lt;p&gt;&lt;img src="contents.jpg" alt="내용"&gt;&lt;/p&gt;
&lt;/object&gt;</pre></div><div data-bbox="3 969 195 986" data-label="Page-Footer"><p>Copyright © NDOORS Corporation. All Rights Reserved.</p></div><div data-bbox="707 962 863 988" data-label="Page-Footer"><p>모든 사람이 즐거워지는 세상!</p></div><div data-bbox="875 943 985 988" data-label="Page-Footer"><p><img alt="NDOORS logo" data-bbox="875 943 915 988"/> <b>NDOORS</b> <small>www.nddoors.net</small></p></div>
```

# Web accessibility (웹접근성)



엔도어즈 웹접근성은 "인터넷 웹콘텐츠 접근성 지침" 을 기준으로 한다.

1. 이미지의 의미나 목적을 이해할 수 있도록 적절한 대체텍스트를 제공해야한다.

ex) 

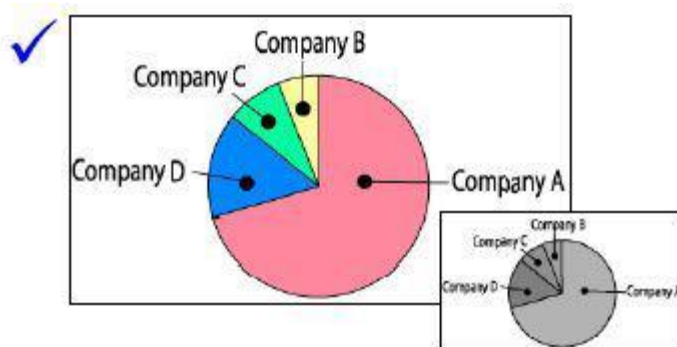
```

<map name="language">
<area alt="English" shape="rect" coords="10,5,66,19" href="/en/">
<area alt="Korean" shape="rect" coords="10,17,66,32" href="/ko/">
</map>
```

2. 의미가 있는 이미지를 배경 이미지로 활용하지 않는 것을 권장하며, 만약 사용할 경우에는 반드시 대체 콘텐츠를 제공해야 한다.

- CSS를 제거하여 배경이미지의 대체 정보를 제공하지 여부를 평가  
(CSS 제거 방법 : Firefox 브라우저 메뉴 '보기' -> '문서 스타일' -> '스타일제거' 선택)
- 스타일 제거 후 정보가 동일하면 준수
- 스타일 제거 후 정보가 동일하지 않으면 미준수

3. 색상을 배제하여도 원하는 내용을 전달할 수 있도록, 색상 이외에도 명암이나 패턴 등으로 콘텐츠 구분이 가능해야 한다.



<올바른 적용(O) : 색상이 아닌 화살표 등을 통해 정보를 인식>



<잘못된 적용(X) : 색상만으로 콘텐츠를 구분할 수 있도록 제공>

다음페이지 계속...

# Web accessibility (웹접근성)



4. 프레임을 제공할 경우, 해당 내용을 이해할 수 있도록 적절한 제목(title 속성)을 제공해야 한다.

ex) `<iframe title="엔도어즈" name="briefing" src="http://www.ndoors.com/"></iframe>`

5. 키보드 만으로 해당 콘텐츠의 포커싱이 가능하도록 제공해야 한다.

6. 메인페이지에 가능한한 팝업창을 사용하지 않아야 하며 레이어 팝업을 사용 했을 시 키보드 접근이 가능하도록 해야한다.

7. 데이터 테이블을 제공할 경우, 테이블의 내용을 이해할 수 있는 정보(제목, 요약정보 등)를 제공해야 한다.

ex) `<table class="data" summary="군주온라인, 아틀란티카, 불멸온라인">  
<caption>엔도어즈의 게임들</caption>`

8. 온라인 서식을 제공할 경우, 레이블(<label>)을 제공해야 한다.

ex) `<label for="uid">아이디</label><input type="text" id="uid" />` ※<label>태그의 for와 서식의 id가 동일하여야 함.

9. 온라인 서식을 제공할 경우, 레이블(<label>)을 제공해야 한다.

10. 플래시를 제공하는 경우, 해당 플래시가 자체적인 접근성을 준수하거나 사용자가 대체 콘텐츠를 선택하여 이용할 수 있어야 한다.

ex)

```
<object type="application/x-shockwave-flash" data="ndoors.swf" width="500" height="300">  
  <param name="movie" value="ndoors.swf" />  
  <param name="wmode" value="transparent" />  
  <param name="menu" value="false" />  
  <p>해당 콘텐츠를 보려면 <a href="http://www.adobe.com/kr/products/flashplayer/">Flash Player</a>가 필요합니다.</p>  
  <p></p>  
</object>
```



# End of Document

※ 이 문서는 공유, 배포가 가능 합니다.