

드림위버를 활용한 웹 표준 및 접근성 향상기법 소개



2006.10.28

정 찬 명

소속 : (주)지토피아

블로그 : <http://naradesign.net/>

메일 : webmaster@naradesign.net

MSN : dece24@hotmail.com

Nate : dece24kr@nate.com

목 차

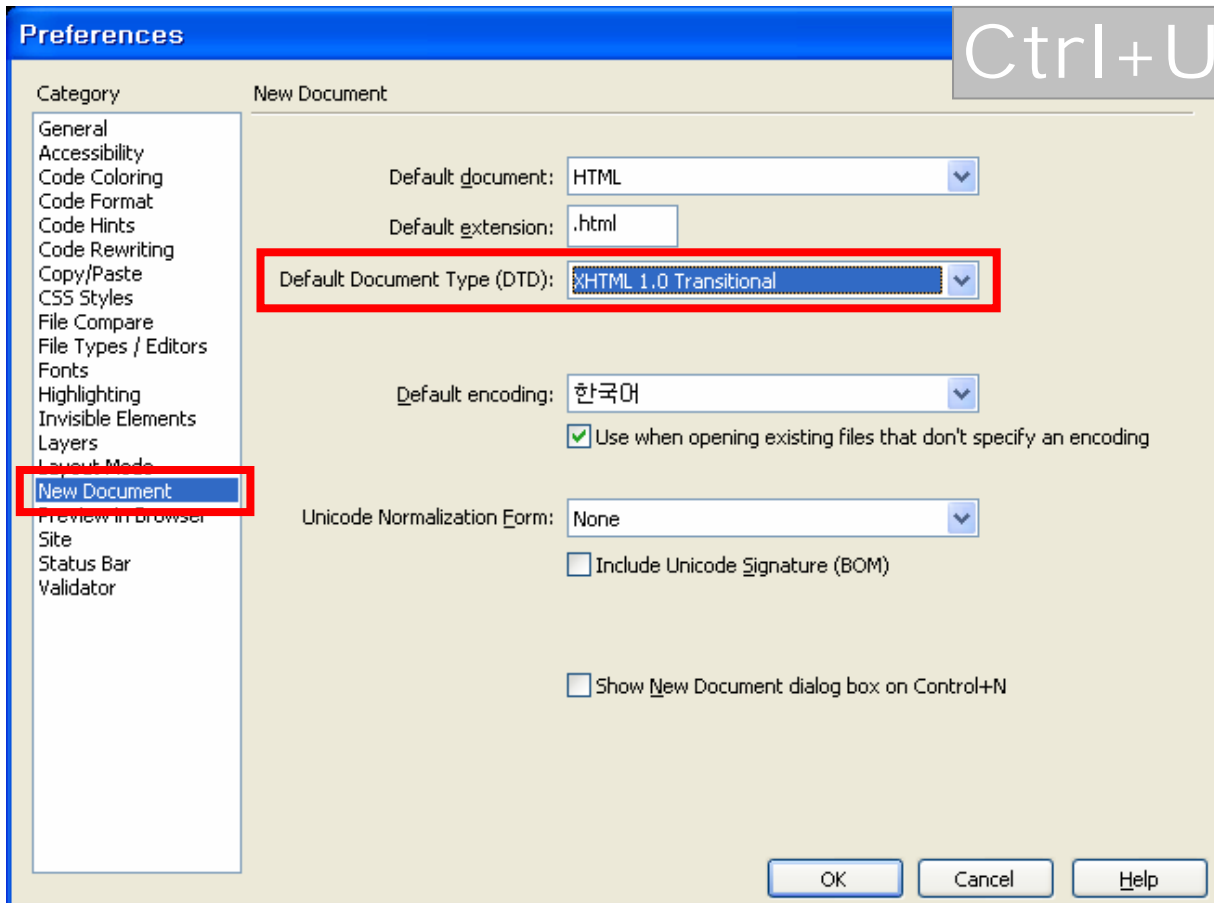
- 1. 문서 형(DTD) 정의(W 3.2)**
 - 가. 기본 문서 형(DTD) 정의를 위한 환경설정
 - 나. 문서 형(DTD) 바꾸기
- 2. 대체문자 입력하기(W 1.1, 9.1, KW 1.1, 2.1)**
 - 가. 대체문자 입력을 위한 환경설정
 - 나. 대체문자 입력
 - 다. 이미지 맵에 대체문자 입력하기
 - 라. 이미지 필드<input type="image">에 대체문자 입력하기
- 3. 서체크기에 대한 확대/축소 지원하기(W 3.4)**
 - 가. 줌 브라우저가 가능한 서체크기 지정하기
- 4. 제목의 마크업과 CSS 디자인(W 3.5)**
 - 가. 제목태그 <h1>~<h6> 를 활용한 제목 마크업
 - 나. 제목태그 디자인을 위한 CSS의 활용
- 5. 목록의 마크업과 CSS 디자인(W 3.6)**
 - 가. 배경이미지를 활용한 목록의 블릿 디자인
- 6. 표의 마크업과 CSS 디자인(W 5.1, 5.2, 5.5, KW 3.1)**
 - 가. 접근성 높은 표 데이터 마크업 하기
 - 나. CSS를 이용한 표 디자인
- 7. 폼의 마크업과 CSS 디자인(W 10.2, 12.3, 12.4, KW 3.3)**
 - 가. 폼 컨트롤 레이블링 하기
 - 나. 폼 컨트롤 요소를 논리적으로 그룹 짓기
- 8. 마크업에 대한 유효성 검사(W 11.1, 11.2)**
 - 가. 마크업 유효성 검사를 위한 드림위버 환경설정
 - 나. 드림위버를 이용한 마크업 유효성 검사
- 9. 마크업과 CSS에 대한 브라우저 호환성 검사(KW 4)**
 - 가. 브라우저 호환성 검사를 위한 환경설정 및 검사
- 10. 미국 재활법 508조 및 WCAG 1.0 접근성 검사**
 - 가. 미국 재활법 508조 및 WCAG 1.0 지침 준수여부 검사
- 11. 프레임을 사용하는 경우(W 12.1, KW 2.2)**
 - 가. '어쩔 수 없이' 프레임을 사용한다면 프레임 요소에 "title" 속성을 사용
- 12. 모호한 링크를 보다 선명하게 만들기(W 13.1)**
 - 가. 태그 편집기를 이용하여 하이퍼링크에 "title" 속성 넣기
- 13. 메뉴 건너뛰기 기능의 지원(W 13.6, KW 2.5)**
 - 가. 시각장애인을 위한 메뉴 건너뛰기 링크
- 14. 페이지 디자인을 위한 CSS 기초(W 3.3, 14.3, KW 3.2)**
 - 가. 페이지 레이아웃을 위한 CSS 기초
 - 나. 다 기종 브라우저간 렌더링 호환성 유지를 위한 CSS 팁
- 15. 출력을 위한 CSS 문서 만들기**
 - 가. 콘텐츠 영역만 출력해주는 CSS 만들기 팁
- 16. 효율적인 CSS 사용을 위한 드림위버 최적화**
 - 가. 대표속성 사용 및 CSS 코드를 위한 사용자 환경설정

1. 문서 형(DTD) 정의 (W 3.2)

가. 기본 문서 형(DTD) 정의를 위한 환경설정

WCAG 3.2 공식 문법에 맞도록 문서를 작성하라. [중요도 2]

예를 들면, 문서의 시작 부분에는 기존에 정의된 DTD(예를 들면, strict HTML 4.0 DTD)를 참조하도록 문서 형식(document type)을 선언해 주어야 한다.



화면설명

웹 문서형식에 대한 W3C의 최신 권고 표준안은 XHTML 1.x 버전이므로 이 지침을 적용한다.

드림위버의 환경설정을 통하여 새 문서의 기본 문서 형(DTD)이 항상 XHTML 1.0 Transitional.dtd 으로 설정되도록 한다.

작업순서

1. Preferences (Ctrl+U).
2. New Document 선택.
3. Default Document Type (DTD).
4. XHTML 1.0 Transitional 선택.

1. 문서 형(DTD) 정의(W 3.2)

나. 문서 형(DTD) 바꾸기

The image shows a browser window with two document types: HTML 4.01 Transitional (top) and XHTML 1.0 Transitional (bottom). A red arrow points from the HTML 4.01 DTD code to the XHTML 1.0 DTD code. Below the browser, the Page Properties dialog box is open, showing the Document Type (DTD) dropdown menu set to XHTML 1.0 Transitional. The dialog box also shows the Title/Encoding tab selected, the encoding set to Korean, and the site folder path.

HTML 4.01 Transitional

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
5 <title>:: ASSE/ANALAB 2006 - 한국 국제 연구분석실험기기 전시회 2006 ::</title>
```

XHTML 1.0 Transitional (ML)*

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr" />
5 <title>:: ASSE/ANALAB 2006 - 한국 국제 연구분석실험기기 전시회 2006 ::</title>
```

Page Properties

Ctrl+J

Category: Title/Encoding

Title: Untitled Document

Document Type (DTD): XHTML 1.0 Transitional

Encoding: 한국어

Unicode Normalization Form: None

Include Unicode Signature (BOM):

Document folder: Site folder: D:\₩드림위버를 활용한 접근성 제고방안₩

Buttons: OK, Cancel, Apply, Help

화면설명

기존의 문서 형 정의가 지난 버전 (HTML4.x)을 참조하고 있다면 최신의 문서 형으로 변경할 수 있다.

문서 형 정의를 변경하게 되면 기존의 마크업 코드는 자동으로 사용자가 선택한 DTD형식에 맞게 변경된다.

대문자 마크업은 소문자로, 종료태그가 없는 마크업은 종료태그를 삽입해줌. 단, Deprecated 된 코드를 사용한 경우 이것을 자동으로 삭제하지는 않음에 주의.

작업순서

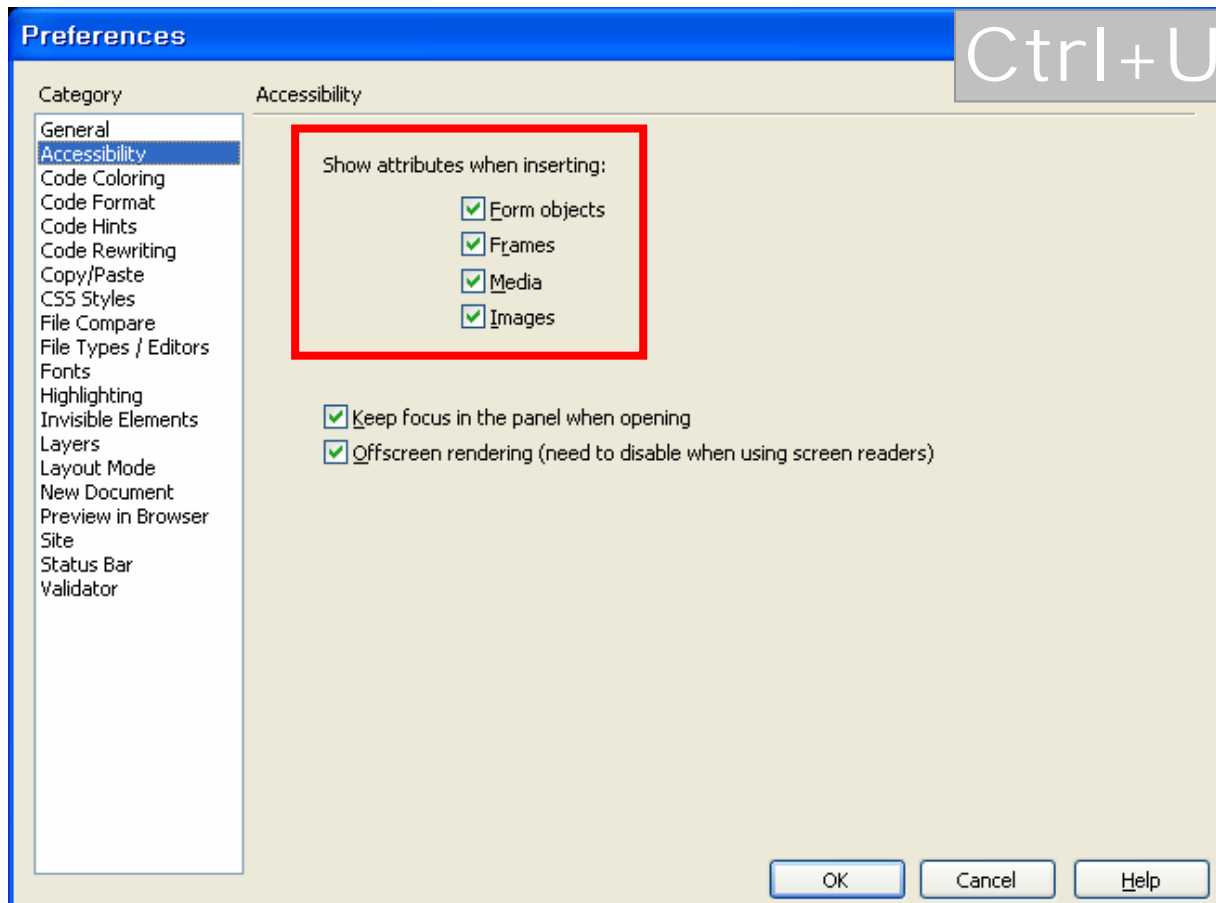
1. 오래된 HTML 4.x 버전의 문서 열기
2. Page Properties (Ctrl+J).
3. Title/Encoding 선택.
4. Document Type (DTD) 항목 확인.
5. XHTML 1.0 Transitional 으로 변경.
6. 변경된 DTD에 준하는 마크업으로 수정된 코드를 확인하고 Deprecated 된 코드는 직접 찾아 바꾼다.

2. 대체문자 입력하기 (W 1.1, 9.1, KW 1.1, 2.1)

가. 대체문자 입력을 위한 환경설정

WCAG 1.1 모든 비-텍스트 콘텐츠에 대해서는 대체 텍스트를 제공한다.

KWCAG 1.1 텍스트 아닌 콘텐츠 중에서 글로 표현될 수 있는 모든 콘텐츠는 해당 콘텐츠가 가지는 의미나 기능을 동일하게 갖추고 있는 텍스트로도 표시되어야 한다.



화면설명

드림위버의 환경설정 메뉴에는 접근성 항목이 존재한다.

폼 요소, 프레임, 미디어, 이미지 요소를 삽입하는 경우 접근성 관련 속성을 삽입할 수 있는 대화상자가 자동으로 나타나도록 설정할 수 있다.

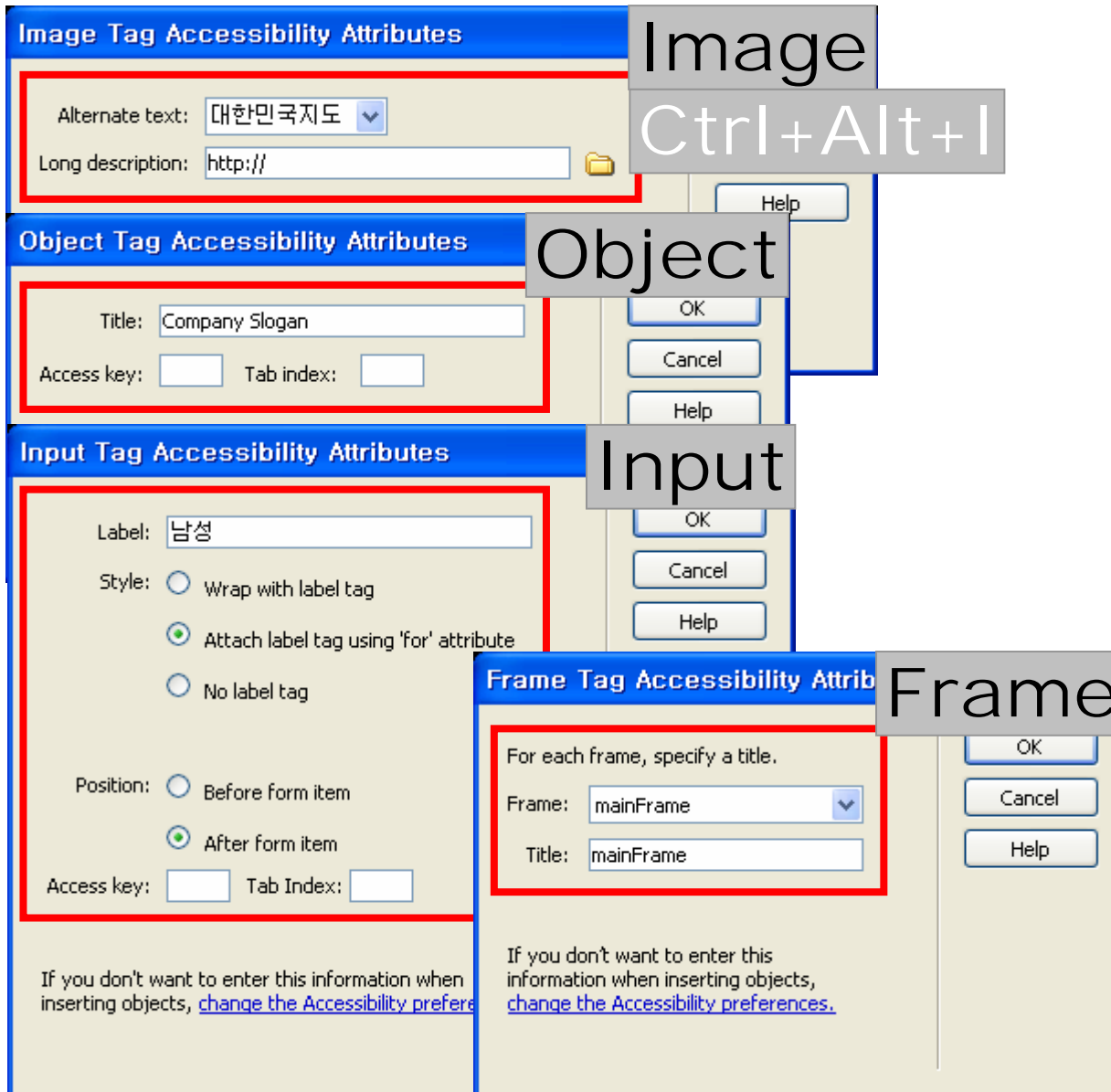
예를 들면 요소를 삽입할 때 alt 또는 longdesc 속성을 삽입할 수 있는 대화상자가 나타난다.

작업순서

1. Preferences (Ctrl+U).
2. Accessibility.
3. Show attributes when inserting.
4. Form object, Frames, Media, Images 항목에 모두 체크한다.

2. 대체문자 입력하기 (W 1.1, 9.1, KW 1.1, 2.1)

나. 대체문자 입력



화면설명

환경설정을 통하여 접근성 항목에 모두 체크하는 경우 보기와 같이 대체문자를 입력할 수 있는 대화상자가 나타난다.

위로부터 , <object>, <input>, <frame> 요소를 삽입할 때 등장하게 되는 접근성 속성 대화상자이다.

작업순서

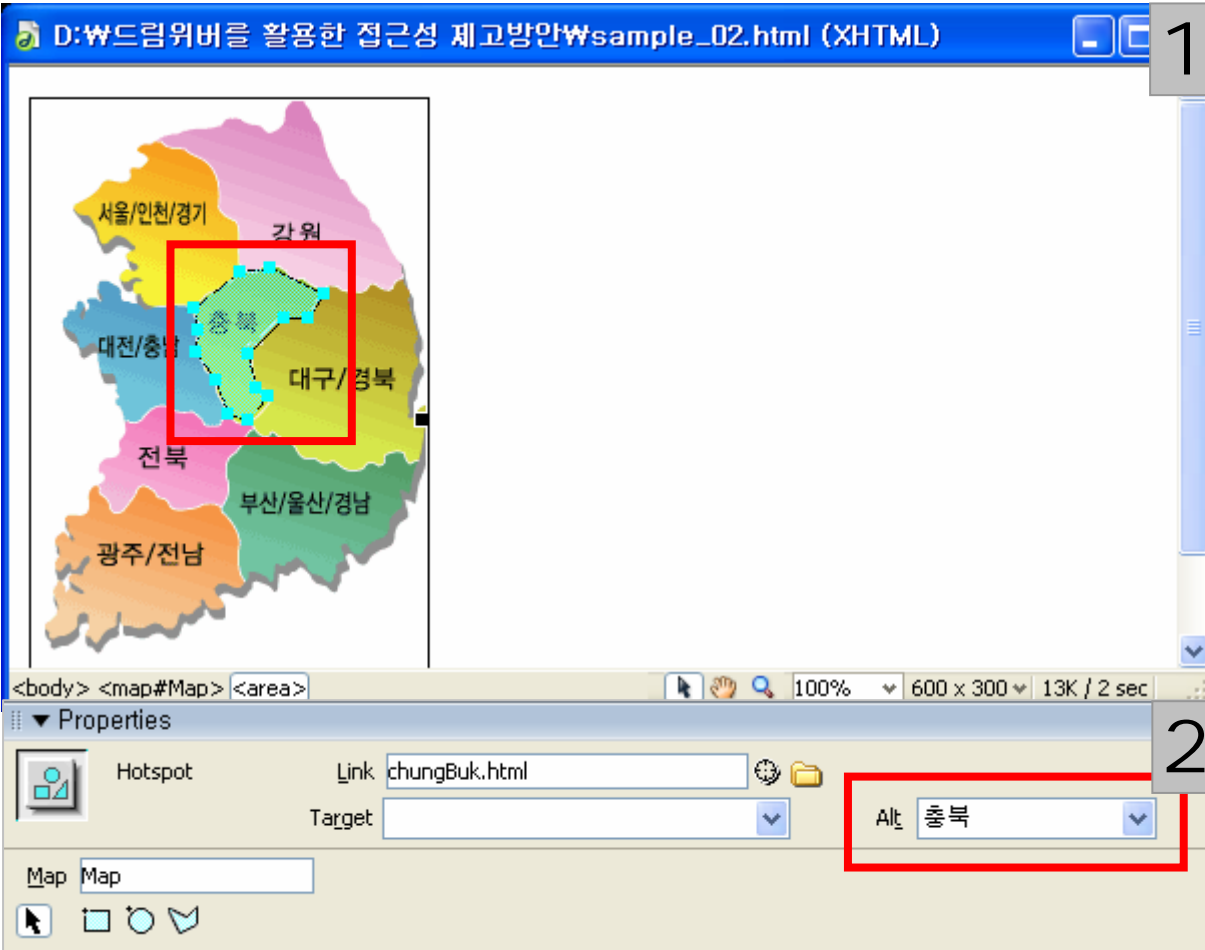
1. , <object>, <input>, <frame> 요소를 문서에 삽입.
2. Tag Accessibility Attributes 대화상자 등장.
3. 태그 요소에 따라 alt, title, label 등의 속성을 입력.

2. 대체문자 입력하기 (W 1.1, 9.1, KW 1.1, 2.1)

다. 이미지 맵에 대체문자 입력하기

WCAG 9.1 기하학적인 도형으로 정의가 안 되는 영역이 있는 경우를 제외하고는 서버측 이미지 맵 대신에 반드시 클라이언트측 이미지 맵을 사용한다.

KWCAG 2.1 이미지 맵 기법이 필요할 경우에는 클라이언트측 이미지 맵을 사용하며 서버측 이미지 맵을 사용할 경우에는 동일한 기능을 하는 텍스트로 구성된 대체 콘텐츠를 제공해야 한다.



The screenshot shows a web browser window with the title "D:\워드림위버를 활용한 접근성 제고방안\sample_02.html (XHTML)". The main content is a map of South Korea divided into regions. A red box highlights the Chungcheong region (충북). The Properties panel is open, showing the "Hotspot" section with "Link" set to "chungBuk.html" and "Target" set to "_self". The "Alt" attribute is set to "충북".

화면설명

이미지 맵은 꼭 필요할 때에만 사용하는 것이 좋다.

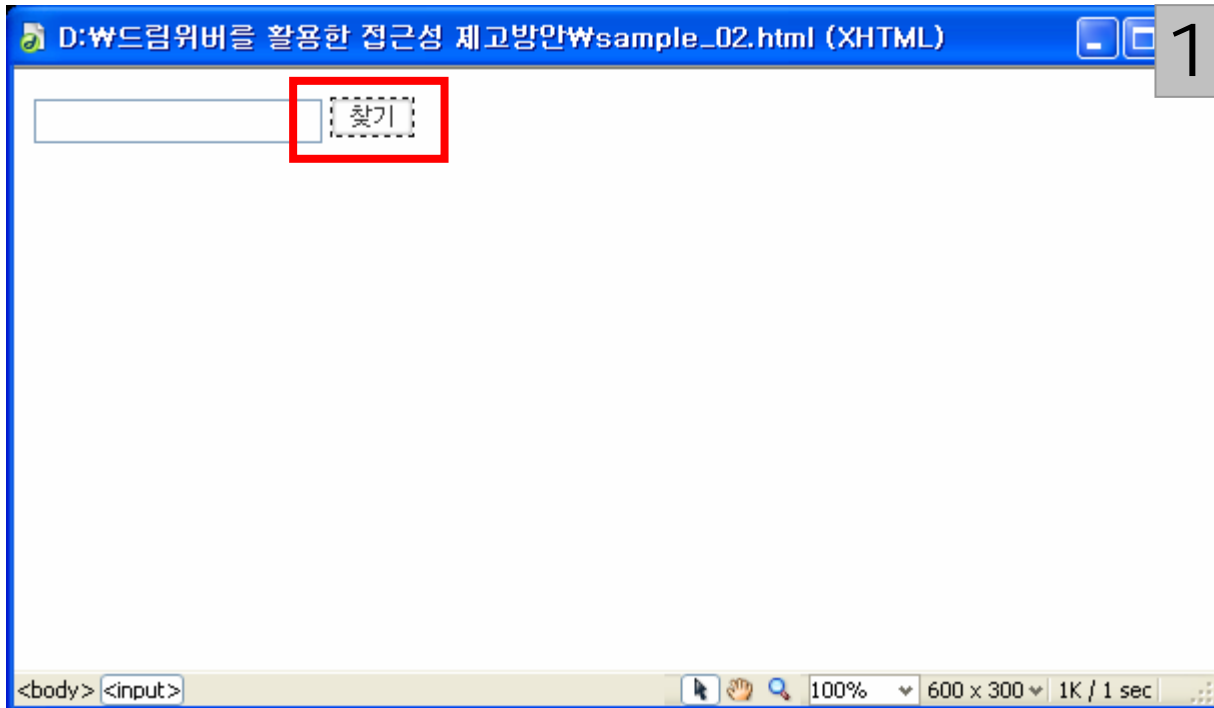
이미지 맵을 사용하는 경우 별도의 접근성 속성 대화상자가 등장하지 않기 때문에 Properties 패널을 이용하여 alt 속성을 입력하면 된다.

작업순서

1. 이미지의 특정 영역을 이미지 맵으로 처리.
2. Properties 패널을 이용하여 alt 속성을 작성.

2. 대체문자 입력하기 (w 1.1, 9.1, KW 1.1, 2.1)

라. 이미지 필드 <input type="image">에 대체문자 입력하기



화면설명

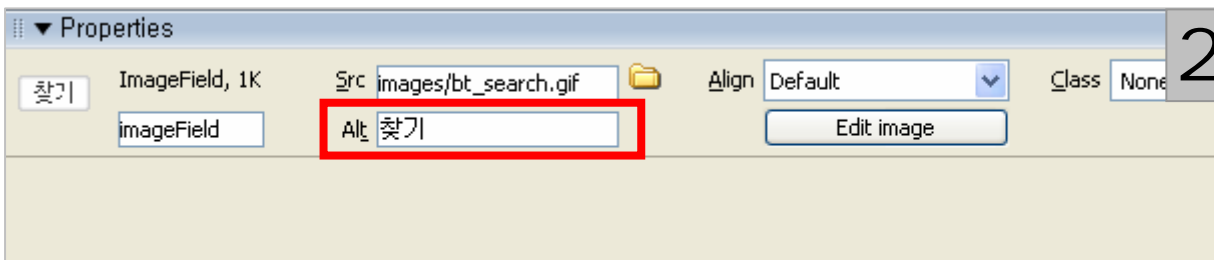
흔히 폼 요소 내부에 들어가는 버튼 목적의 이미지 필드 요소는 label 태그를 필요로 하지 않는다.

보기와 같이 label 태그의 사용이 불필요한 이미지 필드의 경우라면 alt 속성을 반드시 넣어 주어야 한다.

Properties 패널에서 이미지 필드에 대한 alt 속성을 입력할 수 있다.

작업순서

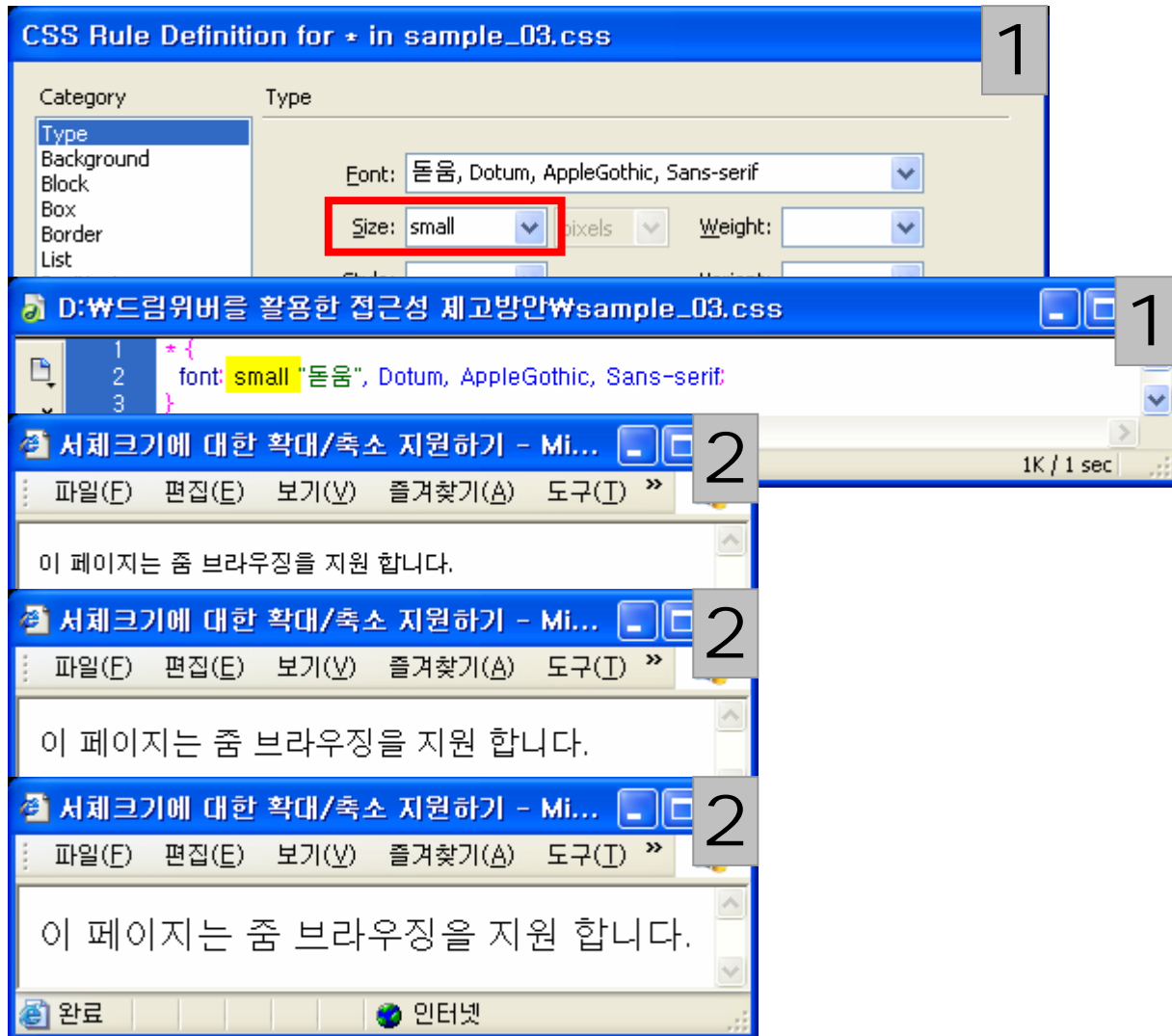
1. label 이 없는 이미지 필드 삽입.
2. Properties 패널을 이용하여 alt 속성을 작성.



3. 서체크기에 대한 확대/축소 지원하기 (W 3.4)

가. 줌 브라우저가 가능한 서체크기 지정하기

WCAG 3.4 마크업 언어의 속성값이나 스타일 시트의 속성값에는 절대 단위를 쓰지 말고 상대 단위를 사용한다.



화면설명

대부분의 웹 브라우저가 줌 브라우저 기능을 지원하고 있다. (Ctrl+MouseWheel, Ctrl+[+], Ctrl+[-], Ctrl+0)

그러나 IE의 경우 서체크기를 절대값으로 지정해 놓으면 줌 브라우저를 지원하지 않는 문제가 있다. 서체크기를 상대단위로 지정할 필요가 있다.

px, pt, cm 등의 단위는 절대값 단위이며 small, 0.75em, 75% 등의 단위가 상대값 단위이다. 0.75em=75%=small 은 같은 크기이며 px 으로 환산하면 12px 정도 크기가 된다.

작업순서

1. 공통서체의 크기를 상대단위 (small, em, %)로 지정.
2. 줌 브라우저 여부를 확인.

4. 제목의 마크업과 CSS 디자인(W 3.5)

가. 제목태그 <h1>~<h6> 를 활용한 제목 마크업

WCAG 3.5 문서의 구조를 나타내기 위해 제목 요소(header elements)를 쓰되, 사양에 맞게 사용한다.

The screenshot shows a website layout with the following header elements highlighted:

- <h1>**: Logo for Earth & Geography Information Technology.
- <h2>**: Main navigation bar with items like '회사소개' (Company Overview), '홍보마당' (PR/IR Information), '사업현황' (Business Field), and '사업' (Business).
- <h3>**: Sub-section header for '경영이념 및 비전' (Business Philosophy and Vision).
- <h4>**: Sub-section header for '01 사업목표' (Business Domain).

At the bottom of the screenshot, a box indicates the keyboard shortcuts: **Ctrl+1 ~ Ctrl+6**.

화면설명

제목태그를 이용한 마크업은 스크린 리더의 음성변화(고저)에 영향을 주기도 하고 검색로봇에게 긍정적인 영향을 주기도 한다. (헤더태그도 중요하지만 검색로봇은 <title> 태그에 더 높은 가중치를 부여한다. 매우 흔한 키워드인 “회사소개+경영이념”을 구글로 검색해 본다)

예제는 <h1>~<h4> 까지 헤더태그로 마크업 되어있다. 텍스트 뿐만 아니라 이미지라도 헤더태그로 마크업 하는 것이 적절하다.

드림위버는 헤더태그에 대하여 Ctrl+1 에서 Ctrl+6 까지 단축키를 부여하고 있다.

작업순서

1. 텍스트 또는 이미지 삽입.
2. 헤더 영역 또는 이미지를 선택.
3. 단축키 Ctrl+[1~6] 을 이용하여 헤더 마크업.

<http://engit.com/> 예제 사이트를 통하여 제목태그의 마크업을 확인.

4. 제목의 마크업과 CSS 디자인(W 3.5)

나. 제목태그 디자인을 위한 CSS의 활용

The screenshot displays three windows from a web browser:

- Rendering:** Shows a list of six titles: 제목1, 제목2, 제목3, 제목4, 제목5, and 제목6. Each title is preceded by a play button icon.
- Markup:** Shows the HTML code for the titles:

```
13 <body>
14 <h1>제목1</h1>
15 <h2>제목2</h2>
16 <h3>제목3</h3>
17 <h4>제목4</h4>
18 <h5>제목5</h5>
19 <h6>제목6</h6>
20 </body>
```
- CSS:** Shows the CSS code for styling the titles:

```
1 /*마크업 초기화*/
2 * { margin:0; padding:0; font-family:"돋움", Dotum, AppleGothic, Sans-serif}
3 body { margin:20px}
4
5 /*제목*/
6 h1, h2, h3, h4, h5, h6 { display:block; margin-bottom:20px; line-height:100%;}
7 h1 {padding: 15px 0 7px 50px; background:#EEE url(images/h1.gif) no-repeat 5px center;
8 h2 {padding: 10px 0 6px 45px; background:#EEE url(images/h2.gif) no-repeat 5px center;
9 h3 {padding: 8px 0 5px 35px; background:#EEE url(images/h3.gif) no-repeat 5px center;
10 h4 {padding: 5px 0 5px 30px; background:#EEE url(images/h4.gif) no-repeat 5px center;
11 h5 {padding: 5px 0 5px 27px; background:#EEE url(images/h5.gif) no-repeat 5px center;
12 h6 {font-size:70%; padding: 5px 0 5px 23px; background:#EEE url(images/h6.gif) no-rep
```

화면설명

제목태그의 디자인을 CSS로 재 정의 하였으며 마크업 요소로부터 디자인 코드를 완전히 분리 하였다.

이렇게 분리된 디자인 파일(CSS)은 웹사이트를 개편할 때 HTML 문서의 수정을 최소화 하게 된다. 적어도 내용이 아닌 디자인을 변경하기 위하여 HTML 문서를 열어볼 필요는 없기 때문이다.

드림위버는 대화상자를 이용한 CSS 코드 작성을 지원하는 한편 전문가를 위한 손코딩 자동완성기능까지 모두 지원하고 있다.

결과코드

h1

```
{padding: 15px 0 7px 50px;
background:#EEE url(images/h1.gif)
no-repeat 5px center; border-
bottom: 6px solid #CCC;}
```

이하 생략...

4. 제목의 마크업과 CSS 디자인 (w 3.5)

나. 제목태그 디자인을 위한 CSS의 활용[계속]

1. 제목태그를 이용하여 콘텐츠를 마크업하고 태그 선택기에서 <h1> 태그를 선택.

2. CSS 패널의 New CSS Rule 클릭.

3. 선택자 타입을 Tag로 지정하고 h1 태그가 선택된 것을 확인. 새로운 CSS 파일에 정의하려면 Define in (New Style Sheet File)을 선택.

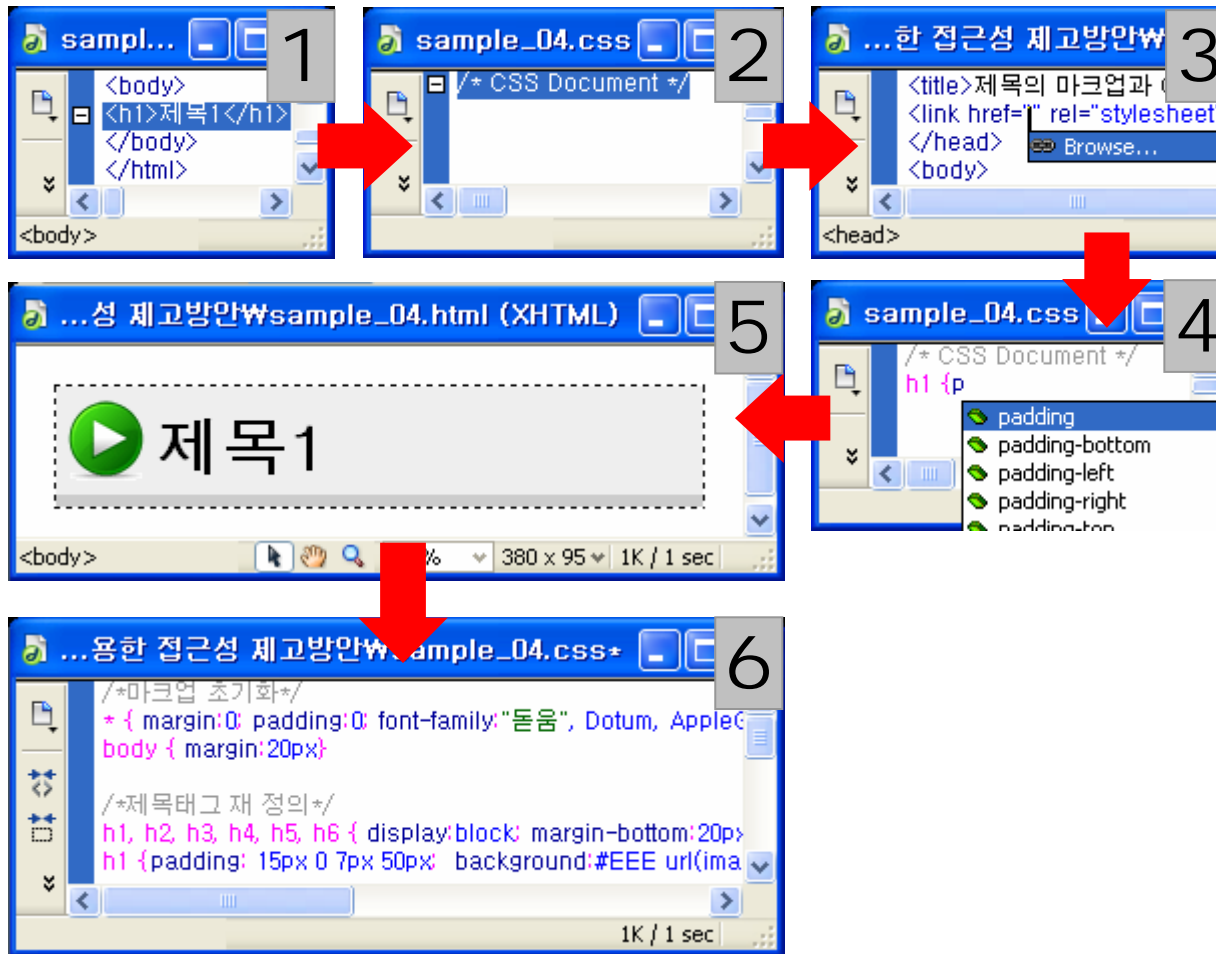
4. 새 CSS 파일 생성 및 저장.

5. 대화상자를 이용하여 <h1> 태그에 대한 속성을 정의.

6. 만족할만한 결과가 나올 때 까지 Apply 버튼을 누르면서 디자인뷰의 렌더링 변화를 확인.

4. 제목의 마크업과 CSS 디자인 (w 3.5)

나. 제목태그 디자인을 위한 CSS의 활용[계속]



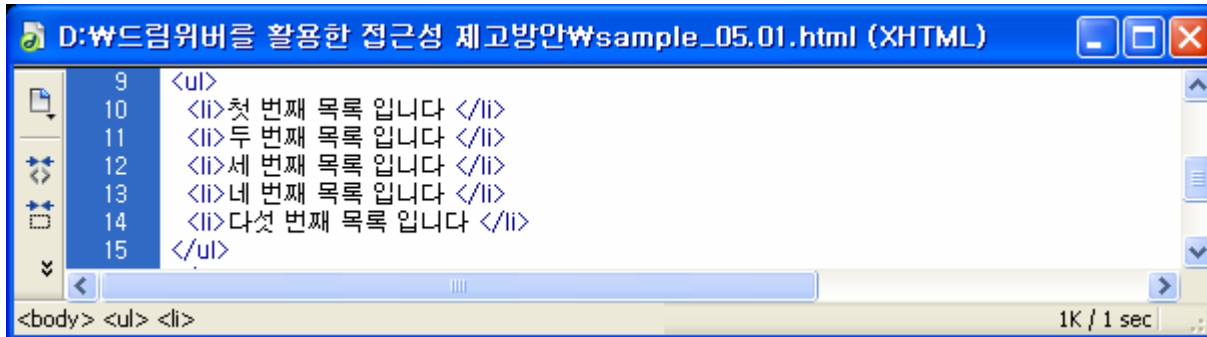
작업순서(손코딩을 이용한 CSS 작성)

1. 단축키(Ctrl+1)를 이용하여 제목 마크업.
2. CSS 파일 생성 및 저장.
3. 코드자동완성 기능(Space Bar, Ctrl+Space Bar)을 사용하여 CSS 파일을 링크.
4. 코드자동완성 기능(Space Bar, Ctrl+Space Bar)을 사용하여 CSS 코드를 작성.
5. 디자인뷰를 통하여 렌더링을 확인.
6. CSS 코드를 정정하고 완료.

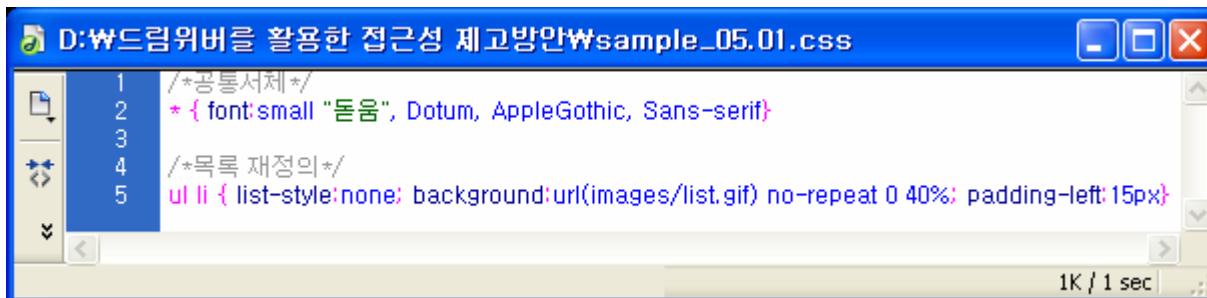
5. 목록의 마크업과 CSS 디자인 (W 3.6)

가. 배경이미지를 활용한 목록의 블릿 디자인

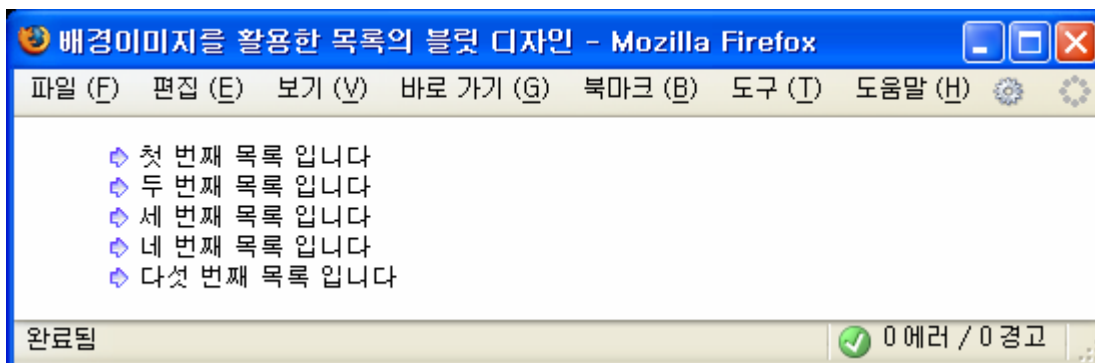
WCAG 3.6 목록(list)과 목록 아이টে을 적절히 사용한다.



```
D:\워드림위버를 활용한 접근성 제고방안\sample_05.01.html (XHTML)
9 <ul>
10 <li>첫 번째 목록입니다 </li>
11 <li>두 번째 목록입니다 </li>
12 <li>세 번째 목록입니다 </li>
13 <li>네 번째 목록입니다 </li>
14 <li>다섯 번째 목록입니다 </li>
15 </ul>
<body> <ul> <li>
```



```
D:\워드림위버를 활용한 접근성 제고방안\sample_05.01.css
1 /*공통서체*/
2 * { font:small "돋움", Dotum, AppleGothic, Sans-serif}
3
4 /*목록 재정의*/
5 ul li { list-style:none; background:url(images/list.gif) no-repeat 0 40%; padding-left:15px}
```



화면설명

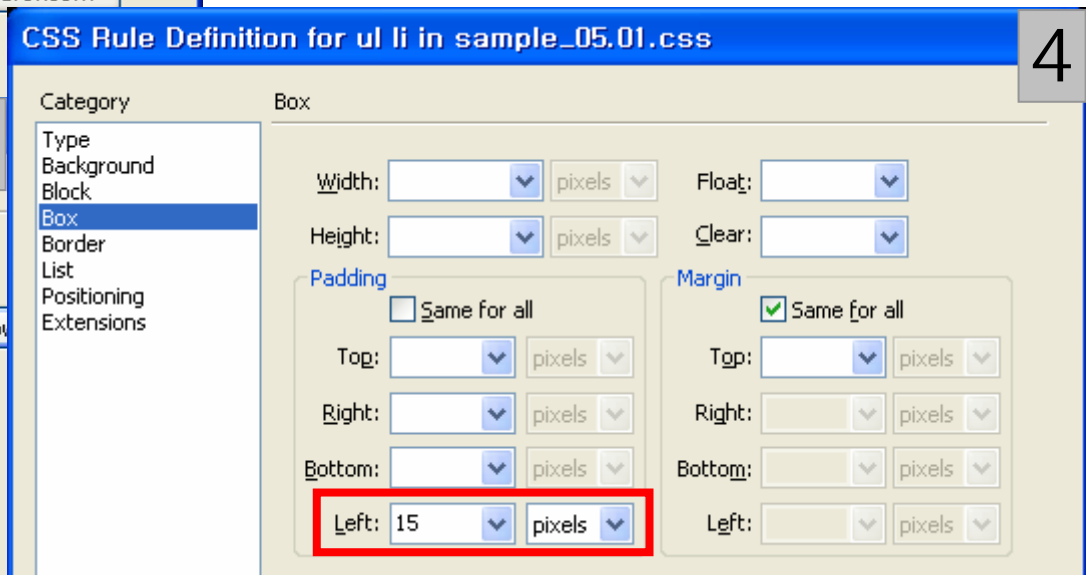
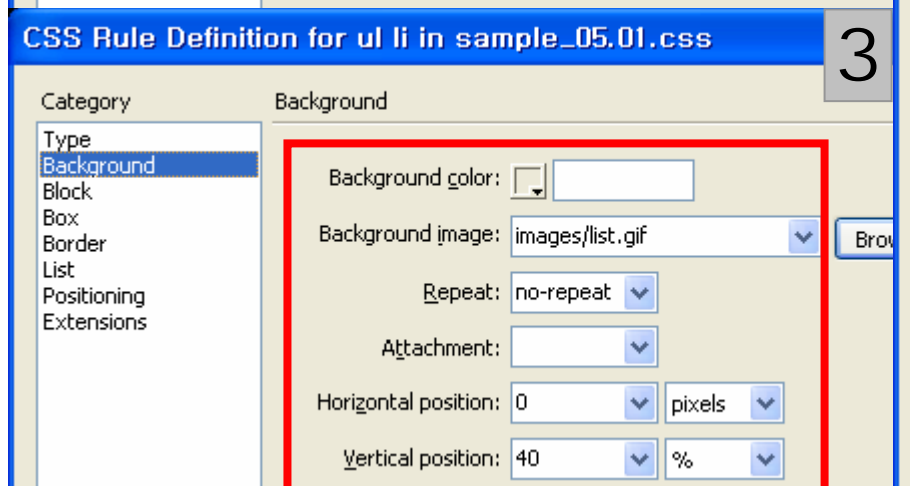
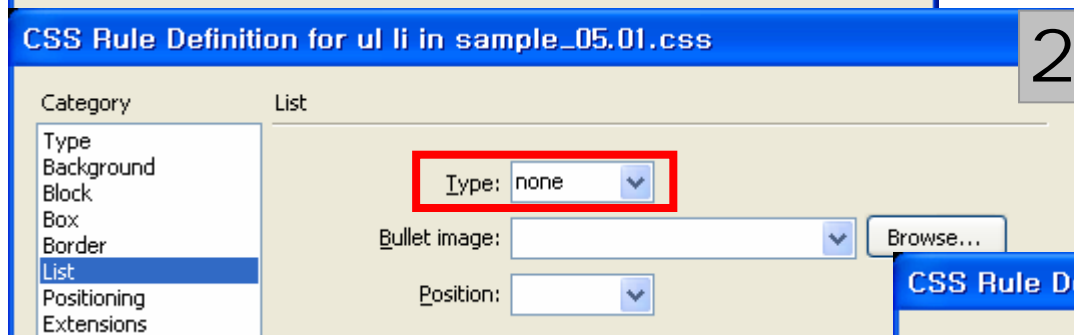
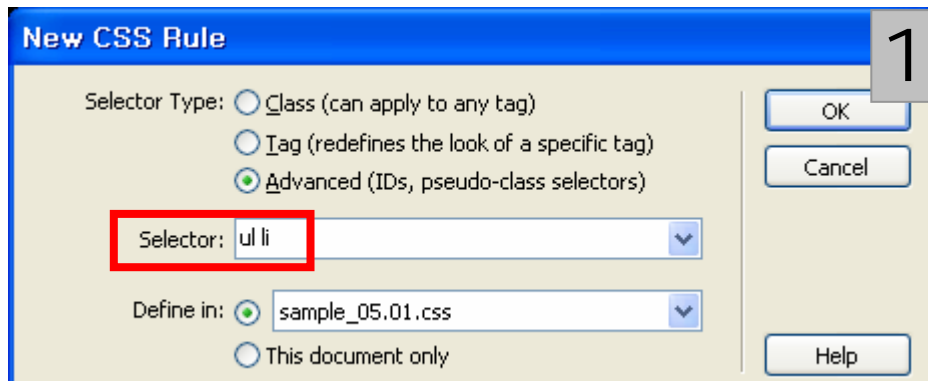
목록의 블릿 디자인을 굳이 배경으로 처리하는 이유는 블릿의 위치를 사용자 편의에 따라 자유자재로 변경할 수 있는 이점 때문이다.

한편 list-style 속성을 사용하는 경우 블릿 이미지의 위치를 자유자재로 변경하기 어려운 문제가 있으며 블릿요소 자체는 본래의 목적이 시각적 표현 이므로 CSS의 배경이미지 속성으로 대체하여도 무방하다.

예제 화면은 태그를 재정의 하여 목록의 블릿을 그래픽 이미지로 교체 하였다.

5. 목록의 마크업과 CSS 디자인 (w 3.6)

가. 배경이미지를 활용한 목록의 블릿 디자인[계속]



작업순서

1. 목록을 또는 요소로 마크업한 다음 CSS 패널의 New CSS Rule 버튼을 이용하여 태그를 재정의. 선택자 타입을 Advanced 에 체크하고 선택자는 ul li 를 입력.
2. 목록의 타입을 제거. list-style:none.
3. 배경이미지의 URL, 반복형태, 수평위치, 수직위치를 설정.
4. 왼쪽 패딩을 이용하여 배경이미지와 문자가 겹치지 않도록 설정.

6. 표의 마크업과 CSS 디자인 (W 5.1, 5.2, 5.5, KW 3.1)

가. 접근성 높은 표 데이터 마크업 하기

WCAG 5.1 데이터가 들어있는 표에는 제목 행과 제목 열(통칭하여 header)을 명시한다.

WCAG 5.2 두 개 이상의 논리적인 제목 행이나 제목 열을 갖는 데이터가 들어있는 표에서는 데이터 칸끼리 또는 제목 칸끼리 관련 짓는 마크업을 사용한다.

WCAG 5.5 표의 요약 정보를 제공하라.

KWCAG 3.1 데이터 테이블은 테이블을 구성하는 데이터 셀의 내용에 대한 정보가 충분히 전달될 수 있어야 한다

The screenshot shows a web browser window with a table and its XHTML code. The table is titled "월드컵 A조 준결승" and contains the following data:

구분	전반전	후반전
한국	1득점	2득점
미국	2득점	0득점

The XHTML code is as follows:

```
9 <table border="0" cellspacing="0" cellpadding="0" summary="
10 "월드컵 A조 준결승에서 한국이 미국을 3대2로 이김">
11 <caption>
12 월드컵 A조 준결승
13 </caption>
14 <thead>
15 <tr>
16 <th scope="col">구분</th>
17 <th scope="col">전반전</th>
18 <th scope="col">후반전</th>
19 </tr>
20 </thead>
21 <tbody>
22 <tr>
23 <th scope="row">한국</th>
24 <td>1득점</td>
25 <td>2득점</td>
26 </tr>
27 <tr>
28 <th scope="row">미국</th>
29 <td>2득점</td>
30 <td>0득점</td>
31 </tr>
32 </tbody>
</table>
```

화면설명

표의 헤더(header)와 유효범위(scope), 캡션(caption), 요약정보(summary)가 모두 활용된 접근성 높은 표의 렌더링과 코드뷰.

스크린 리더는 표의 헤더와 유효범위를 이용하여 셀과 셀을 논리적으로 연관 지어 읽어준다. 또한 요약정보를 제공하면 복잡한 표에 대한 짧은 설명을 들을 수 있으므로 접근성이 크게 개선된다.

스크린 리더는 이 표를 이렇게 읽을 것이다. "표 제목: 월드컵 A조 준결승, 표 머리글: 구분 전반전 후반전, 표 내용: 한국 전반전 1득점, 한국 후반전 2득점, 미국 전반전 2득점, 미국 후반전 0득점"

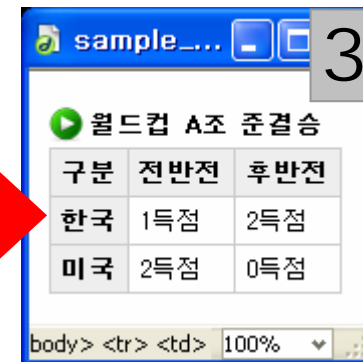
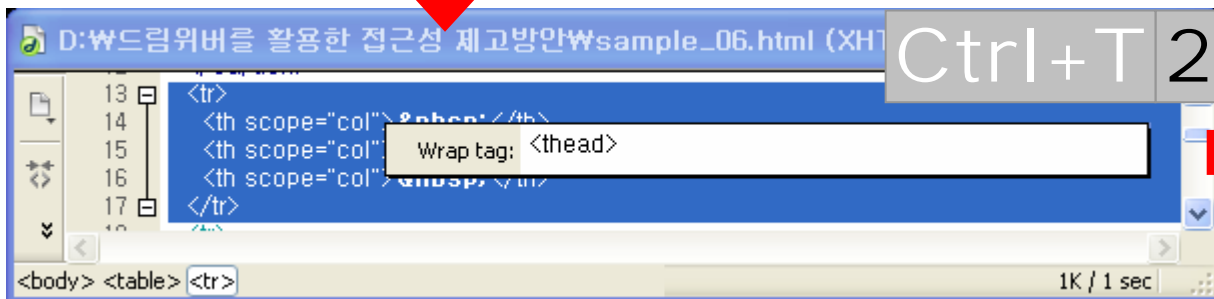
6. 표의 마크업과 CSS 디자인 (W 5.1, 5.2, 5.5, KW 3.1)

가. 접근성 높은 표 데이터 마크업 하기 [계속]



작업순서

1. 표 삽입 (Ctrl+Alt+T). 열과 행의 개수를 입력하고 표의 헤더영역, 캡션, 요약정보를 입력.
2. 디자인뷰에서 <thead>, <tbody>를 지정해주는 옵션이 따로 없으므로 코드뷰에서 빠른 태그편집기를 사용함. 셀의 머리 행 <tr> 영역을 선택하고 빠른 태그 편집 (Ctrl+T) 기능을 이용하여 <thead> 태그로 감싸줌. 셀의 본문 행 <tr> 영역을 선택하여 <tbody> 태그로 감싸줌.
3. 표의 내용을 입력하여 표를 완성.



6. 표의 마크업과 CSS 디자인 (W 5.1, 5.2, 5.5, KW 3.1)

나. CSS를 이용한 표 디자인

```
9 <table border="0" cellspacing="0" cellpadding="0" summary="
10 "월드컵 A조 준결승에서 한국이 미국을 3대2로 이김">
11 <caption>
12 월드컵 A조 준결승
13 </caption>
14 <thead>
15 <tr>
16 <th scope="col">구분</th>
17 <th scope="col">전반전</th>
18 <th scope="col">후반전</th>
19 </tr>
20 </thead>
21 <tbody>
22 <tr>
23 <th scope="row">한국</th>
24 <td>1득점</td>
25 <td>2득점</td>
```

```
5 caption {
6 background: url(images/h5.gif) no-repeat left center;
7 padding: 5px 5px 5px 20px;
8 text-align: left;
9 font-weight: bold;
10 }
11 table {
12 border-top: 1px solid #CCC;
13 border-left: 1px solid #CCC;
14 }
15 th, td {
16 border-right: 1px solid #CCC;
17 border-bottom: 1px solid #CCC;
18 padding: 5px;
19 }
20 th {
21 background: #EEE;
22 }
```

구분	전반전	후반전
한국	1득점	2득점
미국	2득점	0득점

화면설명

HTML 문서에는 디자인 표현 없이 표의 마크업 정보만 코딩 되어 있다.

한편 CSS 문서에는 표의 캡션, 표, 헤더 셀, 일반 셀에 대하여 태그 재정의 되어 있기 때문에 디자인 된 표의 형태를 볼 수 있게 된다.

작업순서

1. 표 제목 아이콘을 <caption>의 배경으로 처리하여 넣기. 그 밖에 캡션의 패딩, 문자정렬, 서체굵기를 지정.
2. <table>의 보더 Top, Left 에 실선을 적용. 예) 1px solid #CCC.
3. <th>, <td> 의 보더 Right, Bottom 에 실선을 적용하고 셀의 패딩을 지정. 예) 1px solid #CCC.
4. <th> 의 배경색을 별도로 지정.

7. 폼의 마크업과 CSS 디자인 (W 10.2, 12.3, 12.4, KW 3.3)

가. 폼 컨트롤 레이블링 하기

WCAG 10.2 웹 표시 장치가 레이블(label)과 폼(form) 컨트롤 사이의 명시적으로 지정된 연관을 명료한 방법으로 인식 가능하게 해 줄 때까지는 레이블을 적절한 위치에 놓아야 한다.

WCAG 12.4 컨트롤과 그 컨트롤의 레이블을 명시적으로 짝지어야 한다.

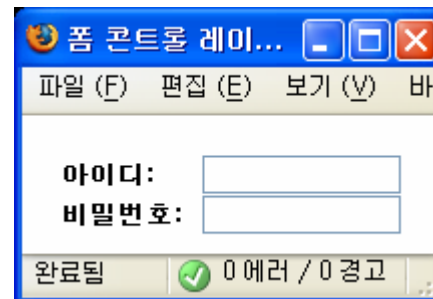
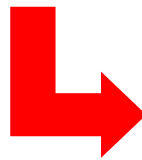
KWCAG 3.3 온라인 서식을 포함하는 콘텐츠는 서식 작성에 필요한 정보, 서식 구성 요소, 필요한 기능, 작성 후 제출 과정 등 서식과 관련한 모든 정보를 제공해야 한다.



화면설명

폼 컨트롤과 레이블이 명시적으로 짝 맞춘 되어 있다. 사용자가 레이블을 클릭하는 순간 폼 컨트롤을 클릭한 것과 같은 효과가 발생하기 때문에 시각장애인 뿐만 아니라 일반인들의 사용성을 증가시켜 주기도 한다.

정의목록 태그는 정의목록 뿐만 아니라 그와 유사한 형태의 콘텐츠를 마크업 하는 것을 허용하고 있으므로 이렇게 폼 컨트롤과 레이블을 정의목록으로 마크업 하는 것은 적절한 방법이다.



7. 폼의 마크업과 CSS 디자인 (W 10.2, 12.3, 12.4, KW 3.3)

가. 폼 컨트롤 레이블링 하기 [계속]

1. Input Tag Accessibility Attributes dialog box for the ID field. The 'Label' is '아이디:' and the 'Style' is 'Attach label tag using 'for' attribute'. The 'Position' is 'Before form item'.

2. Dreamweaver warning dialog: 'Add form tag?'. The 'Yes' button is highlighted.

3. Input Tag Accessibility Attributes dialog box for the password field. The 'Label' is '비밀번호:' and the 'Style' is 'Attach label tag using 'for' attribute'. The 'Position' is 'Before form item'.

4. Code view showing the HTML structure: `<dt> 아이디: <input type="text" value="" /> <dd> 비밀번호: <input type="password" value="" />`. The `<dt>` and `<dd>` tags are highlighted.

5. Code view showing the HTML structure: `<body> <form#login> <dl> <dt> 아이디: <input type="text" value="" /> <dd> 비밀번호: <input type="password" value="" /> </dl>`. The `<dl>` tag is highlighted.

작업순서(폼 컨트롤 마크업)

1. Input Tag Accessibility Attributes 대화 상자를 이용하여 아이디 항목 입력을 위한 `<input>` 태그와 `<label>` 태그를 동시에 마크업 한다.
2. 폼 태그 삽입 대화상자를 이용하여 폼을 마크업 한다.
3. 폼 요소 내부에 비밀번호 항목을 위한 `<input>` 태그와 `<label>` 태그를 마크업 한다.
4. 태그 선택기에서 `<label>` 태그를 선택한 다음 삽입메뉴의 Text>dt, dd 를 이용하여 목록 요소를 마크업 한다.
5. 목록전체를 감싸는 `<dl>` 태그는 코드뷰에서 직접 마크업 한다.

7. 폼의 마크업과 CSS 디자인 (W 10.2, 12.3, 12.4, KW 3.3)

가. 폼 컨트롤 레이블링 하기 [계속]

```

D:\워드럼위버를 활용한 접근성 제고방안\sample_07.css
1 /*공통 서체*/
2 * { font:small "돋움", Dotum, AppleGothic, Sans-serif; margin:0; padding:0; }
3 body { margin:20px; }
4
5 /*라벨 디자인*/
6 #login dl { width:170px; float:left; clear:both; display:block; }
7 #login dt, #login dd { display:block; }
8 #login dt { width:70px; height:17px; padding-top:3px; float:left; clear:left; }
9 #login dt label { font-weight:bold; }
10 #login dd { width:100px; height:20px; float:right; clear:right; }
11 #login dd input { width:95px; }
    
```

작업순서(폼 컨트롤 CSS 디자인)

1. 공통 서체 및 마진, 패딩 초기화.
2. body 마진 설정.
3. dl 에 대한 재정의 코드 작성.
4. dt, dd 에 대한 재정의 코드 작성.
5. dt 에 대한 재정의 코드 작성.
6. label 에 대한 재정의 코드 작성.
7. dd 에 대한 재정의 코드 작성.
8. input 에 대한 재정의 코드 작성.

1 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced

Selector: *

Define in: sample_07.css This document only

2 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced

Tag: body

Define in: sample_07.css This document only

3 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced

Selector: #login dl

Define in: sample_07.css This document only

4 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced (IDs, pseudo-class selectors)

Selector: #login dt, #login dd

Define in: sample_07.css This document only

OK

Cancel

Help

5 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced

Selector: #login dt

Define in: sample_07.css This document only

6 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced (IDs, pseudo-class selectors)

Selector: #login dt label

Define in: sample_07.css This document only

7 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced

Selector: #login dd

Define in: sample_07.css This document only

8 Selector Type: Class (can apply to any tag) Tag (redefines the look of a specific tag) Advanced (IDs, pseudo-class selectors)

Selector: #login dd input

Define in: sample_07.css This document only

OK

Cancel

Help

7. 폼의 마크업과 CSS 디자인 (W 10.2, 12.3, 12.4, KW 3.3)

나. 폼 컨트롤 요소를 논리적으로 그룹 짓기

WCAG 12.3 큰 덩어리로 이루어진 정보는 자연스럽게 그리고 적절하게, 더 쉽게 다룰 수 있는 몇 개의 그룹으로 나눈다.

폼 컨트롤 그룹짓기 - Mozilla Firefox

파일 (F) 편집 (E) 보기 (V) 바로 가기 (G) 북마크 (B) 도구 (T) 도움말 (H)

필수입력정보

이름:

이메일:

아이디:

비밀번호:

비밀번호확인:

선택입력정보

직업:

취미: 컴퓨터/인터넷 레포츠/운동

뉴스레터수신: 수신동의 수신거부

완료됨

폼 컨트롤 그룹짓기 - Mozilla Firefox

파일 (F) 편집 (E) 보기 (V) 바로 가기 (G) 북마크 (B) 도구 (T) 도움말 (H)

필수입력정보

이름:

이메일:

아이디:

비밀번호:

비밀번호확인:

선택입력정보

직업:

취미: 컴퓨터/인터넷 레포츠/운동

뉴스레터수신: 수신동의 수신거부

완료됨 0 에러 / 0 경고

화면설명

예제 화면은 웹사이트 회원가입 시 다루어질 수 있는 일반적인 폼 요소의 묶음으로 구성되어 있다.

폼 요소는 두 개의 덩어리로 나뉘어 지는데 하나는 필수입력정보이며 다른 하나는 선택입력정보 이다.

폼들은 단지 시각적으로만 구분된 것이 아니라 <fieldset> 과 <legend> 요소를 사용하여 논리적으로 분할 하였다.

첫 번째 예제는 CSS가 적용된 화면이며 두 번째 예제는 CSS를 제거한 화면이다.

이렇게 <fieldset> 과 <legend> 태그를 사용하면 스크린 리더의 음성향해를 도울 수 있는데 이 두 가지 태그는 함께 사용함으로써 접근성을 극대화 시킬 수 있다.

7. 품의 마크업과 CSS 디자인 (W 10.2, 12.3, 12.4, KW 3.3)

나. 품 컨트롤 요소를 논리적으로 그룹 짓기 [계속]

```

D:\워드럼위버를 활용한 접근성 제고방안\sample_07.01.css

/*공통 서체*/
* { font:small "돋움", Dotum, AppleGothic, Sans-serif; margin:0; padding:0; }
body { margin:20px; }

/*라벨 디자인*/
#join dl { width:400px; float:left; clear:both; display:block; margin:10px 0; }
#join dt, #join dd { display:block; }
#join dt { width:100px; height:19px; padding-top:3px; float:left; clear:left; text-align:right; }
#join dd { width:295px; height:22px; float:right; clear:right; padding-left:5px; }
#join fieldset { float:left; clear:both; margin-bottom:15px; border:none; border-top:3px solid #DDD; }
#join legend { font-size:medium; font-weight:bold; background:url(images/h5.gif) no-repeat left center; padding:0 10px 0 20px; }
    
```

작업순서(품 컨트롤 그룹 CSS 디자인)

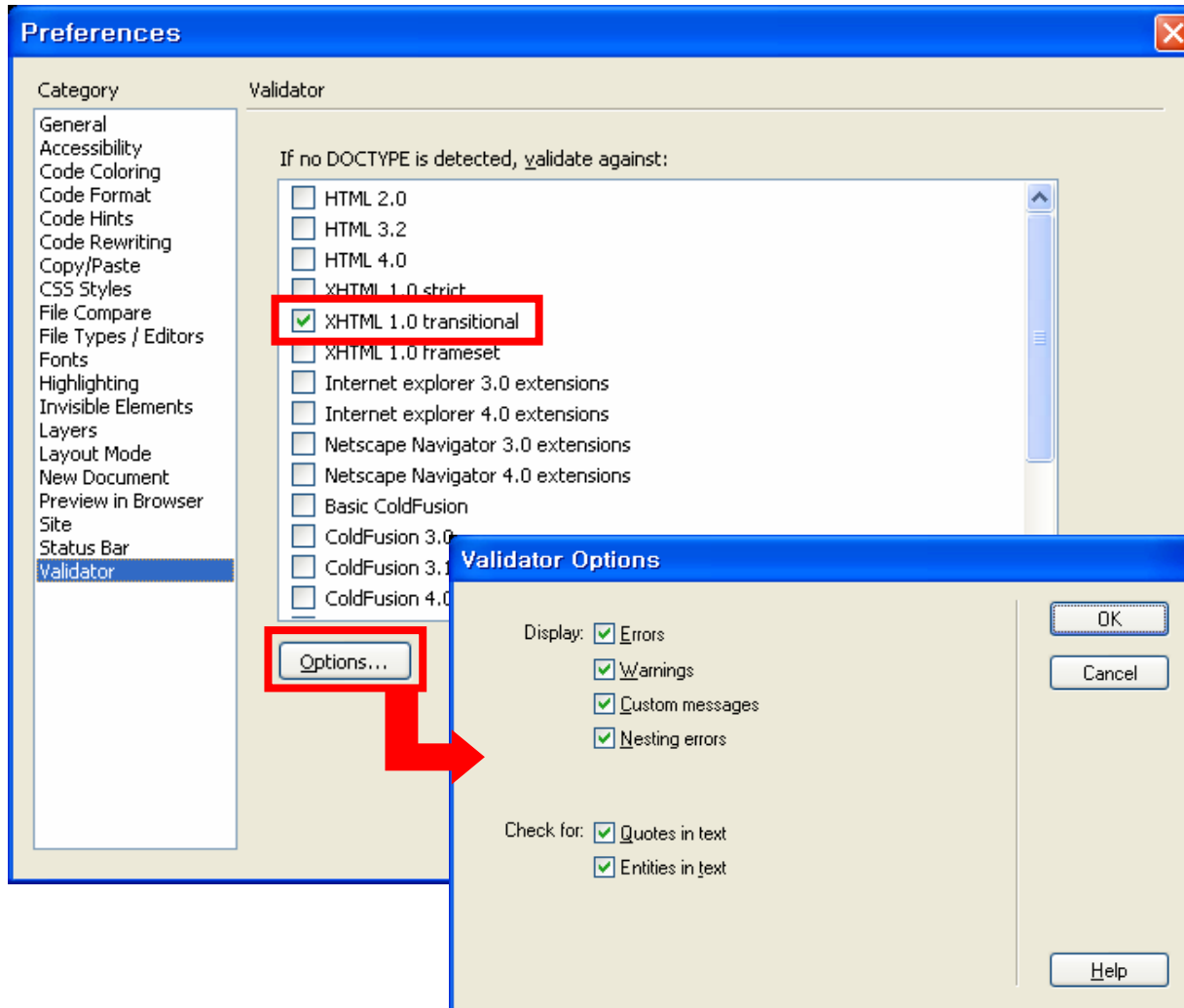
1. 공통 서체 및 마진, 패딩 초기화.
2. body 마진 설정.
3. dl 에 대한 재정의의 코드 작성.
4. dt, dd 에 대한 재정의의 코드 작성.
5. dt 에 대한 재정의의 코드 작성.
6. dd 에 대한 재정의의 코드 작성.
7. fieldset 에 대한 재정의의 코드 작성.
8. legend 에 대한 재정의의 코드 작성.

1	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	2	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	3	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	4	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>
	Selector: *		Tag: body		Selector: #join dl		Selector: #join dt, #join dd	
	Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only		Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only		Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only		Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only	
5	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	6	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	7	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	8	Selector Type: <input type="radio"/> Class (can apply to any tag) <input type="radio"/> Tag (redefines the look of a specific tag) <input checked="" type="radio"/> Advanced (IDs, pseudo-class selectors)	<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>
	Selector: #join dt		Selector: #join dd		Selector: #join fieldset		Selector: #join legend	
	Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only		Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only		Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only		Define in: <input checked="" type="radio"/> sample_07.01.css <input type="radio"/> This document only	

8. 마크업에 대한 유효성 검사(w 11.1, 11.2)

가. 마크업 유효성 검사를 위한 드림위버 환경설정

WCAG 11.1 하고자 하는 일에 적합한 W3C 기술이 있다면 되도록이면 최신판을 찾아 채택한다.
WCAG 11.2 낡은 W3C 기술(deprecated features of W3C technologies)을 사용하지 말라.



화면설명

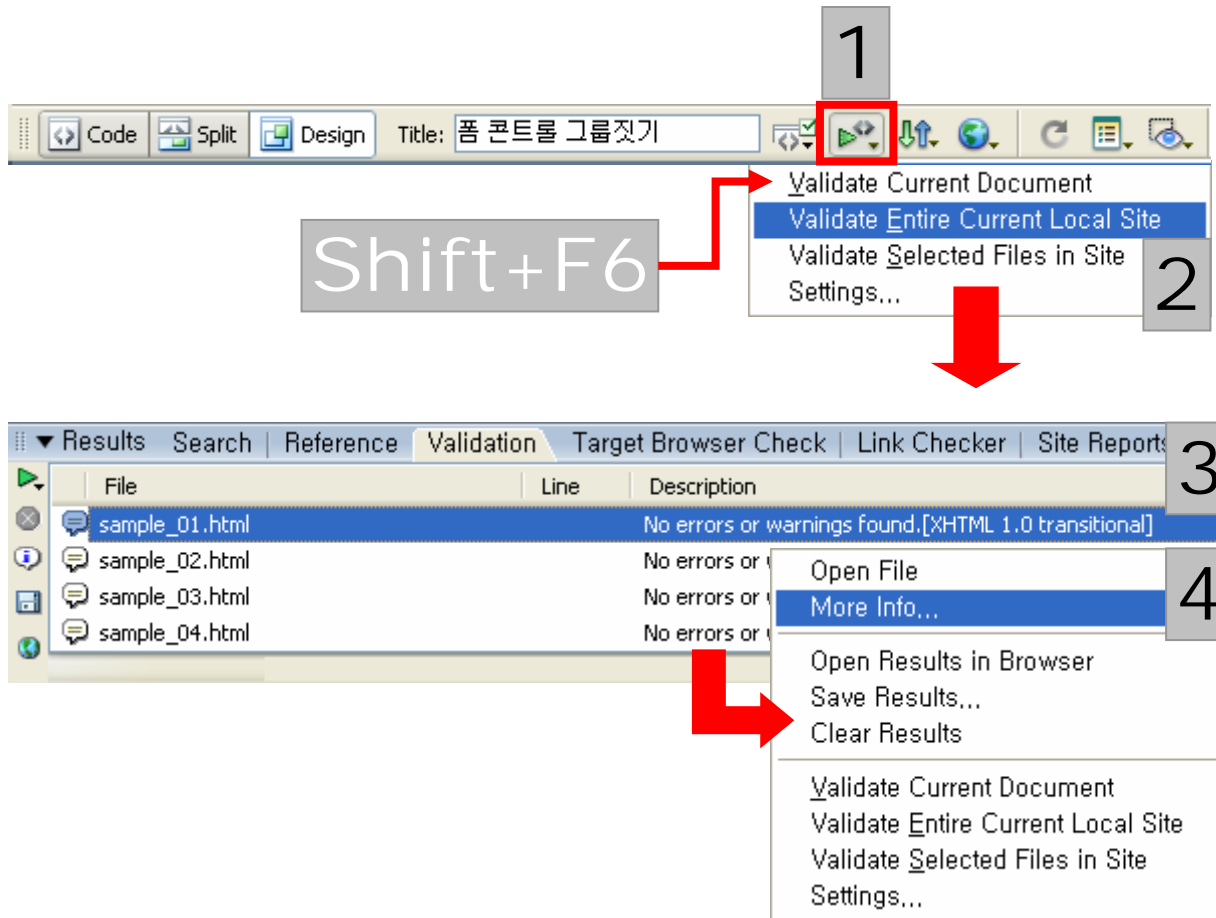
드림위버는 개별적인 문서의 DTD를 인식하여 해당 문서가 DTD 문법에 맞는지 검사해주는 기능을 포함하고 있다. 그러나 흔히 인클루드 되는 문서와 같이 <head> 영역이 없거나 DTD를 포함하지 않는 문서가 발견되면 드림위버의 Validator 환경 설정을 기준으로 하여 유효성 검사를 진행한다.

드림위버의 Validator 는 신뢰할 만한 수준으로 문법오류를 확인해 주지만 유효성 검증을 위한 최후의 수단으로 사용하는 것보다는 실시간으로 문법오류를 체크할 수 있는 방편으로 활용하기에 적절하다.

사이트 전체의 Validation 검증 도구로는 검증결과를 상세하게 Report 해주는 KADO-WAH가 있고 개별 페이지의 검증 도구로는 W3C의 Validation Service 를 이용하는 것이 가장 정확하다.

8. 마크업에 대한 유효성 검사(w 11.1, 11.2)

나. 드림위버를 이용한 마크업 유효성 검사



화면설명

드림위버의 삽입막대에 위치한 Validate markup 기능을 이용하면 현재문서, 사이트 전체, 선택된 파일 및 폴더 단위로 유효성 검사를 진행할 수 있다.

문서 내에 DTD가 있는 경우 해당 DTD에 근거하여 검사하며 DTD가 없는 문서의 경우 Validator 환경설정에서 지정한 DTD를 기준으로 진행한다.

작업순서

1. 삽입메뉴에서 Validate markup 메뉴를 선택한다.
2. 유효성 검사 대상을 선택한다.
3. Results 패널이 열리면서 Validation 검사결과를 보여준다.
4. 오류가 발견되면 더블클릭 하여 파일을 열거나 또는 컨텍스트 메뉴(마우스 오른쪽 메뉴)를 활성화 하여 오류에 대한 더 자세한 정보를 확인할 수 있다.

※현재 문서의 유효성 검사만 진행하려면 단축키 Shift+F6 을 사용한다.

9. 마크업과 CSS에 대한 브라우저 호환성 검사(KW 4)

가. 브라우저 호환성 검사를 위한 환경설정 및 검사

KWCAG 4 구성한 콘텐츠는 웹 브라우저의 종류, 버전 등에 관계없이 사용될 수 있어야 한다.

1

2

3

4

5

6

화면설명

삽입막대에 위치한 Check Browser Support 기능을 이용하면 마크업과 CSS에 대한 다양한 브라우저의 지원여부를 확인할 수 있다.

다만 현존하는 최신의 브라우저에 대한 지원확인은 불가능한 단점이 존재. FF 1.0, IE 6.0, OP 8.0, SF 2.0 버전(오래된 버전)에 대한 지원여부만 확인할 수 있다.

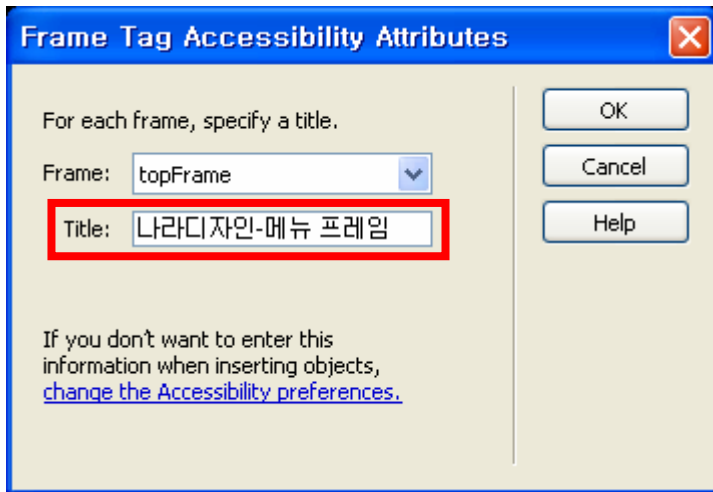
작업순서

1. 삽입막대의 Check Browser Support 메뉴 선택.
2. Settings... 선택.
3. 검사대상 브라우저 설정.
4. Show All Errors 클릭.
5. 검사결과 확인. 지원하지 않는 항목을 더블클릭. (컨텍스트 메뉴를 이용하여 더 자세한 정보도 확인 가능)
6. 브라우저에서 지원하지 않는 마크업을 찾아내어 정정하거나 삭제.

11. 프레임을 사용하는 경우(W 12.1, KW 2.2)

가. '어쩔 수 없이' 프레임을 사용한다면 프레임 요소에 "title" 속성을 사용

WCAG 12.1 프레임을 구분하고 프레임 사이의 탐색을 쉽게 하기 위해서 모든 프레임에는 제목(이름)을 붙여야 한다.
KWACG 2.2 콘텐츠를 구성하는 프레임의 수는 최소한으로 하며, 프레임을 사용할 경우에는 프레임별로 제목을 붙여야 한다.



화면설명

프레임은 최소화 하는 것이 좋고 프레임을 사용하지 않는 것이 가장 좋다. 스크린 리더 사용자들은 페이지보다 먼저 프레임을 탐색하게 되는데 <frame> 요소에 title 속성이 없는 경우 매번 해당 프레임 본문을 모두 탐색해야만 한다. 만약 어쩔 수 없이 프레임을 사용한다면 title 속성을 넣어 해당 프레임의 주제가 무엇인지 명료하게 식별할 수 있도록 도와주어야 한다.



드림위버는 프레임을 삽입할 때 예제와 같이 프레임의 title 속성을 입력할 수 있는 대화상자를 보여준다.(이는 접근성 속성 환경설정에 따른다)

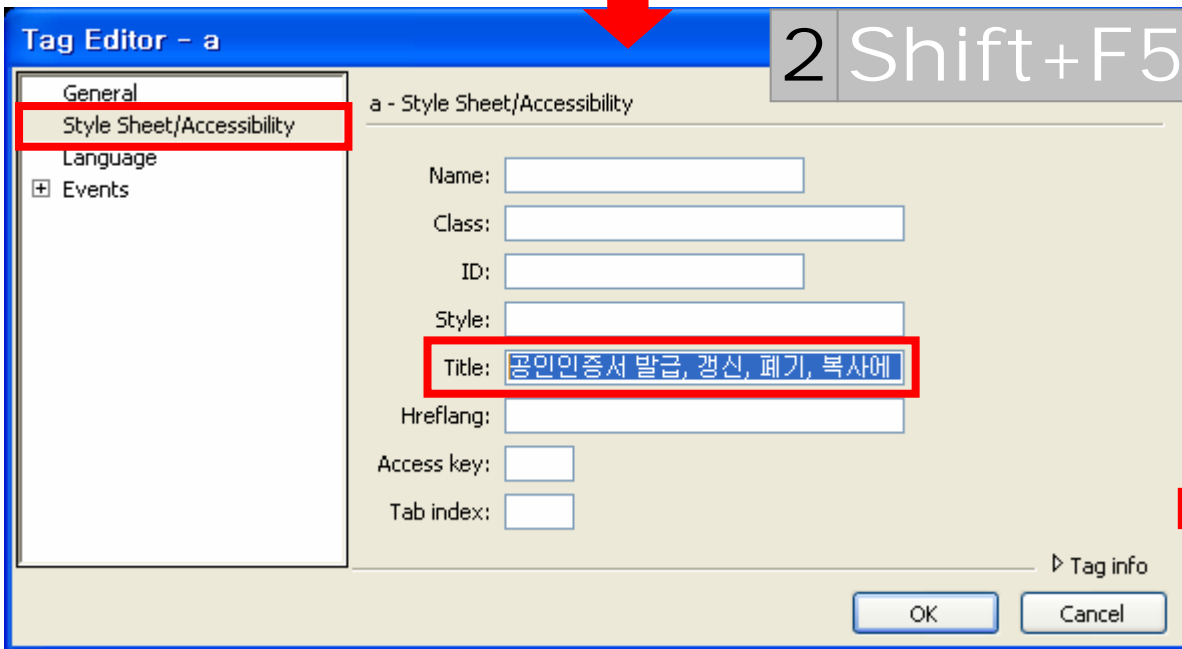
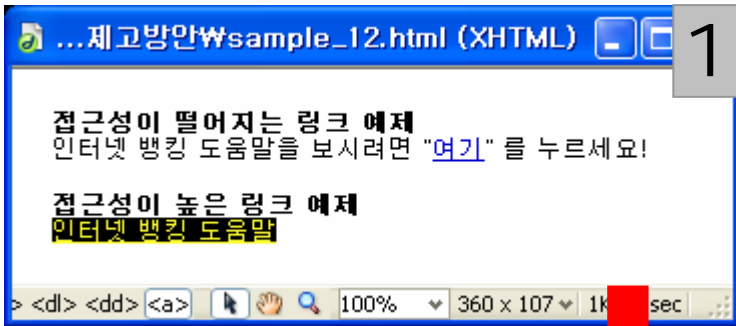
작업순서

1. 프레임 삽입 후 접근성 속성 대화상자에서 <frame> 태그에 대한 title 속성을 입력.

12. 모호한 링크를 보다 선명하게 만들기 (W 13.1)

가. 태그 편집기를 이용하여 하이퍼링크에 "title" 속성 넣기

WCAG 13.1 각 링크의 목표 위치(target)를 명확히 한다.

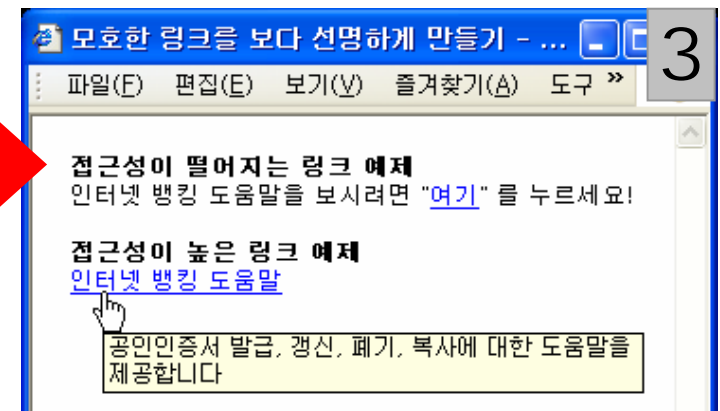


화면설명

링크된 문장 안쪽에 보다 직관적인 단어와 문장을 사용한다면 Tab 키만으로 향해 하더라도 링크의 목적지를 금새 알아낼 수 있다. 만약 그래도 설명이 모호한 경우라면 <a> 태그에 대하여 title 속성을 추가해 준다.

작업순서

1. 태그 선택기에서 <a> 태그를 선택.
 2. 태그 편집기(Shift+F5)를 사용하여 title 속성을 입력.
- ※ 또는 빠른 태그편집(Ctrl+T) 기능을 사용하여 title 속성을 추가.

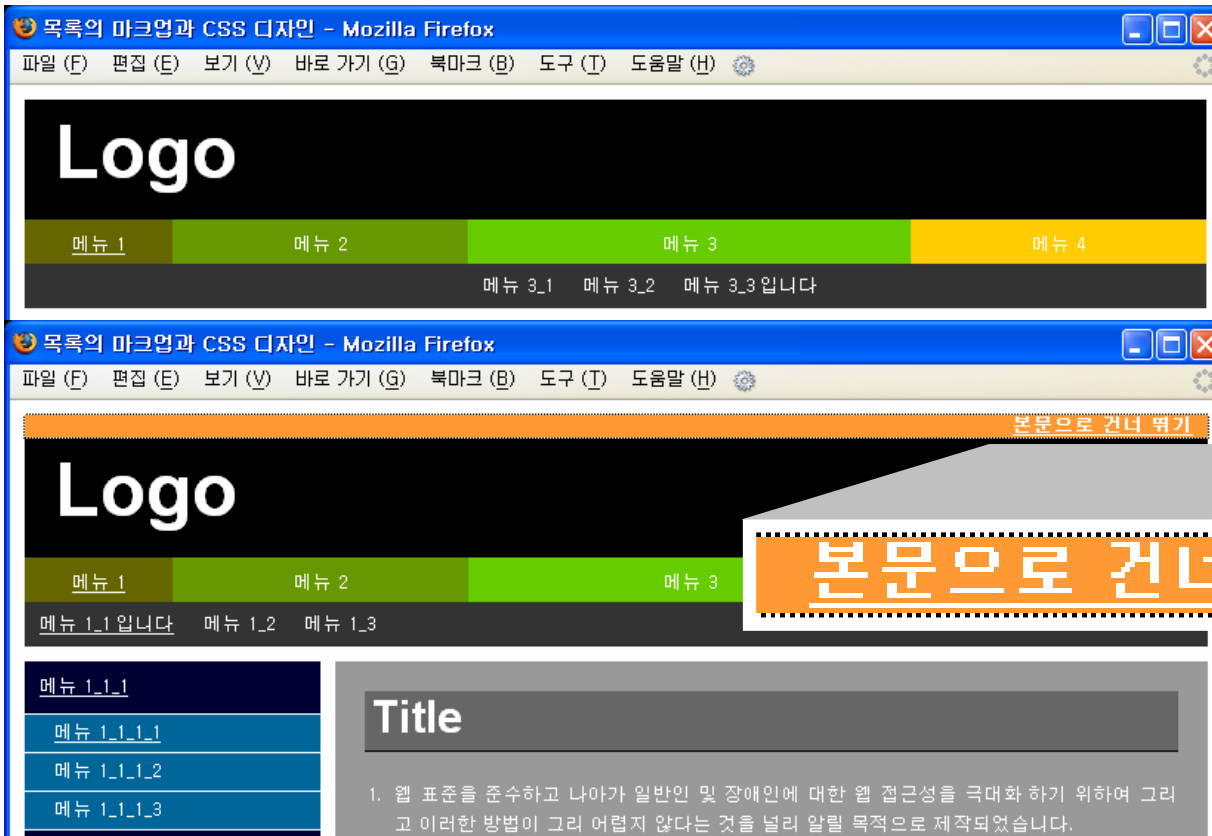


13. 메뉴 건너뛰기 기능의 지원(W 13.6, KW 2.5)

가. 시각장애인을 위한 메뉴 건너뛰기 링크

WCAG 13.6 관련된 링크들은 한데 묶어 두고, 웹 표시 장치가 그 모뎀을 식별할 수 있도록 표시한다. 그리고, 웹 표시 장치가 사용자들에게 그런 기능을 제공하기 전까지는 모뎀을 건너 뛸 수 있는 방법도 제시해 주어야 한다.

KWCAG 2.5 웹 콘텐츠는 반복적인 네비게이션 링크를 뛰어넘어 페이지의 핵심부분으로 직접 이동할 수 있도록 구성하여야 한다.



화면설명

예제 화면은 키보드를 사용할 때에만 “본문으로 건너뛰기” 메뉴가 펼쳐진다.

메뉴 건너뛰기 기능이 일반인들에게 시각적으로 불필요한 요소가 될 수 있는 점에 착안하여 키보드로 조작할 때에만 나타나도록 한 것으로 CSS의 가상선택자 a:focus 속성을 활용한 것이다.

작업순서

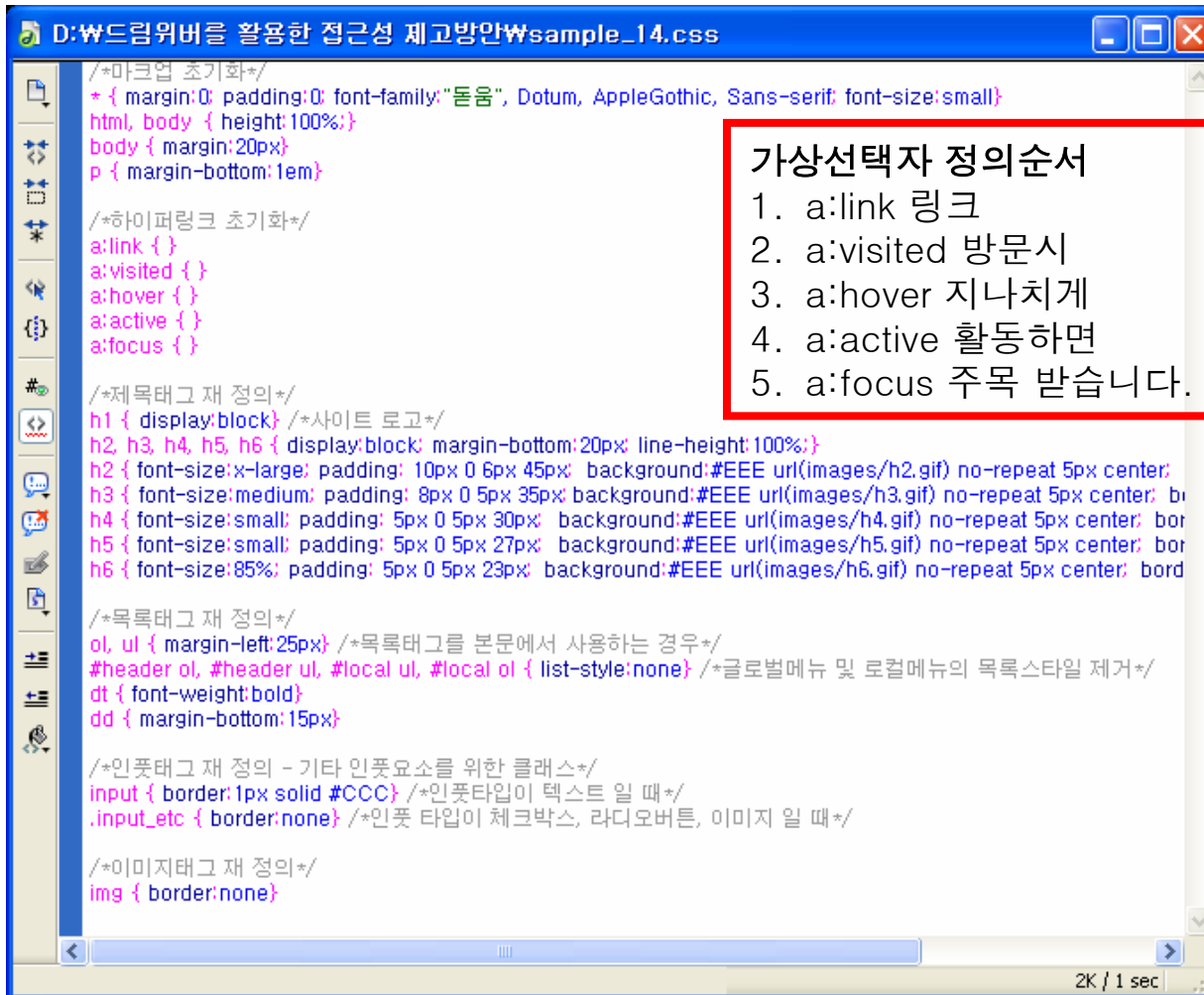
1. a:link 일 때 height:0px 으로 설정.
 2. a:focus, a:active 일 때 height:auto 으로 설정.
- ※active 는 IE의 focus 상태를 위한 설정.

http://naradesign.net/open_content/reference/navigation.html

14. 페이지 디자인을 위한 CSS 기초(W 3.3, 14.3, KW 3.2)

가. 페이지 레이아웃을 위한 CSS 기초

WCAG 3.3 내용물의 자리 배치와 표현 방식을 조절하기 위해서는 스타일 시트를 사용하라.
WCAG 14.3 여러 페이지에 걸쳐 일관성 있게 쓸 표현 방법 스타일을 개발하라.
KWACG 3.2 콘텐츠의 모양이나 배치는 논리적으로 이해하기 쉽게 구성하여야 한다.



화면설명

페이지 레이아웃을 위하여 CSS 를 사용하는 방법 가운데 가장 공을 들여 작성해야 하는 부분은 모든 페이지에 걸쳐 공통적으로 적용되는 기본 태그에 대한 재정의의 스타일 이다.

예제는 *, html, body, p, a, h1~h6, ol, ul, dl, input, img 등의 주요 마크업 태그에 대하여 사전 재정의의 해 둌으로써 모든 페이지에 걸쳐 일관된 형식의 디자인을 제공한다.

특히, “*” 공통선택자에 적용하는 속성은 하위 요소로 상속되는 것이 아니라 모든 태그를 재정의 한다는 점에 유의할 것.

작업순서

1. 웹사이트 구축 초기단계에서 그래픽 시안이 없이도 작업된다.

14. 페이지 디자인을 위한 CSS 기초(W 3.3, 14.3, KW 3.2)

나. 다 기종 브라우저간 렌더링 호환성 유지를 위한 CSS 팁



브라우저 렌더링 호환성 유지를 위한 팁

1. CSS 화면을 가장 표준에 맞게 렌더링 해주는 브라우저를 테스트 브라우저로 선택. 가장 대중적인 브라우저 가운데 현재 OP 9.0 버전이 Acid2 Test 를 통과.
2. “*” 공통선택자의 마진과 패딩을 초기화 한다. 브라우저마다 다른 마진과 패딩 값을 강제로 0px으로 초기화 한다.
3. html, body 태그의 높이를 100%으로 설정한다. 어떤 브라우저는 이 값을 설정해 주지 않으면 문제가 생긴다.
4. float 속성을 정확히 이해한다. float 속성은 float 된 요소끼리만 어울린다. float을 끝내려면 부모박스에 overflow:auto 속성을 준다.
5. 의도하지 않았던 여백이 생기면 inline 속성을 display:block으로 변경하고 상황이 허락된다면 overflow 속성을 hidden으로 설정해 본다.
6. 문자가 들어가는 상자의 높이 값은 가능한 설정하지 않는 것이 좋고 콘텐츠의 양에 따라 가변적으로 늘어나도록 하는 것이 좋다.

15. 출력을 위한 CSS 문서 만들기

가. 콘텐츠 영역만 출력해주는 CSS 만들기 팁

Monitor

Print

Attach External Style Sheet

File/URL: print.css [Browse...]

Add as: Link Import

Media: print

all of media types.

Dreamweaver

OK, Preview, Cancel, Help

http://engit.com/open_content/company_overview/management_Idea_and_vision.p... 2006-07-20

화면설명

출력을 위한 별도의 웹 문서를 생성하여 제공하는 것은 바람직하지 않다. 인쇄된 결과물에 표시된 URL 정보만을 참조하여 해당 웹 문서에 접근하고자 하는 경우 사용자는 본래의 페이지로 접속되지 않고 인쇄를 위한 페이지로 접속되기 때문이다.

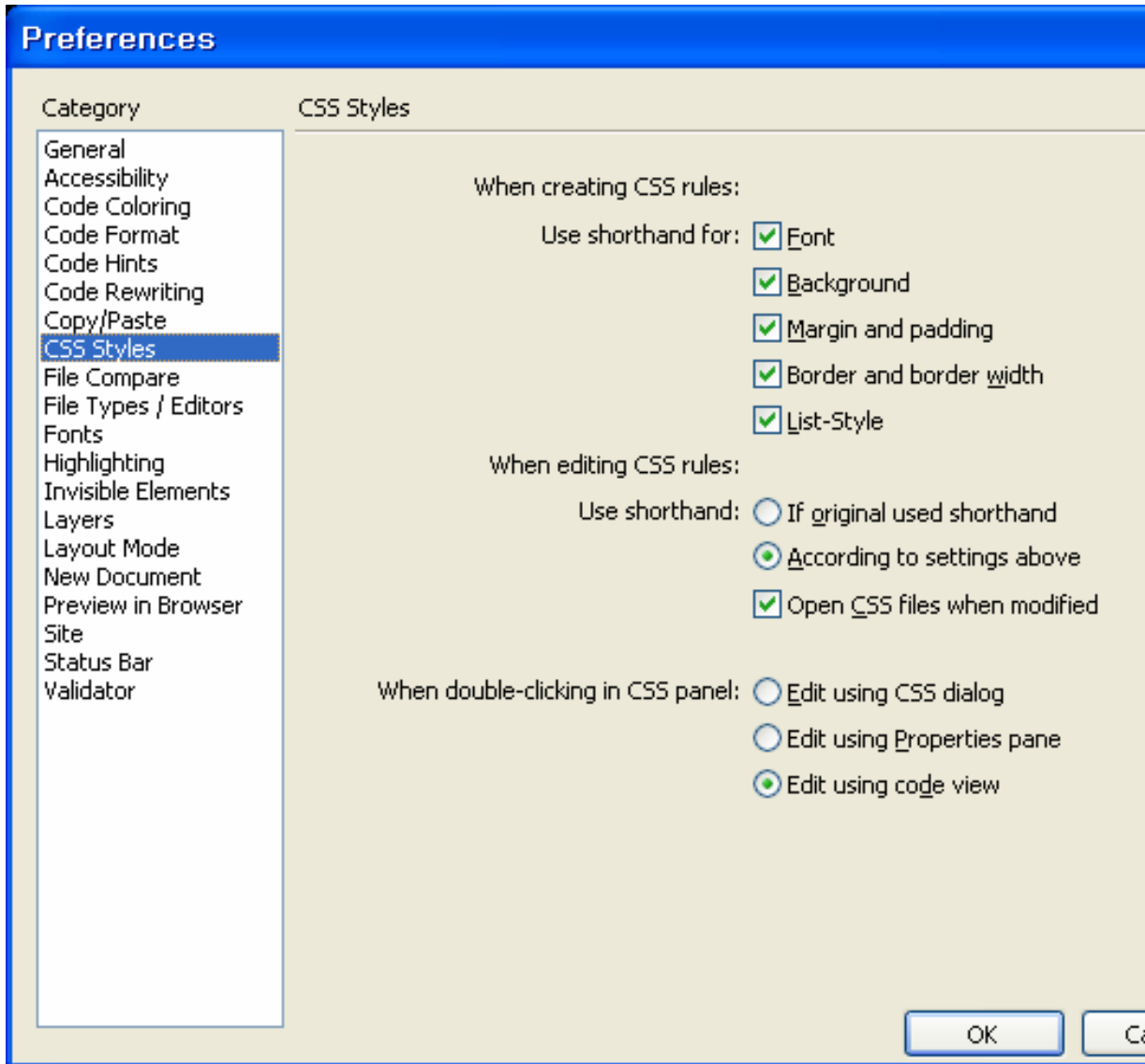
예제는 출력용 웹 문서를 따로 만들지 않고도 CSS를 이용하여 콘텐츠 영역만 출력할 수 있는 페이지의 예이다.

작업순서

1. 프린트 화면을 정의하기 위한 print.css 파일을 생성한다.
2. print.css 파일에는 헤더영역, 글로벌 메뉴, 로컬메뉴, 저작권 영역의 display 속성을 none 으로 변경한다.
3. XHTML 문서 헤더에서 print.css 파일을 link 하거나 import 할 때 media 타입을 print 으로 설정한다.
4. 인쇄 미리보기를 통하여 콘텐츠 영역만 출력되는 것을 확인한다.

16. 효율적인 CSS 사용을 위한 드림위버 최적화

가. 대표속성 사용 및 CSS 코더를 위한 사용자 환경설정



When creating CSS rules:

새로운 CSS 규칙을 생성할 때 대표속성을 사용하려면 모두 체크한다.

When editing CSS rules:

CSS 규칙을 수정할 때 상위의 설정에 따른다면 예제와 같이 체크하고 이미 코딩된 규칙에 따르려면 If original...에 체크한다. 예제와 같이 설정하는 경우 대표속성을 사용하지 않은 코드를 수정할 때 자동으로 대표속성으로 변경된다.

When double-clicking in CSS panel:

CSS Styles 패널을 더블클릭하면 CSS 편집을 위한 편의가 제공되는데 이때 대화상자, 속성패널, 코드뷰 가운데 사용자가 원하는 방식으로 편집할 수 있도록 동작 방식을 미리 설정할 수 있다.