

웹 접근성을 고려한 콘텐츠 제작기법

2004. 12

웹 접근성을 고려한 콘텐츠 제작 기법

2004. 12

연구책임자 : 최두진 센터장 (한국정보문화진흥원 정보격차연구센터)

공동연구책임자 : 김석일 교수 (충북대학교 전기전자 및 컴퓨터공학부)

참여연구원 : 홍경순 팀장 (한국정보문화진흥원 접근기술연구팀)

조웅희 대표이사 (우리인터넷)

신승식 과장 (LG 전자)

현준호 전임연구원 (한국정보문화진흥원 접근기술연구과)

서 문

인터넷은 이제 정치, 경제, 사회, 문화 등 모든 분야에서 이용되고 있습니다. 개인, 기업 및 국가 등 모든 분야에서 인터넷을 활용하여 기존의 산업사회에서는 상상할 수 없었던 형태의 서비스가 나타나고 있으며, 이를 통해 개인들의 삶의 질 및 기업과 국가는 생산성을 보다 높이고 있는 실정입니다.

그러나, 이러한 변화에도 불구하고 신체적인 제약, 소수가 사용하는 웹 브라우저, 운영체제를 사용하는 사람들은 인터넷의 정보에 접근조차 하지 못하고 있는 실정입니다. 이를 개선할 수 있는 방법이 웹 접근성(Web Accessibility) 준수입니다. 하지만, 웹 접근성에 대한 표준 및 지침은 이미 제정되어 활용되고 있으나 실제 웹 콘텐츠 제작시 참고할 수 있는 구체적인 실무 기법은 제공되지 않고 있습니다.

이에 본 연구에서는 우리나라의 웹 개발자 및 관리자들이 웹 접근성을 고려한 웹 콘텐츠 제작시 참고할 수 있는 기법들을 정리하고자 하였습니다. 본 연구를 통해 앞으로 보다 많은 사람들이 손쉽게 웹 접근성을 고려한 콘텐츠를 제작하여 장애인, 소수 사용자들도 인터넷의 혜택을 동등하게 누릴 수 있기를 기원합니다. 끝으로 본 연구를 수행하는 과정에서 많은 수고를 아끼지 않으신 한국정보문화진흥원 접근기술연구팀 연구원 여러분, 충북대학교 김석일 교수님, 우리 인터넷 조윤희 대표이사, LG 전자 신승식 과장님께 깊은 감사를 드립니다.

2004년 12월
한국정보문화진흥원
원장 손 연 기

목 차

I. 웹 접근성이란?	1
1. 웹 접근성 정의 및 효과	1
(1) 연구배경	1
2. 웹 접근성 정의 및 효과	4
(1) 접근성이란?	4
(2) 웹 접근성 준수의 효과	5
3. 국내외 웹 접근성 표준 동향	9
(1) 미국의 웹 접근성 표준 동향	9
(2) 일본의 웹 접근성 표준 동향	12
(3) W3C 웹 접근성 표준 동향	14
(4) 우리나라의 웹 접근성 표준 동향	15
II. 대체 텍스트	17
1. 대체 텍스트의 종류	18
(1) 텍스트 이미지	18
(2) 그림과 사진	21
(3) 링크가 걸린 이미지	23
(4) 불릿 이미지 등 의미 없는 그래픽 이미지	25
(5) 여러 개의 조각으로 나누어진 그림	31
(6) 여러 개가 합쳐진 한 이미지, 이미지 맵	33
2. 주요 사례별 대체 텍스트 제공 방법	38
(1) 소제목 텍스트 이미지	38

(2) 공지사항 등의 more(전체보기) 링크	40
(3) 약도	45
(4) 조직도	47
(5) 사이트 맵	53
(6) 반복적 이미지	59
(7) 광고, 캠페인	61
(8) 팝업창 내부	64
(9) 전화번호와 주소	66
 III. 동영상	69
1. 동영상과 멀티미디어 객체	69
2. 캡션(Caption)	71
3. 오디오 설명(Audio Description)	81
4. 원고/대본(Transcripts)	82
 IV. 색상	88
1. 색상을 이용한 정보 구분	88
2. 정보 전달을 위한 올바른 색의 사용	100
 V. 이미지 맵	106
1. 이미지 맵이란?	106
2. 클라이언트측 이미지 맵	107
3. 서버측 이미지 맵	109

VI. 프레임	116
1. 문서의 제목 붙이기	117
2. 프레임 제목 붙이기	125
3. 흔히 발견되는 프레임 오류	128
VII. 감박임, 움직임	134
VIII. 장치 독립성	140
1. 장치독립적인 이벤트 발생기 사용	141
2. 키보드 탐색	144
(1) tabindex 속성	145
(2) accesskey 속성	150
(3) 자동 초점 이동	152
(4) 키보드를 통한 메뉴 선택	156
IX. 링크를 통한 이동	160
1. 건너뛰기 링크	161
2. 사이트 맵	165
3. 페이지 내 목차	167
X. 시간 제한	170
1. 페이지 자동 이동	171

(1) 메타 리프레시(meta refresh)	171
(2) HTTP redirects	173
(3) 단순한 링크를 통한 이동	175
2. 빠른 내용의 전환	177
3. 새 창, 팝업 창	179
(1) 링크에 연결된 새 창	179
(2) 원하지 않는 팝업 창	184
 XI. 데이터 테이블	188
1. 표(table)의 종류	188
2. HTML에서 데이터 테이블의 바른 사용법	189
 XII. 콘텐츠의 배치	201
1. 스타일 시트와 테이블을 이용한 배치 비교	201
2. 스타일 시트의 바른 사용	208
3. 배치용 테이블(layout table)	217
 XIII. 온라인 양식(form)	222
1. 양식의 접근성	222
2. 각 컨트롤에 정확한 이름 붙이기(LABEL)	223
3. 유사한 컨트롤의 묶음 (FIELDSET)	238
4. 유사한 목록의 묶음 (OPTGROUP)	230
5. 양식의 키보드 접근성	232

XIV. 스크립트, 애플릿	241
1. 스크립트, 애플릿 삽입	241
(1) 대체적인 접근 방법 제공	241
(2) 흔히 발견되는 자바스크립트 오류	244
2. 플래시(Flash)	247
3. PDF(Portable Document Format)	255
(1) 실제 텍스트로 문서를 구성해야 함	255
(2) 적절한 태그를 붙여야 함	256
(3) 적절한 마크업을 붙여야 함	257
 XV. 별도 페이지	 262
1. 대체 페이지	262
2. 대체 페이지의 갱신	271

첨 부 : 한국형 웹 콘텐츠 접근성 지침 1.0

그림 목 차

<그림 1> 텍스트 이미지의 예	18
<그림 2> 대체 텍스트가 없는 텍스트 이미지	19
<그림 3> 대체 텍스트를 넣어 수정한 텍스트 이미지	19
<그림 4> 대체 텍스트가 없는 사진	21
<그림 5> 대체 텍스트를 추가한 사진	22
<그림 6> 대체 텍스트가 없는 링크 걸린 이미지	23
<그림 7> 대체 텍스트를 넣은 링크 걸린 이미지	25
<그림 8> 불릿 이미지	26
<그림 9> 대체 텍스트가 없는 불릿 이미지	27
<그림 10> 불릿 이미지 대신 아예 목록 마크업을 사용한 경우	28
<그림 11> 여러 개의 조각으로 나뉘어진 그림	31
<그림 12> 대체 텍스트가 없는 조각 그림	32
<그림 13> 대체 텍스트를 넣은 조각 그림	33
<그림 14> 여러 내용이 합쳐진 한 개의 큰 이미지	34
<그림 15> 대체 텍스트가 없는 소제목 텍스트 이미지	38
<그림 16> 대체 텍스트가 없는 전체 보기 링크	40
<그림 17> more(전체보기) 링크가 다수로 제공되는 경우	44
<그림 18> 대체 텍스트가 없는 약도 이미지	45
<그림 19> 대체 텍스트가 없는 클라이언트측 이미지 맵으로 된 조직도 ...	49
<그림 20> 클라이언트측 이미지 맵을 활용한 조직도	49
<그림 21> 제목에 대체 텍스트가 없는 사이트맵	54
<그림 22> 대체 텍스트가 잘못 지정된 예	55

<그림 23> 대체 텍스트가 없는 그래픽 사이트 맵	56
<그림 24> 대체 텍스트가 없이 등급을 이미지로 나타낸 경우	60
<그림 25> 배너 광고	62
<그림 26> 대체 텍스트가 없는 하나의 이미지로 된 팝업창	65
<그림 27> 대체 텍스트가 없이 이미지로만 제공되는 주소와 전화 번호 ...	67
<그림 28> 캡션이나 원고가 전혀 제공되지 않는 홈페이지	73
<그림 29> 캡션 미제공 사례	74
<그림 30> SMIL과 QuickTime Player을 이용하여	76
<그림 31> Real One Player에서 닫힌 캡션의 예 (On & Off 가능)	77
<그림 32> Windows Media Player에서 닫힌 캡션의 예	78
<그림 33> MAGpie 프로그램을 이용한 캡션 제작 화면	79
<그림 34> MAGpie를 이용해 캡션을 넣은 비디오 화면	80
<그림 35> MakeSAMI 프로그램 구동 모습	81
<그림 36> 동영상과 함께 원고(대본)를 제공한 사례	83
<그림 37> 미디어 형식을 사용자가 선택하도록 한 예	85
<그림 38> 색깔만으로 항목을 구분하기가 힘든 예	89
<그림 39> 색깔로만 '예'와 '아니오'를 구분한 사례	90
<그림 40> 계열들을 색과 모양으로 구분한 그래프	92
<그림 41> 색깔로만 버스 종류를 구분해놓은 사례	93
<그림 42> 필수 입력 필드를 색깔로만 표시한 잘못된 경우	94
<그림 43> 필수 입력 필드를 색깔과 문자로 동시에 표시한 사례	95
<그림 44> 글자의 굵기만으로 필수 기입사항을 표시한 사례	96
<그림 45> 명도(brightness)로 그래프의 막대를 구분한 사례	97
<그림 46> 색깔과 색이름을 같이 제시한 사례	98
<그림 47> 색깔만으로 필수 입력 항목을 표시한 예	99

<그림 48> 접근성을 고려해 필수 항목을 표시한 사례	100
<그림 49> 선천성 색각이상의 분류	103
<그림 50> 일반인, 1색각이상, 2색각이상에게 보이는 색상 시뮬레이션 ..	103
<그림 51> 전경과 배경이 대비가 잘 되지 않아 가독성이 낮은 사례 ...	105
<그림 52> 색상 대비가 잘 된 목록 상자	105
<그림 53> 서버측 이미지 맵 사용 예	111
<그림 54> 대체 텍스트가 없는 클라이언트측 이미지 맵	113
<그림 55> 제목이 없는 문서(웹 페이지)	118
<그림 56> 작업 표시줄에 나타난 제목이 없는 문서	119
<그림 57> 제목 없는 문서가 검색된 사례	120
<그림 58> 제목에 이상한 문자를 사용한 사례	121
<그림 59> 세부 페이지마다 정확한 제목을 제공한 사례	122
<그림 60> 제목이 없는 팝업창	123
<그림 61> 프레임 제목(title)을 붙이지 않은 경우	126
<그림 62> 깜박거림이 심한 국내의 사이트	136
<그림 63> 깜박임이 전혀 없는 영국의 BBC 방송 사이트	137
<그림 64> 대부분의 사진이 동시에 깜박거리 매우 혼란스러운 사례 ...	138
<그림 65> 국내 사이트들에서 자주 쓰이는	139
<그림 66> 마우스로만 조작 가능한 메뉴	143
<그림 67> 복잡한 tabindex 사용 예	148
<그림 68> 입력 순서를 tabindex 값을 통해 바르게 지정한 예	149
<그림 69> 탭키 순서가 지정되지 않아 불편을 주는 사례	150
<그림 70> 회원 가입시 흔히 사용되는 주민등록번호 입력의 문제점 ...	153
<그림 71> 키보드로 이동과 선택이 가능한 2단계 메뉴의 예	156
<그림 72> 경우에 따라 키보드로 이동과 선택이 가능한 2단계 메뉴 ...	157

<그림 73> 키보드 접근성을 고려하지 않은 플래시 메뉴	158
<그림 74> 보이지 않는 건너뛰기 링크가 제공되는 사례	161
<그림 75> 위의 사이트를 텍스트 전용 브라우저로 본 경우	162
<그림 76> 너무 많은 메뉴를 건너뛰는 방법이 없는 경우	164
<그림 77> 사이트 맵이 없는 홈페이지	165
<그림 78> 위계적으로 잘 구성해놓은 사이트 맵.	166
<그림 79> 목차 바로 가기가 필요한 경우	167
<그림 80> 복잡한 페이지에 먼저 제시된 목차	168
<그림 81> 자동으로 다른 페이지로 이동하도록 잘못 제작된 페이지의 예 ..	173
<그림 82> 수초마다 자동 갱신되는 뉴스	178
<그림 83> 경고 없이 무조건 새 창으로 뜨는 링크	181
<그림 84> 새 창으로 열릴 것을 미리 알려준 사례.	183
<그림 85> 여러 개의 원하지 않는 팝업 창이 뜨는 사례	186
<그림 86> 머릿글과 내용을 적절하게 짝지은 표의 예	195
<그림 87> 머릿글이 미치는 범위(scope)를 명시한 표	197
<그림 88> 표를 이용한 데이터의 표현	199
<그림 89> 짝짓기가 안 된 머릿글과 내용	199
<그림 90> 표를 바르게 수정한 예	200
<그림 91> CSS Zen Garden: 스타일 시트를 제거했을 때	206
<그림 92> CSS Zen Garden: 기본 스타일 적용	206
<그림 93> CSS Zen Garden: Paravion 스타일 적용	207
<그림 94> CSS Zen Garden: Golden Cut 스타일 적용	207
<그림 95> CSS Zen Garden: The hall 스타일 적용	208
<그림 96> 잘못된 스타일 시트의 사용	208
<그림 97> 스타일 시트를 빼면 순서가 어긋나는 경우	209

<그림 98> 스타일 시트를 바로잡은 예	211
<그림 99> 스타일을 빼도 논리적 이해가 가능한 경우	211
<그림 100> 잘못된 스타일 시트 사용 예	215
<그림 101> 스타일 시트를 빼면 어긋나는 메뉴	216
<그림 102> 양식 입력시 테이블을 잘못 사용한 예	217
<그림 103> 테이블을 없애면 순서가 뒤바뀐 예	218
<그림 104> 테이블 대신 CSS로만 내용물 배치를 한 사례	219
<그림 105> CSS를 제거했을 때 모습	220
<그림 106> 라디오 버튼과 레이블을 짝지은 예	224
<그림 107> 시각적 결함을 <label>로 보완한 예	225
<그림 108> 레이블을 명시하지 않아 장애인에게 선택을 어렵게 한 예 ...	227
<그림 109> <fieldset>을 사용하여 유사 컨트롤을 하나로 묶은 예	229
<그림 110> 목록상자 내의 유사 항목을 <optgroup>으로 묶은 예	231
<그림 111> 잘못된 양식 사용 예	233
<그림 112> 이동 버튼이 따로 없는 잘못된 목록 상자(list box)	237
<그림 113> 이동 버튼이 따로 있는 바른 목록 상자	239
<그림 114> 키보드(탭)를 이용한 이동이 불가능한 경우	239
<그림 115> 접근성이 떨어지는 플래시 메뉴	250
<그림 116> 접근성을 고려한 플래시 메뉴	251
<그림 117> 플래시와 텍스트 메뉴를 함께 제공한 예 (플래시 버전) ...	252
<그림 118> 플래시와 텍스트 메뉴를 함께 제공한 예 (텍스트 버전) ...	253
<그림 119> 캡션을 삽입한 플래시 무비	254
<그림 120> 워드에서 구조적으로 문서를 작성한 예	258
<그림 121> 파워포인트에서 이미지에 대체 텍스트를 삽입하는 대화창 ...	259
<그림 122> 구조적인 PDF 문서	260

<그림 123> 저속 인터넷 사용자를 배려한 Low Graphics version 링크 ...	264
<그림 124> 동일한 뉴스를 담고 있는 Low Graphics version	265
<그림 125> 전체 플래시로 제작된 사이트	266
<그림 126> 전체가 플래시 애니메이션으로 제작된 사이트	267
<그림 127> 별도 장애인 페이지로 가는 링크가 없는 경우	268
<그림 128> 전혀 다른 인터페이스를 사용하는 별도의 음성 홈페이지 ...	269
<그림 129> 최신 소식이 올라온 일반 홈페이지	271
<그림 130> 최신 소식이 올라오지 않은 시각장애인용 페이지	272

I. 웹 접근성이란?

1. 웹 접근성 정의 및 효과

(1) 연구배경

- 정보통신기술, 특히 인터넷의 급속한 확산은 개인, 기업, 국가 활동을 급격하게 변화시키고 있는 실정
 - 즉, 개인적인 측면에서는 인터넷을 통한 인터넷 뱅킹, 재택근무, e-Learning, 온라인 게임 등 다양한 디지털 문화 및 경제를 향유할 수 있게 되었으며,
 - 기업측면에서는 전자상거래, 산업 정보화 등을 통해 기존의 산업사회에서는 보지 못하던 급격한 변화에 직면함
 - 국가 또한 인터넷의 발전으로 전자민주주의, 전자정부 등을 통한 정부의 효율성 및 투명성 증대를 요구 받고 있는 실정
- 인터넷은 현대사회에서 가장 핵심적이고 포괄적인 사회 변동 요인으로 평가되어지고 있음
- 이러한 급격한 변화와 더불어 우리사회에서 직면하고 있는 문제가 바로 고령인구의 증가와 장애 인구의 증가 문제임
 - 통계청(2001) 인구통계에 따르면, 우리나라는 2000년에 들어서면서 고령화 사회(Aging Society)의 문턱을 넘어섰으며,

- 2019년에 이르면 고령사회(Aged Society)로 진입하고 2026년에는 초고령 사회(Super-aged Society)가 될 것이라고 예측함
- 보건복지부(2003. 6)가 발표한 장애인 출현율 예측에 따르면 1997년 장애인 출현율이 2.51%이었던 반면,
 - 2004년에는 3.69%로 증가할 것이라고 예측하고 있으며, 2004년 추정 장애인 수는 약 170만명으로 파악됨
- 하지만, 이러한 새로운 변화에 발빠르게 대응하는 개인, 조직 및 국가가 있는 반면,
 - 정보·지식의 독점 및 불평등으로 인하여 이를 향유하지 못해 기존의 사회에서는 나타나지 않은 불평등으로 인해 새로운 사회적 갈등이 나타남
- 즉, 지식정보화시대에서 사회 생활을 영위하기 위한 필수적인 수단인 컴퓨터와 인터넷을 활용하지 못해 불이익을 받는 정보격차(Digital Divide) 현상이 나타나고 있음
 - 이를 방치할 경우 개인적으로는 사회참여와 소득창출의 기회가 제한되고,
 - 사회적으로는 빈부격차 심화 및 계층 간의 갈등이 심화될 우려가 있으며,
 - 이는 사회·경제적인 격차를 확대 재생산함은 물론 사회통합을 저해할 것임

- 이에 저소득층, 장애인, 노인 등의 취약계층이 소득증대와 삶의 질을 향상시킬 수 있는 가장 효과적인 수단인 IT를 생산적으로 활용할 수 있도록
 - 정보격차해소를 참여복지 및 삶의 질 향상을 위한 핵심 정책과제로 채택하여 추진해 나가야 함
- 정보격차 문제를 발생시키는 신체적, 경제적, 지역적인 격차 등에 대한 논의는 다소 이루어져 온 반면,
 - 인터넷에 접근하지 못하고 활용하지 못하는 근본적인 원인인 웹 접근성(Web Accessibility) 부재에 대한 우리나라의 노력은 부족한 실정임
 - 이미 미국, 영국, 호주 등 선진국에서는 웹 접근성 부재를 정보격차 문제의 주요한 장애요인으로 파악하고 이에 대한 적극적인 노력을 진행 중에 있음
- 우리나라는 웹 접근성 제고를 위해 권장지침 제정, 표준화를 위한 민간 포럼 운영, 세미나 등을 진행함
 - 그러나, 웹 접근성에 대한 논의가 이루어지고 있음에도 불구하고 실제 웹 콘텐츠 제작자들이 참고할 수 있는 구축 실무 사례는 전혀 제공되지 않고 있는 실정
- 이에 본 연구에서는 웹 콘텐츠 제작자들이 접근성을 고려한 콘텐츠를 작성할 수 있도록 다양한 사례를 제공하고자 함

2. 웹 접근성 정의 및 효과

(1) 접근성이란?

- 정보통신 접근성(Accessibility) 개념은 이를 정의하는 학자 및 기관에 따라 다양하게 나타나고 있는 실정
 - 접근성에 대한 개념의 다양성으로 인한 인식의 부족보다는 접근성에 대한 개념을 잘 못 이해하고 있는 것이 더욱 문제임
 - 즉, 접근성을 단지 장애인에게 국한된 문제라고 잘못 이해하고 있는 경우가 대부분이라는 것이 문제임
 - 비록 접근성 준수가 장애인에게 가장 혜택이 많이 돌아가는 것은 사실이지만, 접근성이란 장애인뿐만 아니라 모든 사람이 정보통신기기나 서비스를 손쉽게 활용할 수 있도록 만드는 것을 말하는 것임
 - 예를들어, 리모컨, 전화, 자동문 등의 사례에서 알 수 있듯이, 초기에는 노인·장애인들을 위해 개발된 제품들이 궁극적으로는 모든 사람들이 편하게 활용할 수 있다는 것
 - 이러한 접근성 개념은 정보통신 분야에서 다양한 제품 및 서비스간의 호환성 문제, 이동 정보통신기기 및 서비스의 확대 등으로 그 필요성 및 중요성이 더욱 증대

< 정보통신 접근성 주요 정의 >

구 분	정 의
「장애인·노인 등의 정보통신 접근성 향상을 위한 권장지침」	정보통신기기·소프트웨어 및 콘텐츠와 정보통신 서비스 등 정보통신 제품과 정보통신 서비스를 활용하고자 하는 사람에게 자유로운 접근과 이용을 보장 (2002. 1 정보통신부 고시)
ISO (International Standards Organization)	다양한 능력, 숙련, 요구, 취향을 가진 개인들이 정보에 접근할 수 있도록 하는 글로벌 요구사항 (ISO 13497, 1997).
Benyon (2001)	장비의 물리적 접근(Physical access to equipment)과 모든 잠재적 사용자들을 위한 하드웨어와 소프트웨어 조작의 적합성(Operational suitability)

(2) 웹 접근성 준수의 효과

- 웹의 창시자인 Tim Berners-lee에 따르면, 웹의 힘(Power)은 보편성에 있으며, 장애에 구애 없이 모든 사람이 접근할 수 있는 것이 가장 중요한 요소라고 설명
- 웹 사이트에서 접근성이 보장되어야 하는 당위성으로는 다음과 같이 8가지를 들 수 있다(Thatcher et.al., 2002; Joe Clark, 2003).

1) 규정과 법적 요구사항에 대한 준수

- 법적인 규정을 통해 접근성 보장에 대한 동기를 유발시키고, 이로 인해 웹 사이트를 사용할 수 있는 이용자의 범위를 확대시킴
- 정부에서의 규정과 요구사항 강제화를 촉발시킨 2가지 주요 사례는 호주의 시드니 올림픽 조직위원회의 소송과 미국 AOL(America Online) 소송을 들 수 있음

2) 장애인, 노인 등을 포함한 다양한 범위의 이용자 확대

- 웹 접근성이 확보가 되면 웹 사이트를 효과적으로 이용할 수 있는 사람의 수가 증가되며, 이는 결국 대상 고객 및 이용 환경의 범위를 보다 확대시키는 결과를 얻을 수 있음
- 비즈니스 측면에서 볼 때, 웹 접근성에 대한 보장으로 고객들의 웹사이트 이용을 제한하는 장애 요인들이 제거됨으로써 웹 사이트를 통한 매출이 증대되는 효과가 나타남

3) 새로운 장소, 새로운 기기 개발 등 이용 상황의 확대

- 접근성이 확보된 웹 사이트는 효과적으로 이용될 수 있는 상황들이 확대됨에 따라, 결국 이러한 경우의 수가 증가되어 이용의 확대를 가져옴
- 접근성의 기법을 이해하고 이를 웹 사이트에 적용하게 되면, 현재의 최신기기 뿐만 아니라 향후 등장하게 될 어떤 정보기기에서나 큰 불편 없이 해당 웹 사이트를 제공할 수 있는 장점이 있음

- 자연환경이나 주변 환경에 따른 상황적 제약은 새롭게 등장하는 기술 및 기기들의 디자인에 영향을 주며, 때로는 이러한 제약 조건들이 제품에 영향을 미치는데,
- PDA, 휴대폰 등도 이제는 인터넷 접속을 할 수 있게 된 것이 그 대표적 예라 할 수 있음

4) 디자인 및 설계에 있어서의 효율성 제고

- 접근성의 수준을 향상시키면 웹 사이트의 디자인 및 작업 설계에 있어서 상당한 효과를 가져올 수 있음
- 이는 궁극적으로 검색엔진의 색인과정에도 편의성을 도모할 수 있으며, 사이트 개편, 다국어 버전 제작 시에도 편리하며, 다른 언어 버전으로 사이트를 추가 제작할 때에도 편의성이 확보되는 장점이 있음

5) 비용 절감의 효과

- 초기에는 접근성의 개념을 이해하고 이를 구현하기 위해 많은 비용과 시간이 투입되지만, 접근성 개선으로 사이트 제작 총 비용은 절감되는 효과를 얻을 수 있음
- 접근성 향상으로 유용성(usability)도 확보될 수 있기 때문에 온라인 쇼핑의 매출이 증가하고 사이트 운영에 필요한 비용은 감소하며,
- 따라서 장기적인 관점에서 비용 절감 효과를 거두게 되며, 접근성에 투자한 만큼 수익이 증대되는 효과를 얻을 수 있음

- 온라인으로 제품과 서비스를 제공함으로써 오프라인에서 장애인을 위한 별도 시설을 마련하기 위해 지출해야 하는 비용 등을 줄일 수 있으며,
- 전화 요금 등의 각종 공과금, 신용카드 비용, 은행 입출금 내역 등을 온라인으로 서비스한다면, 오프라인으로 청구서를 발행하는 비용을 줄일 수 있으므로 회사측에도 유리하고,
- 온라인으로 편리하게 청구 금액 등을 손쉽게 확인하고 결제할 수 있으므로 이용자 측에도 유용한 것임

6) 홍보 효과 향상

- 웹 접근성을 고려한 웹사이트라는 인식이 제고되면 사이트의 주체가 되는 조직이나 단체의 이미지에 긍정적인 영향을 줄 수 있음
- 기업의 사회적 책임(Corporate Social Responsible)이 중요하게 부각되고 있는 시점에서 웹 접근성 준수는 홍보 수단으로도 큰 역할을 할 수 있을 것임

7) 자발적 관심유도

- 접근성에 대한 인식을 확산시키기 위해서는 웹 사이트 제작 담당 실무자 몇 사람이 아니라 웹사이트를 구축하는 주체인 조직이나 단체에서 자발적인 관심을 가지는 것이 바람직함

- 개인과 조직이 접근성 확보로 인해 얻어지는 혜택을 인지한다면, 자발적으로 접근성 확보를 위해 노력할 것임

8) 균등기회 보장

- 웹 접근성을 고려한 웹 사이트를 제작하게 되면 사용자 누구에게나 균등하게 기회를 제공하게 되며,
 - 조직의 목적과 비즈니스를 위하여 웹 사이트의 활용 범위와 활용 가능 상황을 확대하면 보다 많은 이용자에게 보다 많은 이용 기회를 부여하게 된다는 것임
 - 장애인에게 업무, 여가, 사회적 활동에 참여할 기회를 제공할 뿐만 아니라 보다 많은 사람들이 필요한 일을 보다 쉽게 처리할 수 있도록 도와 줄 것임

3. 국내외 웹 접근성 표준 동향

(1) 미국의 웹 접근성 표준 동향

- 미국은 웹 접근성 제고를 위해 다양한 법·제도를 마련
 - 1996년 개정된 미국 통신법(The Telecommunication Act) 255조, 미국 장애인법(ADA : American with Disability

Act) 등을 통해 장애인의 정보통신 접근성 제고의 기틀을 마련함

- 특히 1998년 개정된 미국 재활법(The Rehabilitation Act)에서는 전자 및 정보기술을 개발, 보급, 유지보수 및 사용하는 경우
 - 연방정부가 무리한 부담(Undue Burden)되지 않는 한 장애인들이 일반인과 동등하게 접근할 수 있도록 보장해야 한다고 명시하고 있음
- 동 법은 장애인들의 웹 접근성 향상에 매우 큰 기여를 하였는데, 본 법의 준수를 위해 제정된 기술표준 중 §1194.22인 ‘웹 기반 인트라넷과 인터넷 정보 및 응용 프로그램’에서
 - 웹 접근성 제고를 위해 아래와 같이 16개 지침을 제정·공포하여 미국의 모든 연방정부 홈페이지가 이를 준수하고 있음(2001년 6월부터)

§1194.22 항목번호	요구 조건 (내용)
(a)	텍스트가 아닌 요소에도 해당하는 텍스트 요소가 반드시 제공되어야 한다 ("alt," "longdesc"를 사용하거나 요소 내용 안에 표시한다).
(b)	멀티미디어에는 동등한 대체 요소가 멀티미디어 내부에 합성되어야 한다.
(c)	웹 문서에 표시된, 색상을 이용한 정보는, 색상이 없는 상태에서도 판독할 수 있어야 한다. 예를 들어 문맥을 통해서나 마크업으로 표시되어야 한다.
(d)	문서는 연결된 스타일 시트를 불러오지 않고도 읽을 수 있어야 한다.
(e)	서버 측의 이미지맵의 각 활성화된 부분마다 별도의 텍스트 링크가 지정되어야 한다.
(f)	이미지맵의 각 부분을 기하학적 모양으로 지정할 수 없을 경우가 아니면, 서버 측의 이미지맵 대신 클라이언트 측의 이미지맵을 제공해야 한다.
(g)	데이터 테이블의 행과 열 헤더를 반드시 지정해야 한다.
(h)	행과 열의 헤더에 있어서 2개 이상의 논리 수준을 갖는 데이터 테이블에서는 마크업을 사용하여 데이터 셀과 헤더 셀을 연결해야 한다.
(i)	프레임은 프레임 확인과 이동을 쉽게 하기 위하여 텍스트로 이름을 지정해야 한다.
(j)	웹 문서는 스크린이 2Hz 이상 55Hz 이하의 주파수로 깜빡이는 일이 없도록 작성해야 한다.
(k)	어떤 웹 사이트가 이 장의 조항을 준수하도록 작성하기 위해 다른 방법이 없을 때는, 정보와 기능을 동등하게 제공하는, 텍스트만으로 작성된 별도의 웹 문서를 함께 제공해야 한다. 텍스트만으로 작성된 웹 문서의 내용은 원래의 문서 내용이 업데이트 될 때마다 함께 업데이트해야 한다.
(l)	웹 문서에서 스크립트 언어를 사용하여 내용을 표시하거나 인터페이스 요소를 사용할 때는, 스크립트가 제공하는 정보를 기능적 텍스트로 표시하여 보조 기기로 판독할 수 있게 해야 한다.
(m)	웹 문서에서 내용 판독을 위해 애플릿, 플러그인, 기타 다른 응용 프로그램을 클라이언트 시스템에 요구하는 경우, 해당 문서에는 §1194.21의 (a)항부터 (l)항을 준수하는, 플러그인이나 애플릿을 제공하는 주소를 링크해야 한다.
(n)	전자 양식을 사용하고 온라인 상태에서 작성하도록 하는 경우, 사용자가 전자 양식을 작성 및 제출하고 안내문 및 지시문을 이해하는데 필요한 정보, 필드 요소, 기능 등을 접근하기 위해 보조 기술을 사용할 수 있어야 한다.
(o)	반복되는 이동 링크를 건너뛸 수 있는 방법을 사용자들에게 제공해야 한다.

(2) 일본의 웹 접근성 표준 동향

- 일본은 우리나라와 마찬가지로 고령화가 급속히 진행중이어서 2015년에는 국민의 25%가 65세 이상이 될 것으로 전망
 - 이러한 초고령화 사회로의 진입에 대비하기 위하여 일본 정부는 고령자가 사용하기 쉬운 제품, 서비스, 생활환경을 정비할 계획을 수립하고 정보통신기기와 서비스를 제공함에 있어서 접근성을 고려하도록 하고 있음
- 이에 따라 일본에서는 처음으로 일본이 의장국으로서 세계 각국의 협력을 얻어 성립시킨 최초의 ISO 가이드인 'ISO/IEC 가이드 71'¹⁾를 제정하였으며, 이를 근간으로 2003년 6월20일에 일본공업규격 'JIS Z 8071'²⁾을 제정
 - 그러나 ISO/IEC 가이드 71 또는 JIS Z 8071는 어디까지나 규격 책정을 위한 지침이므로 이 지침에 의거하여 제정된 제품이나 서비스의 관련되는 표준이 제정되지 않는다면 의미가 없음
 - 이에 따라 일본규격협회에서는 2001년 4월, 협회산하의 정보기술표준화센터(INSTAC)에 “정보기술 분야공통 및 소프트웨어 제품의 접근성 향상에 관한 표준화 조사위원회”를 설치하고

1) ISO/IEC GUIDE 71, “Guidelines for standards developers to address the needs of older persons and persons with disabilities,” 2001년 11월 제정

2) JIS Z 8071, “고령자 및 장애인을 위한 규격 작성 배려 지침,” 2003년 6월20일 제정

- 정보기술분야의 접근성향상을 위한 공통지침, 정보처리장치 접근성 향상을 위한 지침 및 웹콘텐츠의 접근성 향상을 위한 지침 등 3가지 분야의 규격을 초안을 마련하고
- 의견 수렴과정을 통해 2004년 5월20일에는 제1부: 공통지침(JIS X 8341-1:2004), 제2부: 정보처리장치(JIS X 8341-2:2004)에 관한 일본공업규격을 제정하였으며,
- 2004년 6월20일에는 제3부: 웹컨텐츠(JIS X 8341-3:2004)에 관한 일본공업규격을 제정

- 현재 웹 콘텐츠 접근성에 관한 일본공업규격은 임의 표준이나 앞으로 공공기관이나 대기업체를 중심으로 강제 표준화될 것으로 보임

□ JIS X 8341-3에서 웹 접근성 향상을 위한 지침은 모두 9개로 구성되어 있으며, 각각의 지침은 또다시 몇 개의 세부 검사항목으로 구성되어 있음

- 총 검사항목의 수는 39개임

지침번호	제 목 (내용)	검사항목의 수
5.1	규격 및 사양	2
5.2	구조 및 표시 스타일	7
5.3	조작 및 입력	9
5.4	텍스트 아닌 콘텐츠의 표현	5
5.5	색깔 및 형태	3
5.6	글자의 속성	3
5.7	오디오의 속성	2
5.8	콘텐츠의 움직임과 깜박임의 제한	2
5.9	언어 및 문장과 관련한 접근성	6

(3) W3C 웹 접근성 표준 동향

- 미국을 제외한 영국, 호주, 뉴질랜드, 유럽연합, 영국 등에서는 국제표준화 기구인 W3C(World Wide Web Consortium) 웹 접근성 이니셔티브(Web Accessibility Initiatives)의
 - 웹 콘텐츠 접근성 지침(WCAG : Web Contents Accessibility Guidelines) 1.0을 표준으로 채택하고 있음
 - 본 지침은 웹 접근성 이니셔티브 산하의 웹 콘텐츠 접근성 가이드라인 워킹그룹에서 1999년 5월 제정함
- WCAG 1.0 지침은 크게 14개의 지침(Guideline)과 65개의 검사항목(Checklists)으로 구성되어 있으며,
 - 본 지침은 중요도에 따라 1, 2, 3으로 구성되어 있음
- 2004년 10월 현재 WCAG 2.0 지침의 working draft (2004. 7월 제정)가 공개 중으로 일반인들을 대상으로 의견을 수렴하고 있으며,
 - WCAG2.0에 따르면 웹콘텐츠의 접근성을 제고하기 위해 반드시 지켜야 할 원칙으로 인지가능성(Perceivable), 운영가능성(Operable), 이해가능성(Understandable), 견고성(Robust)의 4가지 원칙을 제시

(4) 우리나라의 웹 접근성 표준 동향

- 국내 웹 접근성 관련 표준으로는 2002년 1월 정보통신부 고시로 제정된 “장애인·노인등의정보통신접근성향상을위한권장지침”이 있음
 - 정보통신부 권장지침은 크게 총칙, 정보통신 제품과 서비스의 기능에 대한 설계지침, 웹 문서에 대한 설계 지침 및 보칙으로 구성되어 있으며,
 - 웹 문서에 대한 설계지침은 10개로 구성
- 2002년 제정된 정보통신부 권장지침이 실제 홈페이지 제작시 참고하기에는 어려우며 새로운 웹 기술을 반영하지 못하고 있어
 - 정보통신접근성향상 표준화포럼을 중심으로 2003년 12월에 “한국형웹콘텐츠 접근성지침 1.0”을 마련(2004년 TTA 단체표준으로 제정)

지침 번호	지침의 요구 조건 (내용)
1	텍스트 아닌 콘텐츠(non-text contents) 중에서 글로 표현될 수 있는 모든 콘텐츠는 해당 콘텐츠가 가지는 의미나 기능을 동일하게 갖추고 있는 텍스트로도 표시되어 있어야 한다.
2	시간에 따라 변화하는 영상매체는 해당 콘텐츠와 동기되는 대체 매체를 제공해야 한다.
3	콘텐츠가 제공하는 모든 정보는 색상을 배제하더라도 인지할 수 있도록 구성되어야 한다.
4	이미지 맵 기법이 필요할 경우에는 클라이언트측 이미지 맵을 사용하거나 서버측 이미지 맵을 사용할 경우에는 동일한 기능을 하는 텍스트로 구성된 대체 콘텐츠를 제공해야 한다.
5	콘텐츠를 구성하는 프레임의 수는 최소한으로 하며, 프레임을 사용할 경우에는 프레임별로 제목을 붙여야 한다.
6	콘텐츠는 스크린의 깜빡거림을 피할 수 있도록 구성되어야 한다.
7	키보드 (또는 키보드 인터페이스)만으로도 웹 콘텐츠가 제공하는 모든 기능을 수행할 수 있어야 한다.
8	웹 콘텐츠는 반복적인 네비게이션 링크(repetitive navigation link)를 뛰어넘어 페이지의 핵심부분으로 직접 이동할 수 있도록 구성하여야 한다.
9	실시간 이벤트나 제한된 시간에 수행하여야 하는 활동 등은 사용자가 시간에 구애받지 않고 읽거나, 상호작용을 하거나 응답할 수 있어야 한다.
10	데이터 테이블은 테이블을 구성하는 데이터 셀의 내용에 대한 정보가 충분히 전달될 수 있어야 한다.
11	콘텐츠의 모양이나 배치는 논리적으로 이해하기 쉽게 구성하여야 한다.
12	온라인 서식을 포함하는 콘텐츠는 서식 작성에 필요한 정보, 서식 구성 요소, 필요한 기능, 작성 후 제출 과정 등, 서식과 관련한 모든 정보를 제공해야 한다.
13	스크립트, 애플릿 또는 플러그 인(plugin) 등과 같은 프로그래밍 요소들은 현재의 보조기술의 수준에서 이들 프로그래밍 요소들의 내용을 사용자에게 전달해줄 수 있을 경우에만 사용하여야 한다.
14	콘텐츠가 항목 1에서 13에 이르는 13개 검사 항목을 만족하도록 최대한 노력하였으나 해결되지 않는 부분이 남아있다면 텍스트만의 콘텐츠를 제공하는 웹 페이지(또는 웹사이트)를 별도로 제공해야 한다.

II. 대체 텍스트

지침 1. 텍스트가 아닌 콘텐츠(non-text contents) 중에서 글로 표현될 수 있는 모든 콘텐츠는 해당 콘텐츠가 가지는 의미나 기능을 동일하게 갖추고 있는 텍스트 모드로 표시되어 있어야 한다.

① 텍스트 아닌 콘텐츠 중에서 글로 표현될 수 있는 콘텐츠는 모두 해당 콘텐츠와 정확하게 일치하는 텍스트 콘텐츠(text contents)와 함께 제공되어야 한다.

② 만약, 텍스트 아닌 콘텐츠 중에서 글로 표현될 수 없는 경우는 해당 콘텐츠에 대한 설명을 텍스트로 제공하여야 한다. 이 때, 제공되는 텍스트 콘텐츠는 설명하고자 하는 원래의 콘텐츠의 내용이나 형태 등의 상세한 정보를 제공해야 한다.

③ 텍스트 아닌 콘텐츠를 대체하거나 설명하기 위해 제공되는 텍스트는 콘텐츠 제작자가 원래의 콘텐츠를 사용해 표현하고자 했던 의미나 기능도 설명해야 한다.

④ 텍스트 아닌 콘텐츠를 대체하거나 설명하기 위해 제공되는 텍스트를 읽어줌으로 인하여 의미의 혼동을 줄 경우에는 빈 문자(blank text)를 대체 텍스트로 사용해야 한다.

1. 대체 텍스트의 종류

(1) 텍스트 이미지

- 텍스트 이미지란 글자나 단어, 문장 등 간단하고 짧은 글자들의 집합을 이용해서 이미지로 디자인한 것을 말함.
- 이 경우, 웹 접근성 준수를 위해서는 원래 제공하고자 했던 텍스트를 그대로 대체 텍스트에 넣어주면 됨.
- 때로 이미지 상으로 나타난 텍스트가 시각적인 효과를 위해 두 번 이상 같은 말을 중복적으로 나타낸 경우에는,
- 대체 텍스트를 한 번만 기술해주어야 함.

X (Bad)

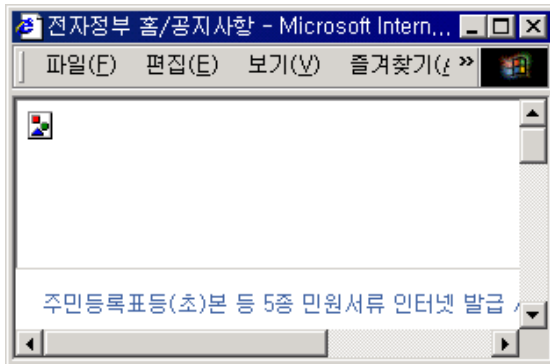
<그림 1> 텍스트 이미지의 예



※ 출처: 대한민국 전자정부(<https://www.egov.go.kr>), 2004. 04. 08

X (Bad)

<그림 2> 대체 텍스트가 없는 텍스트
이미지



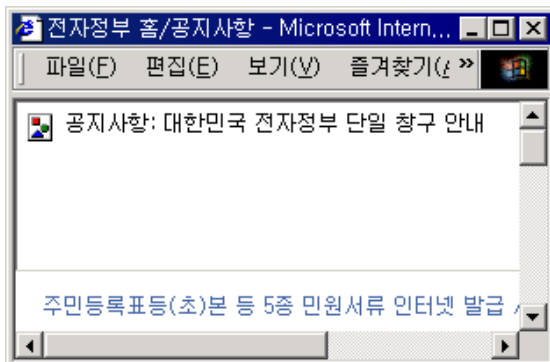
브라우저 옵션에서 그림을
꺼놓았을 때 그림이 무엇이
있는지 전혀 알 수 없음

```

```

O (Good)

<그림 3> 대체 텍스트를 넣어 수정한
텍스트 이미지



브라우저 옵션에서 그림을
꺼놓았을 때에도 그림이
무엇을 나타내는지 짐작할
수 있음.

```
<h1></h1>
```

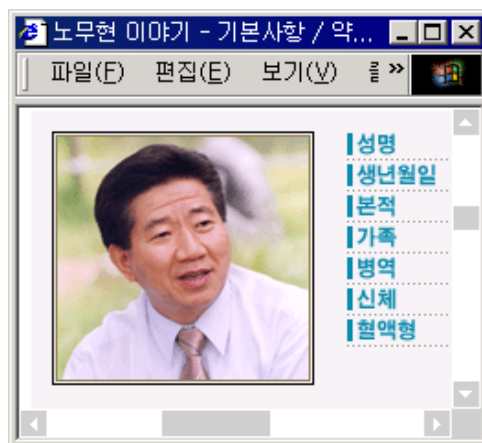
- 위의 사례에서는 공지사항의 제목을 이미지로 구성하였는데, 대체 텍스트가 없음.
- 이 경우 대체 텍스트를 추가해주기 위해 `` 요소에 `alt="공지사항: ..."`을 추가해주어야 하고,
- 해당 문서의 가장 큰 제목임을 의미적으로 나타내는 `<h1></h1>`을 추가해주는 것도 좋은 방법임.

(2) 그림과 사진

- 일반적인 인물, 풍경, 회사 로고, 상품 및 제품 사진 등에는
 - 해당 인물의 이름, 풍경의 제목 및 간단한 묘사, 회사 이름, 상품이나 제품의 이름 등을 그대로 대체 텍스트로 넣는 것이 바람직함.

X (Bad)

<그림 4> 대체 텍스트가 없는 사진

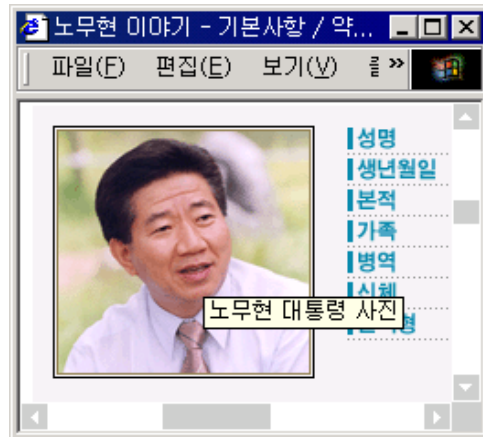


※ 출처 : 청와대 홈페이지(www.president.go.kr), 2004. 5. 9

```
<IMG  
src="/kr/pages/president/story/images/title_030101_  
img.gif">
```

O (Good)

<그림 5> 대체 텍스트를 추가한 사진



```
<IMG  
src="/kr/pages/president/story/images/title_030101_  
img.gif" alt="노무현 대통령 사진">
```

(3) 링크가 걸린 이미지

- 이미지에 링크가 걸려있을 때에는 그 링크의 성격에 맞는 대체 텍스트를 제공해야 함.
 - 추가로 링크 자체에도 링크의 성격을 기술해주는 title 속성을 넣어주면 일반 그래픽 브라우저 사용자에게도 풍선 도움말(tooltip)로 링크에 대한 설명이 제공되는 장점을 줄 수 있음.
- 주의할 것은 ‘이미지의 역할을 그대로 대체할 수 있는’ 대체 텍스트를 제공해야 한다는 것
 - 즉, 이미지의 모양에 대해 설명하는 것이 아니라, 이미지가 제공하는 의미적인 내용을 대체 텍스트로 제공해야 함.

X (Bad)

<그림 6> 대체 텍스트가 없는 링크
걸린 이미지



※ 출처 : 대한민국 전자정부(<https://www.egov.go.kr/>), 2004. 4. 11

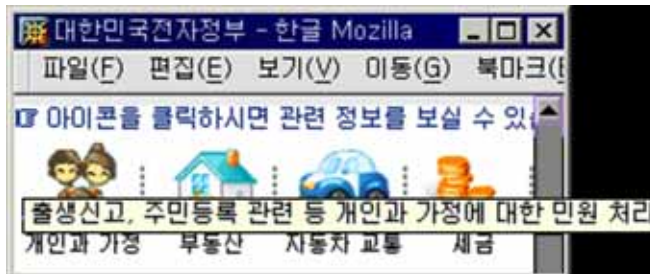
<!--대체 텍스트를 빼먹은 경우-->

<!--이미지의 모양에 대해 설명하려고 하여 대체 텍스트를 잘 못
넣은 경우-->

- ☐ 위 그림의 경우 다음과 같이 대체 텍스트를 넣어 수정할 수 있음.

O (Good)

<그림 7> 대체 텍스트를 넣은 링크 걸린 이미지



```
<!-- 이미지의 성격에 알맞은 대체 텍스트를 alt 속성값으로  
넣었으며, 링크에 대한 보다 자세한 설명은 title로 넣었음.-->  
<a  
href="https://www.egov.go.kr/main?a=AA020InfoSubMainApp&HighCtgCD=A01" title="출생신고, 주민등록 관련  
등 개인과 가정에 대한 민원 처리"></a>
```

(4) 불릿 이미지 등 의미 없는 그래픽 이미지

□ 의미 없는 이미지란 불릿 이미지, 단순히 디자인을 위해 사용하는 이미지를 말함.

○ 불릿(bullet) 이미지란 목록(list) 앞에 조그맣게 표시하는 별표, 동그라미, 사각형 같은 이미지를 말함.

○ 그 밖에도 페이지를 꾸미기 위해 가로로 늘어진 선 모양

의 그림, 세로선 모양의 그림, 또는 모서리 부분을 꾸미기 위한 그림 등 다양하게 사용되고 있음.

□ 의미 없는 이미지의 대체 텍스트 제공을 위해서

- 그림 모양을 설명하는, 또는 그림과 비슷한 모양의 문자를 넣는 것은 잘못된 방법이며,
- 그렇다고 대체 텍스트를 아예 쓰지 않는 것도 잘못된 것임.

□ 의미 없는 이미지에 대해서는 따옴표만 있고 아무 내용도 없는 (alt=""와 같은) 대체 텍스트를 반드시 넣어주어야 함.

X (Bad)

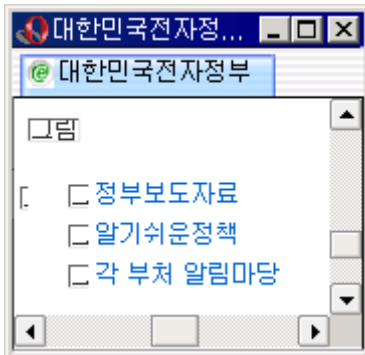
<그림 8> 불릿 이미지



※ 출처: 대한민국 전자정부(<https://www.egov.go.kr>), 2004. 4. 11

X (Bad)

<그림 9> 대체 텍스트가
없는 불릿 이미지



그림을 표시하는 옵션을 껐을 때, 대체 텍스트가 없으므로 목록 앞에 있는 그림도 깨져 보임.

- 의미 없는 이미지에 대해 시각 장애인이 잘못된 정보를 얻게 되는 이유로는 크게 다음과 같은 세 가지 이유가 있음.

- 대체 텍스트를 아예 빼먹은 경우

```

```

- 대체 텍스트에 비슷한 모양의 문자를 붙인 경우

```

```

- 대체 텍스트에 이미지 모양을 그대로 설명한 경우

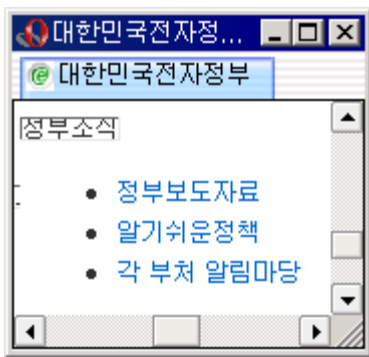
```

```

- 의미 없는 이미지에 대해서도 빈 문자로 된 대체 텍스트 제공은 반드시 필요함.

O (Good)

<그림 10> 불릿 이미지 대신
아예 목록 마크업을 사용한
경우



목록을 나타내는 마크업(,)을
사용하였으므로, 그림 표시 옵션을 꺼
도 목록을 알아볼 수 있음.

- 빈 대체 텍스트를 넣어 수정한 경우

```

```

- 스타일시트와 목록으로 대체한 경우

```
<!-- 목록의 스타일을 정의함. 별도의 CSS 파일로 저장할 수도
있고, 같은 HTML 파일 내에서 HEAD 부분에 넣을 수도 있음.-->
```

```
<STYLE type="text/css">
```

```

ul.news li {list-style-image:
url(main_data/mt3_government_icon.gif);}

/*
ul.news는 ul요소중에 news라는 클래스를 갖는 것. ul.news
li는 ul.news 요소의 자손으로 li가 나타난 경우를 의미.(자손
선택자(descendant selector))
mt3_government_icon.gif 파일이 불릿 이미지로 쓰임.
*/
</STYLE>
:
<!--
news라는 클래스 이름을 갖는 순서 없는 목록 요소(ul)를
사용
-->
<ul class="news">
<li>정부 보도 자료 <!-- 실제 목록들은 li 요소를 사용-->
<li>알기 쉬운 정책
:
:
</ul>

```

□ 스타일 시트를 사용하여 목록을 나타내는 방법

- 목록(list)을 나타낼 때에는 그냥 그림으로 항목들을 나열하는 것보다는 목록을 나타내는 마크업(예를 들면 순서 없는 목록인 , 순서 있는 목록인 , 용어 정의 목록인 <dl>)을 사용하는 것이 바람직함.
- 각각의 목록에 대한 불릿 이미지의 모양은 스타일 시트(style sheet)를 통해서 정의할 수 있으며,
- 스타일 시트를 통해 정의된 불릿 이미지는 대체 텍스트를 붙일 필요가 전혀 없으며, 장애인을 위한 접근성도 매우 높아지는 장점이 있음.

※ CSS(Cascading Style Sheet)란?

- ☐ HTML이 발전하면서 새로 더해진 기능으로, HTML이나 XML 문서의 구조 또는 내용과는 독립적으로 문서의 형식(문단 모양, 글꼴, 색깔, 크기, 경계선 모양, 배치, 줄 간격 등)만을 웹 개발자나 웹 사용자가 조정할 수 있도록 한 것
- ☐ CSS를 이용해 디자이너와 사용자들은 자기만의 모양을 정의해놓고, 이것을 여러 웹사이트에 적용시킬 수 있음.
- ☐ 대체적으로 HTML 내부에 문서의 모양을 나타내는 것보다 CSS를 통해 모양을 분리하면, 문서의 유지보수가 쉬워지고, 장애인을 위한 접근성도 높아짐.
- ☐ CSS에 대한 자세한 사양은
 - <http://www.w3.org/TR/CSS21/> 를 참조할 것

(5) 여러 개의 조각으로 나누어진 그림

- 여러 개의 조각 이미지란, 의미적으로는 하나의 이미지이지만 큰 그림이나 사각형이 아닌 부정형의 그림을 테이블 셀에 집어넣기 위해 두 개 이상의 이미지로 잘라 나눈 것으로,
 - 이러한 경우에는 어떤 그림에 대체 텍스트를 넣어야 할지 홈페이지 개발자의 결정이 필요함.
 - 만약 하나의 의미적인 그림(예를 들어 기관 로고)이 두 개의 파일로 나누어져 있다면,
 - 그 중에 더 중요한 것(또는 더 크기가 큰 것)에는 대체 텍스트를 넣어주고, 나머지에는 그냥 공백의 대체 텍스트를 넣어줌.

X (Bad)

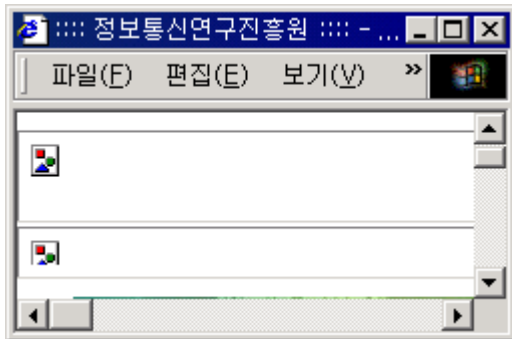
<그림 11> 여러 개의 조각으로 나누어진 그림



※ 출처 : 정보통신연구진흥원(<http://www.iita.re.kr>), 2004. 4. 11

X (Bad)

<그림 12> 대체 텍스트가 없는 조각
그림



두 개로 나누어진 이미지
모두 대체 텍스트가 붙어
있지 않아 무엇을 나타내
는 그림인지 알 수가 없
음.

```

```

:

중략

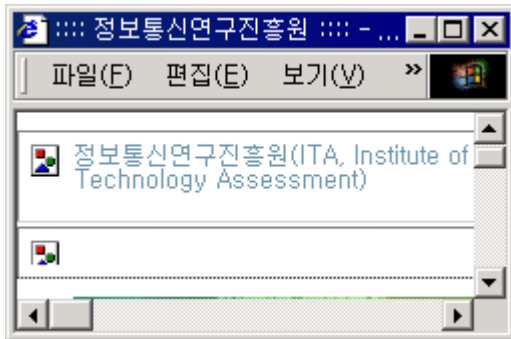
:

```

```

O (Good)

<그림 13> 대체 텍스트를 넣은 조각
그림



위에 큰 이미지에는 정상
적인 대체 텍스트를 넣고
아래 이미지에는 공백 대
체 텍스트를 넣었음.

```

```

:

중략

:

```

```

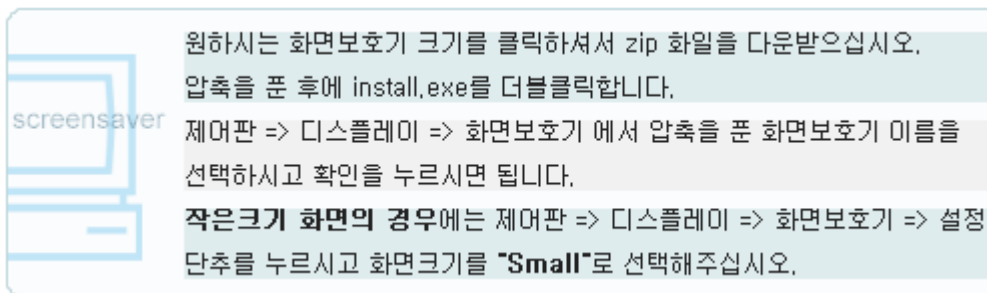
(6) 여러 개가 합쳐진 한 이미지, 이미지 맵

- 논리적으로는 여러 개로 나뉘어져야 하지만, 한 개의 파일로 크게 그려진 이미지는 (통상적으로 통 이미지로 불림) 적합한 대체 텍스트를 붙이기가 매우 어려움.

- 큰 그림의 경우 간결하게 대체 텍스트를 넣기가 쉽지 않으므로,
 - alt에는 제목만 넣고, title에 약간 자세한 설명, longdesc을 이용해 별도로 자세한 설명을 넣는 것이 바람직함.
- 그러나 그림을 영역별로 나누어 클라이언트측(client-side) 이미지 맵을 사용하는 것이 더 쉽고 정확하게 이미지가 나타내고자 하는 의미를 구현하는 방법임.
- 이미지 맵을 사용할 때에는 큰 이미지 전체에 대한 대체 텍스트와 부분 영역에 대한 대체 텍스트를 반드시 넣어주어야 함.

X (Bad)

<그림 14> 여러 내용이 합쳐진 한 개의 큰 이미지



※ 출처: 국정홍보처 (<http://www.allim.go.kr>), 2004. 4. 11

- 위의 그림처럼 여러 개가 합쳐진 한 이미지에 대해 시각장애인의 접근이 어려운 것은 다음과 같은 2가지 이유 때문에 발생함.

X (Bad)

- 대체 텍스트를 빼먹은 경우

```

```

- 대체 텍스트만으로 부족한 경우

```

```

- 여러 개가 합쳐진 한 이미지에 대해 시각장애인의 접근을 원활히 하기 위해서는 다음과 같이 대체 텍스트를 제공해야 함.

O (Good)

- 다단계 대체 텍스트를 활용한 경우
 - 가장 간단한 설명은 alt를 이용하고, 그것보다 조금 더 상세한 설명은 title을 이용하고, 아주 상세한 설명은 별도의 파일로 제작하여 longdesc에 넣어줌.

```

```

○ 이미지 맵을 활용한 경우

```


<map name="info-map">
<area shape="rect" coords="90,14,523,47"
href="nohref" alt="1단계: 원하시는 화면 보호기를
클릭하셔서 zip 파일을 다운받으십시오. 압축을 푼 후에
install.exe를 더블 클릭합니다.">
<area shape="rect" coords="90,57,523,91"
href="nohref" alt="2단계: 제어판 → 디스플레이 →
화면보호기에서 압축을 푼 화면보호기 이름을 선택하시고 확인을
누르시면 됩니다.">
:
중략
:
</map>
```

※ 이미지 맵(image map)이란?

이미지나 객체의 영역을 나누어 각 영역별로 특정한 동작을 지정할 수 있도록 한 것. 예를 들면, 두 개의 등이 있는 신호등을 하나의 그림으로 그린 다음, 위쪽 빨간색 부분에 대해서는 “멈추십시오” 라는 메시지가 나오도록 하고, 아래쪽 초록색 부분에 대해서는 “가셔도 좋습니다” 라는 메시지가 나오도록 할 수 있음.

- 클라이언트측(client-side) 이미지 맵: 사용자가 마우스로 이미지의 특정 부분을 클릭하면, (사용자의 클라이언트 PC에 있는) 브라우저가 그 좌표를 해석하여 그 영역과 연관된 동작을 수행함.
 - 일반적으로 클라이언트측 이미지 맵의 접근성이 서버측 이미지 맵보다 좋음.
- 서버측(server-side) 이미지 맵: 사용자가 마우스로 이미지의 특정 부분을 클릭하면, 서버에 있는 엔진이 좌표를 해석하여 연관된 동작을 수행함.

이미지 맵에 대한 자세한 정의는 W3C의 HTML 4.01 사양 중 이미지 맵 부분

(<http://www.w3.org/TR/html401/struct/objects.html#h-13.6>)을 참조

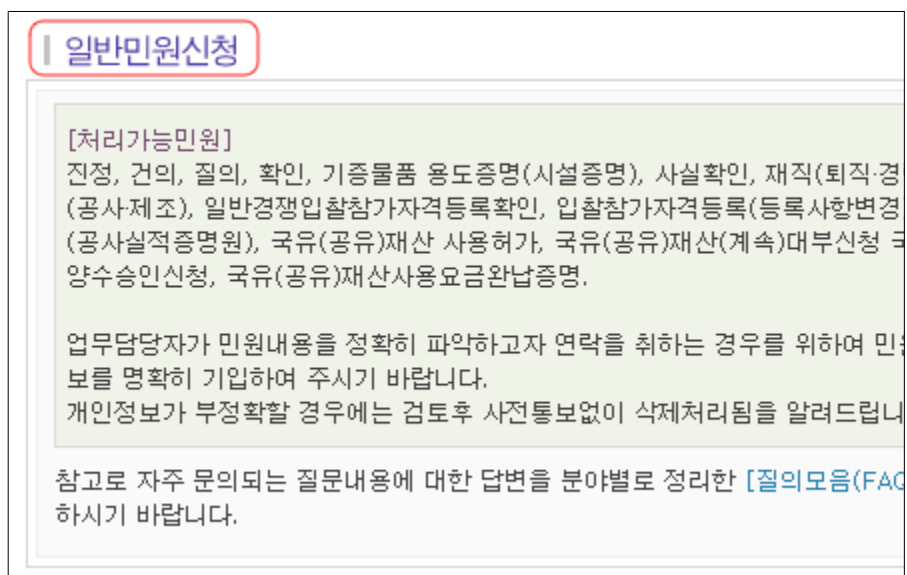
2. 주요 사례별 대체 텍스트 제공 방법

(1) 소제목 텍스트 이미지

- 소제목 이미지는 주로 텍스트로 된 콘텐츠의 제목 등을 이미지로 처리하여 사용자가 보다 쉽게 제목을 인지할 수 있도록 하기 위해 사용되는 것으로,
 - 아래 그림에서는 ‘일반 민원 신청’ 부분이 이미지로 되어 있고 그 아래에는 테이블과 텍스트로 구성되어 있음.

X (Bad)

<그림 15> 대체 텍스트가 없는 소제목 텍스트 이미지



※ 출처 : 통일부 홈페이지, 2004. 5. 8

```

```

- 이러한 사례는 우리 나라의 정부 홈페이지에서 아주 많이 찾아 볼 수 있는 것으로,
 - 제목을 이미지로 처리하는 이유는 심미적인 이유를 포함해 여러 이유가 있겠지만 무엇보다도 그 부분이 중요하고 돋보여야 하기 때문일 것임.
- 그러나 이렇게 중요한 것에 대해 대체 텍스트 속성을 지정하여 주지 않으면
 - 시각 장애인들은 알아 볼 방법이 전혀 없음.
- 그러므로, 소제목 이미지는 다음과 같이 대체 텍스트를 꼭 넣어 주어야 하며
 - 추가로, 문서 전체에서 소제목이 차지하는 역할에 걸맞게 <h1>, <h2>, <h3> 또는 <h4> 등을 넣어주는 것이 좋음.

O (Good)

```
<h2></h2>
```

(2) 공지사항 등의 more(전체보기) 링크

□ 특정한 게시판의 최근 목록 몇 개만을 뽑아 첫 페이지에 보여주는 형태는 국내 홈페이지에서 자주 활용되는 기법임.

- 이 경우 메인 페이지에서는 전체 목록 가운데 4~5개의 최근 목록만 보여주며
- “more”나 “전체목록” 등의 링크를 통해 전체 목록을 볼 수 있게 구성되어 있음.

X (Bad)

<그림 16> 대체 텍스트가 없는 전체 보기 링크

새소식		전체보기▶
▶ 제46회 9급공채 필기시험 시간 및 장소 공고	05,09	
▶ 행정자치부 e-Newsletter 제54호	05,07	
▶ 제42회 7급 공채시험 출원현황(최종)	05,07	
▶ 제49회 현충일 태극기(弔旗)달기 추진 협조	05,07	
▶ 2004년도 6월중 정보화교육 대상자 추천 의뢰 ...	05,06	
▶ 행정자치부 e-Newsletter 제53호	05,05	
▶ <공직자전용비리신고방> 운영	05,03	

※ 출처 : 행정자치부 홈페이지, 2004. 5. 9

```
<td class="first_home_center_title_back"
width="210">

```

```



</td>
<td rowspan="2" width="95" color:#959595;
font-size:11px;text-align:right;" valign="bottom">
<a href="/webapp/bbs/notice/list.action?bid=96"
style="color:#999999 !important!;">

</a>
</td>

```

- 위의 예에서 ‘**새소식**’의 대체 텍스트는 “새소식”으로 지정이 되어 있지만, ‘**전체보기**’의 대체 텍스트는 누락되어 있음.
- 우선 두 이미지 모두 대체 텍스트가 기본적으로 삽입되어야 함.
- 그러나 **more** 링크와 같은 경우 몇 가지 고려해야 할 사항이 있음.
- 아래의 예처럼 텍스트와 똑같은 대체 텍스트만을 제공하는 것으로는 부족함.

X (Bad)

이미지	대체 텍스트	접근성 문제점
	새소식	시각장애인이 새소식의 전체 목록이 있는 것으로 오인할 수 있음. 엄밀히 따지면 이 새소식이라는 것은 새소식 중 최근 몇 개만 추려놓은 최근 목록임.
	전체보기	무엇에 대한 전체 보기인지 명확하게 알기 어려우며, 특히 여러 개의 최근 목록과 more가 존재한다면 더욱 혼란을 줌.



□ 또, 아래와 같이 서술형으로 과도하게 텍스트를 제공하는 것도 오히려 혼란을 줌.

○ 일반적인 화면 낭독 프로그램에서는 이미지의 경우 “그래픽 ~”, 링크는 “링크 ~”이라는 음성 정보를 먼저 제공한 다음 alt의 내용이나 링크의 내용을 읽어주므로,



- “공지사항 링크입니다”와 같이 “링크”, “이미지” 등을 알리는 단어를 과도하게 대체 텍스트에 사용할 필요가 없음.

□ 따라서, 대체 텍스트에 “링크”, “이미지”, “Click Here” 등의 단어는 사용을 하지 않는 것이 바람직함.

X (Bad)

이미지	대체 텍스트
	새소식 최근목록 최신목록입니다.
	새소식 전체목록으로 가시려면 여기를 누르십시오.

O (Good)

이미지	대체 텍스트
	새소식 최근 목록
	새소식 전체 보기

- ☐ 위와 같이 대체 텍스트를 제공해야 시각 장애인이 최신 목록과 전체 목록에 대해 명확하게 구분하고 이해할 수 있음.

```
<td class="first_home_center_title_back"
width="210">

</td>
<td rowspan="2" width="95" color:#959595;
font-size:11px;text-align:right;" valign="bottom">
```

```
<a href="/webapp/bbs/notice/list.action?bid=96"
style="color:#999999 !important!;">

</a>
</td>
```

- 아래 그림은 여러 개의 최근 목록과 전체 보기가 한 페이지에 있는 사례로,
 - more라는 이미지가 무엇을 뜻하는지 구분하기 위해서 여러 개의 more마다 각기 다른 대체 텍스트를 붙여 주어야 함.

<그림 17> more(전체보기) 링크가 다수로 제공되는 경우

What's news

- 강릉의 숨겨진 절 보현사
- 주문받아 나르는 나룻터 주문진항
- 150만년전 용암이 만들어 놓은 장군석
- 겨울바다가 따뜻한 칠라봉 '박수기정바위'
- 우도는 영락없이 누운 소 모양이었다

more



모닥불에 캅데일 한잔 어떠세요?

모닥불에서 평이도 치고 썰매도 타고

동호인 편선

자세히

〈작은 여행자의 마을〉



Tboom Travel

more



여자가 건강해야 집안이...

▶ 여행일시 : 4.29

▶ 여행마감 : 4.20

▶ 여행내용

건강을 찾아서 떠나는 여행

[경고] 이 여행은 티봄의 순수동호회여행이지 여행상품이 아닙니다.

5/9)보물캐기 비밀여행! 자연산두루따러 떠...

Travel Column

more

2003년 6월 27일부터 9월 26일까지 3개월에 걸쳐 진행되었던 잡월드야원의 서운원 자연계 여행기 연재가 시작되었습니다.

more

여행기자들의 여행기사

more

- 오랜만에 떠난 가족 나들이
- 소록도..
- 돌과 정을 찾아 떠난 길
- 크리스마스 종합선물 세트
- 차탈기가 그윽한 수종사

이야기가 있는 체험여행

삼척 용소골 여행

삼척의 용소골 18km는 우리나라에서 가장 아름다운 3대 계곡이다. 지리산의...



- 여성조성과 함께 떠난 체험여행

※ 출처 : 티붐 홈페이지 (www.tbboom.co.kr), 2004. 5. 9

(3) 약도

- 약도와 같이 복잡한 그림의 경우에는 원래 그림의 제시 목적인 위치 설명을 위해,
 - alt, title, longdesc 속성 값을 통해 대체 텍스트를 제시해주는 것이 바람직함.
- 주의할 것은 설명이 긴 경우에 title이나 longdesc을 활용하더라도 alt 속성값은 절대 빠뜨려서는 안 됨.

X (Bad)

<그림 18> 대체 텍스트가 없는 약도 이미지



※ 출처 : 교육인적자원부 홈페이지 (<http://www.moe.go.kr>), 2004. 4. 11

```

```

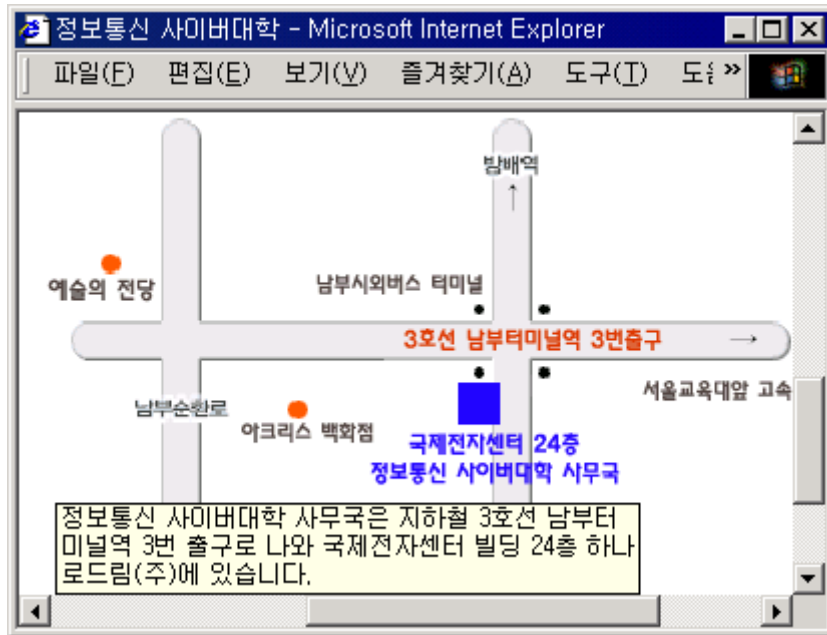
- 위의 사례에서 대체 텍스트를 제공하지 않거나 단순히 “약도 이미지”라는 대체 텍스트만을 제공하는 경우,
 - 시각장애인은 약도가 주는 원래 의미에 대해 전혀 정보에 접근할 방법이 없음.
 - 그러므로, 다음과 같이 대체 텍스트를 자세하게 제공하는 것이 바람직함.

```

```

- 만약, 약도를 설명하는 문구가 그림과는 별도의 텍스트로 제공이 된다면, 그림에서는 간단히 “교육인적자원부 약도”라고만 대체 텍스트를 표시하여도 무방함.

O (Good)



※ 출처: 정보통신 사이버대학 (<http://www.ituniv.or.kr>), 2004. 4. 8

```

```

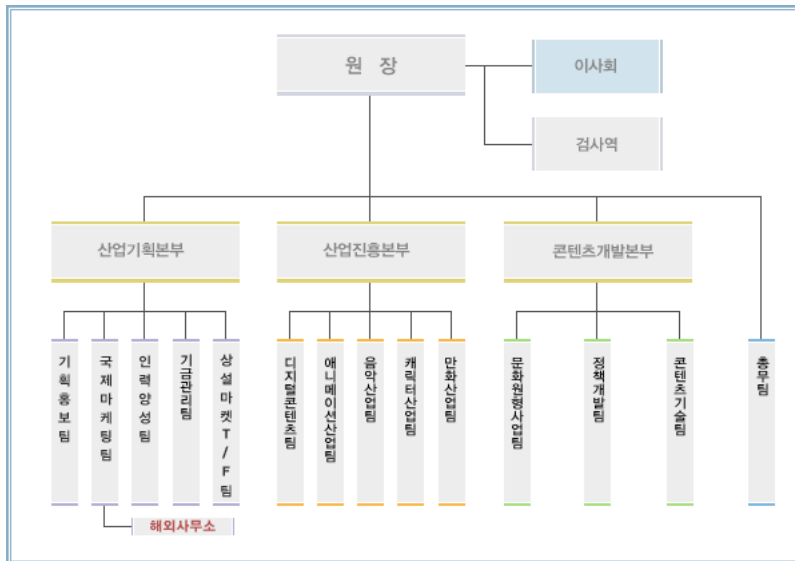
(4) 조직도

- 한 기관의 조직도나 사이트 맵은 보통 높은 단계에서 작은 단계 또는 대분류에서 소분류 등으로 위계적인 구조를 가지고 있어서

- 짧은 글로 이런 위계적인 구조들을 설명하기는 매우 복잡함.
- 그러므로, 시각장애인을 위한 대체 텍스트 제공은 `longdesc` 속성을 이용해 별도의 파일에서 자세하게 설명을 해야 함.
 - 별도의 파일은 독립적으로 완전해야 하며, 위계적인 구조를 잘 표현할 수 있는 마크업(예를 들면, ``, `<dl>`, `` 이나 `<h1>`, `<h2>`, `<h3>` 등)을 사용하는 것이 바람직함.
- `longdesc` 속성을 사용해서 별도의 파일을 만들 경우에도, `longdesc`에서 제공하는 설명에 접근하지 못할 경우가 있으므로, 원래 이미지에 대한 `alt` 속성은 빠뜨리지 말고 반드시 넣어주어야 함.
- 우리 나라 정부 홈페이지들을 조사한 결과 조직도는 대부분 조각나지 않은 하나의 이미지 위에 클라이언트측 이미지 맵을 사용하여 조직도를 링크로 제공하고 있음.

X (Bad)

<그림 19> 대체 텍스트가 없는 클라이언트측 이미지 맵으로 된 조직도



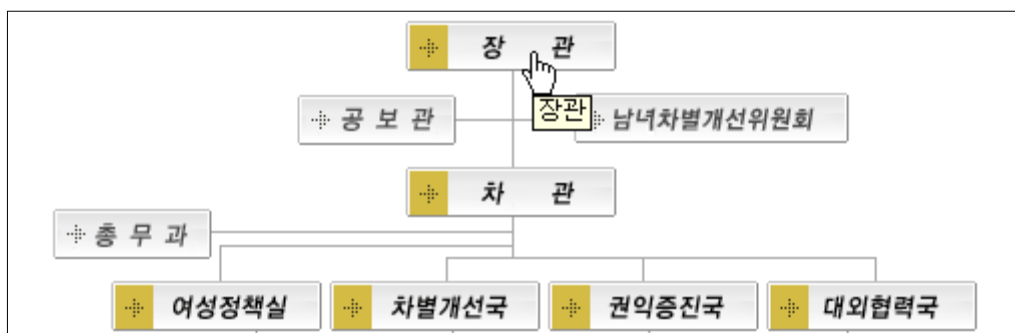
출처 : 문화콘텐츠진흥원 (<http://www.kocca.or.kr>), 2004. 4. 11

```

```

△ (Good)

<그림 20> 클라이언트측 이미지 맵을 활용한 조직도



※ 출처 : 여성부 홈페이지, 2004. 4. 11

```
<map name="FPMap0">
<area alt="장관"
href="http://www.moge.go.kr/main/open_address.html"
shape="rect" coords="233, 40, 340, 65">
</map>
```

□ 대체 텍스트를 넣는 것만으로 이러한 조직도를 시각장애인에게 완벽히 전달할 수는 없으나,

○ 대체 텍스트만이라도 모두 넣어 준다면 시각장애인은 대략적으로라도 조직의 구조를 알 수 있을 것임.

□ 대체 텍스트와 더불어 조직도에 대한 상세한 설명을 제공함으로써, 조직도가 원래 전하고자 했던 기능을 똑같이 전달할 수 있음.

O (Good)

```

```

□ alt 텍스트와 더불어 제공한 longdesc 파일은 아래와 같이

구성하는 것이 바람직함.

- 조직의 위계적인 구조를 설명하기 위해서 <h1>, <u1> 등 위계적인 마크업을 사용할 수 있음.

□ ld-org_chart.html 별도 파일 내용:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="ko">
<head>
<meta http-equiv=Content-Type content="text/html;
charset=utf-8">
<title>문화콘텐츠진흥원 조직도</title>
</head>
<body>
<h2>원장
<p>원장과는 별도로 이사회와 감사역이 있으며,
문화콘텐츠진흥원에는 다음과 같은 3개의 본부가 있다.
<h3>산업기획본부
<ul>
<li>기획홍보팀
<li>국제마케팅팀
:
중략
:
```

```
</body>
```

```
</html>
```

※ **<h1>, <h2>, <h3>** 는 글자를 크게 하는 마크업인가?

<h1>, <h2>, ... , <h6>와 같은 HTML 요소는 글자 크기를 키울 목적으로 사용되는 마크업이 아니다. 이것들은 문서 내에서 제목(heading)의 수준을 나타내주며, 지능적인 브라우저에서는 이 제목들을 자동으로 취합하여 목차를 만들어주기도 한다.

<h1>의 모양(예: 보통보다 2배 크기, 매우 두껍게, 보라색 바탕, 흰 글자, 점선 테두리, 테두리와 글자 간격은 2픽셀, 가운데 정렬 등)을 원하는 대로 바꾸고 싶다면 스타일 시트(style sheet)를 이용하면 된다.

(5) 사이트 맵

- 사이트 맵의 경우에도 조직도처럼 대체 텍스트 제공을 위한 세심한 배려가 필요함에도 불구하고
 - 많은 홈페이지의 경우에는 이를 적용하지 않고 있는 실정임.

- 전형적인 홈페이지는 아래의 예처럼 사이트의 최상위 메뉴 이미지로 구성되어 있고 하위 메뉴들은 텍스트로 되어 있음.
 - 그러나 이미지에 대체 텍스트가 붙여져 있지 않아 상위메뉴는 시각장애인이 알 수 없도록 구성되어 있음.

X (Bad)

<그림 21> 제목에 대체 텍스트가 없는 사이트맵



※ 출처 : 해양수산부 홈페이지, 2004. 5. 10

X (Bad)

<그림 22> 대체 텍스트가 잘못 지정된 예

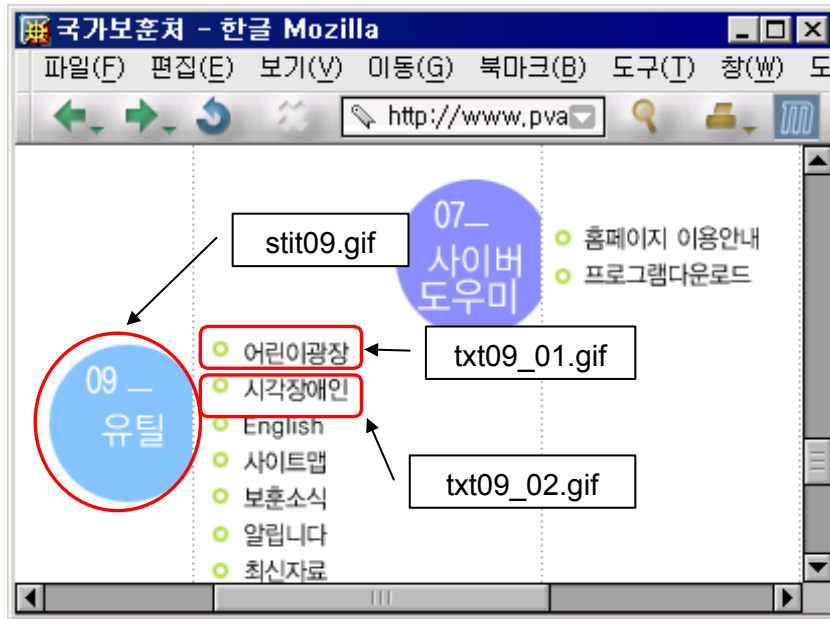


※ 출처 : 국회 홈페이지, 2004. 5. 10

- 위와 같은 경우에 소제목 이미지인 “사이트 맵”에 대체 텍스트를 제공하고 있으나,
 - 내용과 달리 ‘FAQ’라는 잘못된 대체 텍스트를 제공하고 있음.

X (Bad)

<그림 23> 대체 텍스트가 없는 그래픽 사이트 맵



※ 출처 : 국가보훈처 홈페이지 (<http://www.pvaa.go.kr>), 2004. 5. 10

X (Bad)

```

:

:

```

□ 위의 그림에서와 같은 사이트 맵은 '09_유틸' 이라는 큰 범주 밑에 '어린이 광장', '시각 장애인'과 같은 하위 페이지들이 들어가는 위계적인 구조를 가지고 있음.

- 따라서 HTML에서 위계적인 구조를 나타내는 마크업(<h1>, <h2>, <h3> 등)과 목록을 나타내는 마크업(, , 등)을 사용하여 사이트 맵을 표시하는 것이 바람직함.

O (Good)

```
<style type="text/css">
  /*
    first-level이라는 클래스를 갖는 ul 요소의 하위 요소로
    오는 li 요소에 대해 불릿 이미지를 초록색 원 그림으로
    지정한다.
  */
  ul.first-level li {list-style-image :
  url(circle.gif)}
</style>
<body>
  :
  <!--
    그림에도 두 번째 수준의 제목임을 나타내는 h2 마크업을
    넣고, 대체 텍스트를 삽입하였다.
```

```

-->

<h2></h2>
:
<!--
    first-level 이라는 클래스에 속하는 순서없는 목록을
    나타내는 ul 요소를 사용하였다.
-->

<ul class="first-level">
<li>어린이 광장
<li>시각 장애인
:
</ul>

```


※ 목록을 나타내는 세 가지 마크업

동등한 수준의 항목 여러 개가 나열된 것을 목록이라고 한다. HTML에서는 목록을 쉽게 나타낼 수 있는 구조적인 마크업을 지원하므로, 목록을 나타낼 때에 그냥 네모(□), 동그라미(○), 마이너스 기호(-)와 같은 불릿(bullet) 문자를 직접 써주는 것보다는 해당 마크업을 사용하는 것이 구조적으로 견고한 홈페이지가 되며, 접근성과 디자인의 유연성 측면에서도 유리하다.

HTML에서는 순서가 없는 목록(), 이를테면 1, 2, 3, 4와 같이 순서가 있는 목록(), 그리고 '용어 이름'과 '용어 풀이'와 같이 두 개씩 짝지어지는 정의(definition) 목록(<dl>)을 제공한다.

□ 목록 마크업의 자세한 사용법:




○ <http://www.w3.org/TR/html401/struct/lists.html>

□ 목록 마크업의 불릿 문자, 모양, 들여쓰기 등을 바꾸는 방법

○ <http://www.w3.org/TR/CSS2/generate.html#q11>

(6) 반복적 이미지

□ 흔히 쇼핑몰이나 커뮤니티에서 볼 수 있는 것으로, 해당 제품이나 해당 글에 대한 고객의 평가를 시각적으로 나타내기 위해 ★★★ 과 같이 똑같은 이미지를 반복적으로 사용한 경우가 있음.

- 또, 막대 그래프의 적절한 길이를 표시하기 위해 , , 와 같이 똑같은 이미지 박스를 반복적으로 사용한 경우도 있음.
- 이 경우에는 이미지 하나(★)에 대해 대체 텍스트를 넣어주는 것보다는,
 - 별 하나짜리 이미지(★)와 별 두 개짜리 이미지(★★), 별 세 개짜리 이미지(★★★)를 따로 제작한 다음,
 - 이러한 묶음들에 대해 “별 하나”, “별 둘”, “별 셋”과 같이 대체 텍스트를 넣어주어 등급이 어떻게 되는지 파악할 수 있도록 해주어야 함.

X (Bad)

<그림 24> 대체 텍스트가 없이 등급을 이미지로 나타낸 경우

Product Details

- ◆ **Paperback:** 900 pages
- ◆ **Publisher:** O'Reilly; 4th edition (December 15, 2001)
- ◆ **ISBN:** 0596000480
- ◆ **Product Dimensions:** 9.1 x 7.0 x 1.7 inches
- ◆ **Shipping Weight:** 2.9 pounds. ([View shipping rates and policies](#))
- ◆ **Average Customer Review:** ★★★★★ based on 200 reviews. ([Write a review](#))
- ◆ **Amazon.com Sales Rank in Books:** #2,370
(Publishers and authors: [improve your sales](#))
- ◆ **Other Editions:** [Hardcover](#) | [All Editions](#)

※ 출처: Amazon.com

(http://www.amazon.com/exec/obidos/ASIN/0596000480/qid=1110120029/sr=2-1/ref=pd_bbs_b_2_1/103-8361002-6676616), 2005. 03.06

Average Customer Review:
```

- 위와 같은 이미지는 아래와 같이 대체 텍스트를 붙여주어야 시각적으로 보이지 않더라도 등급을 파악할 수 있음.

## O (Good)

```
Average Customer Review:
```

- 물론, 이러한 대체 텍스트는 개별 항목에 대해 직접 입력해 주는 것이 아니고, 이미지 종류(즉, 등급의 종류)에 따라 자동으로 해당하는 대체 텍스트가 입력되도록 프로그램을 작성해야 함.

## (7) 광고, 캠페인

- 홈페이지에서 자주 사용되는 광고 이미지는 크게 두 가지 종류로 나눌 수 있음.
  - 쓰이는 이미지가 광고의 내용을 직접적으로 연상시키는 것

- 이미지 내용만으로는 광고의 내용이 무엇인지 모르게 하고, 클릭을 유도하는 것
- 이런 광고들에 대한 대체 텍스트는 그림에 대한 묘사나 부가 설명이 아니라 광고가 전하고자 하는 목적을 정확히 기술해주어야 함.

<그림 25> 배너 광고



※ 출처 : 한국정보문화운동협의회 홈페이지 (<http://www.icm.or.kr>), 2004. 4. 11

## **X (Bad)**

- 위와 같은 배너 광고의 경우에 광고의 기능을 대체하는 텍스트를 넣어주어야 함.
- 대체 텍스트를 빼먹으면 안 됨.

```

```

- 대체 텍스트에 그림 설명을 그대로 넣어 무슨 광고인지 알 수가 없음.

```

```

## **O (Good)**

```

```

- 또, 여러 프레임으로 된 움직이는 GIF 형식을 사용하는 경우에는 가장 대표적인 프레임의 광고 메시지를 대체 텍스트로 넣어 줄 수 있음.

## (8) 팝업창 내부

- 많은 홈페이지에서는 중요한 정보를 알리기 위한 수단으로 팝업창을 사용하고 있음.
- 국내 홈페이지의 경우 팝업창에 전체 내용 모두를 이미지로 처리한 것과 텍스트 위주로 처리한 두 가지 종류가 주로 발견됨.

## X (Bad)

<그림 26> 대체 텍스트가 없는 하나의 이미지로 된 팝업창



※ 출처 : 과학기술부 2004. 4. 21

- 대체 텍스트가 없이 하나의 이미지로 처리한 팝업은 시각장애인이 전혀 정보를 얻을 수 없음.

□ 팝업창의 과도한 사용은 바람직하지 않지만 사용할 경우에도 이미지에 대해서는 정확한 대체 텍스트를 넣어주어야 함.

○ 위 경우 시각장애인은, “제37회 과학의 날”, “오늘은 더 이상 열지 않습니다. 닫기”만 듣게 됨.

□ 따라서 아래와 같이 대체 텍스트를 넣어주어야 함.

```

```

## (9) 전화번호와 주소

□ 많은 홈페이지 하단부에는 아래와 같이 매 페이지마다 주소와 전화번호 등 연락처를 넣어 두고 있음.



## X (Bad)

<그림 27> 대체 텍스트가 없이 이미지로만 제공되는 주소와 전화 번호



※ 출처 : 보건복지부 홈페이지, 2004. 5. 10

- 그러나 대체 텍스트를 제공하지 않아 시각장애인에게는 전화번호 및 주소에 대한 정보에 접근하지 못하고 있는 실정임.
- 시각적인 깔끔함을 목적으로 텍스트 대신 이미지로 처리된 전화번호와 주소의 경우에는 반드시 적절한 대체 텍스트를 넣어주어야 함.

## O (Good)

```

```

- 특히 주소를 나타내는 마크업인 <address>를 사용하면 의미적으로 해당 부분이 주소임을 나타내줄 수 있음.

```
<address></address>
```

### III. 동영상

지침 2. 시간에 따라 변화하는 영상 매체는 해당 콘텐츠와 동기되는 대체 매체를 제공해야 한다.

- ① 오디오 정보로부터 사용자가 충분한 정보를 얻을 수 없는 모든 영상매체는 캡션을 함께 제공하여야 한다.
- ② 캡션은 영상매체나 음향매체와 동기되어야 한다.
- ③ 모든 생방송 (**real time broadcasting**)은 캡션을 실시간으로 제공하여야 한다. 단 음성이 없는 음악 방송의 경우에는 예외로 한다.
- ④ 콘텐츠는 사용자가 캡션과 영상정보를 동시에 접해야 그 내용을 이해할 수 있도록 구성되어서는 안 된다.

#### 1. 동영상과 멀티미디어 객체

□ 홈페이지 이용자들의 흥미 유발과 상호작용성(interaction)을 높이기 위해 동영상과 음향의 사용이 빈번해지고 있음.

- 동영상 및 멀티미디어 웹 콘텐츠를 제작할 경우에도 다음과 같이 접근성 환경에 대한 고려가 필요함.
  - 청각 장애인과 시각 장애인, 뇌병변 장애인 등 동영상에 부분적으로 접근이 불가능한 사람들에 대한 접근성 제공
  - 또한 동영상을 볼 수 없는 환경(예: 저속 인터넷, 무선 인터넷, 공공장소, 제한된 시스템 자원, 서로 다른 OS 등)에

## 농인 사람들을 위한 접근성 제공

- 동영상 및 멀티미디어 객체에 대한 접근성 보장을 위해 브라우저 자체적으로 표현할 수 없는 요소들에 대한 고려가 필요
  - 즉, 브라우저 자체적으로 표현되기 어려운 플래시, 동영상, 애니메이션, 애플릿 프로그램 등을 나타내기 위해서 HTML의 <object>라는 요소를 사용해야 하며,
  - 현존 브라우저에서 <object> 요소에 포함되는 모든 동영상이나 애플릿 프로그램을 정확히 보여준다는 보장이 없으므로 다음과 같이 계층적으로 대체 콘텐츠를 제공해야 함.
- 예를 들어 아래의 HTML 문서는 Python으로 작성된 우주에서 본 지구의 모습을 보여주는 프로그램을 포함하고 있는데,
  - 프로그램이 작동되지 않을 경우에, 프로그램이 작동하는 방식의 동영상(MPEG 비디오 실행)을 보여주고,
  - 동영상이 안 될 경우, 프로그램이 작동하는 방식을 애니메이션으로 처리한 GIF 그림을 보여주고,
  - 그것도 안 될 경우에는 동작 방식을 텍스트로 설명한 것을 보여주도록 한 것임.

## O (Good)

```
<P> <!-- 첫째, Python 애플릿을 실행함. -->
<OBJECT title="The Earth as seen from space"
classid="http://www.observer.mars/TheEarth.py">

 <!-- 실패할 경우, MPEG 비디오를 실행 -->
 <OBJECT data="TheEarth.mpeg"
type="application/mpeg">

 <!-- 실패할 경우 움직이는 GIF를 실행 -->
 <OBJECT data="TheEarth.gif" type="image/gif">

 <!-- 실패할 경우, 텍스트를 보여줌. -->
 The Earth as seen from space.

 </OBJECT>
 </OBJECT>
</OBJECT>
```

※ 출처: Objects, Images, and Applets, in HTML documents  
(<http://www.w3.org/TR/html401/struct/objects.html#h-13.3.1>), 2004. 12. 30

## 2. 캡션(Caption)

□ 캡션이란 동영상 비디오의 오디오 트랙 부분에 대한 텍스트

원고 가운데 비디오 및 오디오 트랙과 동기화된 것을 말하며,

- 실시간으로 나오는 텍스트 자막을 캡션이라고 생각하면 됨.

- 캡션은 일반적으로 비디오 화면 위에 겹쳐져서 시각적으로 표현되는 것이 일반적인 경우인데,

- 이를 통해 청각 장애인이나 오디오를 들을 수 없는 상황의 사람(예를 들면 시끄러운 방에 있는 사람, 듣기 능력이 떨어지는 외국인 등)들의 접근성을 높일 수 있음.

- 하지만, 아래의 사례처럼 대부분의 홈페이지에서 동영상에 대해 캡션을 제공하지 않음.

## X (Bad)

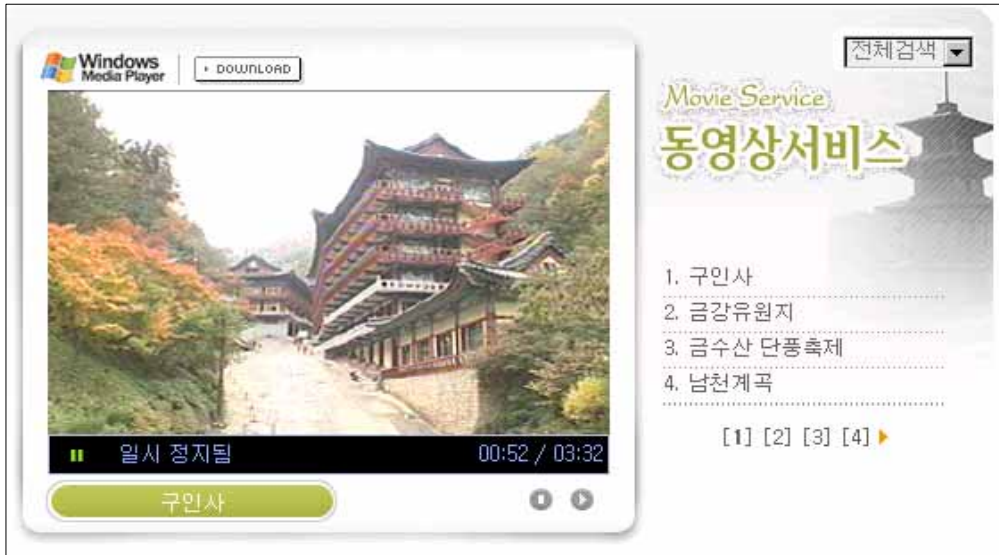
<그림 28> 캡션이나 원고가 전혀 제공되지 않는 홈페이지



※ 출처: KTV (<http://www.ktv.go.kr/>), 2004. 11. 20

## X (Bad)

<그림 29> 캡션 미제공 사례



※ 출처: 충청북도 관광

([http://www.cbtour.net/cb\\_new/cb\\_tour/kw\\_toursite\\_v?id=00185&pflag=1&num=2&tp1=A202&tp2=](http://www.cbtour.net/cb_new/cb_tour/kw_toursite_v?id=00185&pflag=1&num=2&tp1=A202&tp2=)), 2004. 11. 20

□ 캡션이 동영상과 오디오 자료에 대해 대안적으로 접근성을 높이는 기능을 제대로 하려면 다음의 세 가지 조건을 충족시켜야 함.

- 동기화(synchronized) : 오디오가 나오는 시간에 맞추어 거의 동시에 텍스트가 제공되어야 함
- 동일 내용(equivalent) : 오디오에서 제공되는 음성정보와 동일한 내용을 텍스트에도 담고 있어야 함
- 접근성(accessible) : 캡션 자체가 접근 가능해야 함



□ 캡션에는 다음과 같이 닫힌 캡션(Closed Caption)과 열린 캡션(Open Caption)이라는 2가지가 범용적으로 활용됨.

- 닫힌 캡션은 사용자의 필요에 따라 캡션을 끄거나 켤 수 (On & Off) 있는 것으로,
  - 더 발전된 형태의 닫힌 캡션은 사용자의 개인 선호도에 따라 캡션이 나타나는 위치, 글자 크기, 색깔, 속도 등도 조절할 수 있으며,
  - 시각과 청각 모두 중복적인 장애를 가진 사용자를 위해 캡션의 텍스트 내용을 별도의 데이터로 추출하여 다른 형식의 데이터로 변환이 가능함.
- 이에 반해 열린 캡션은 일반적으로 비디오에 캡션 정보가 같이 녹화되어 들어 있으므로 사용자가 임의로 캡션을 끄거나 켤 수 없는 것을 말하며,
  - 또한 캡션의 옵션을 조정하는 것이나 캡션의 내용을 텍스트로 추출하여 활용하는 것(예: 검색, 화면 낭독 프로그램을 통한 접근 등)도 불가능함.

□ 미국의 경우에는 인터넷뿐만 아니라 13인치 이상의 TV 수상기에는 의무적으로 닫힌 캡션 기능을 탑재토록 의무화하고 있으며,

- 대부분의 녹화된 프로그램의 경우 반드시 캡션 신호를 송

출하도록 법으로 규정하고 있음.

## **O (Good)**

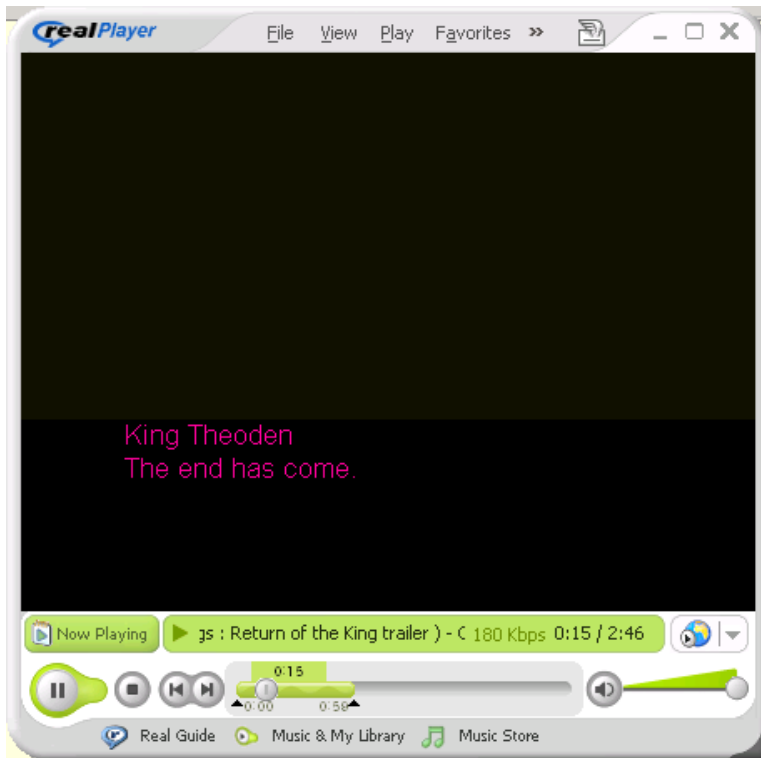
<그림 30> SMIL과 QuickTime Player을 이용하여  
단한 캡션을 구현한 예



※ 출처: web essentials 04 - zeldman keynote captioned with quicktime SMIL  
([http://www.splintered.co.uk/experiments/archives/we04\\_zeldman\\_captioned/](http://www.splintered.co.uk/experiments/archives/we04_zeldman_captioned/)),  
2004.12.31

## O (Good)

<그림 31> Real One Player에서 닫힌 캡션의 예 (On & Off 가능)



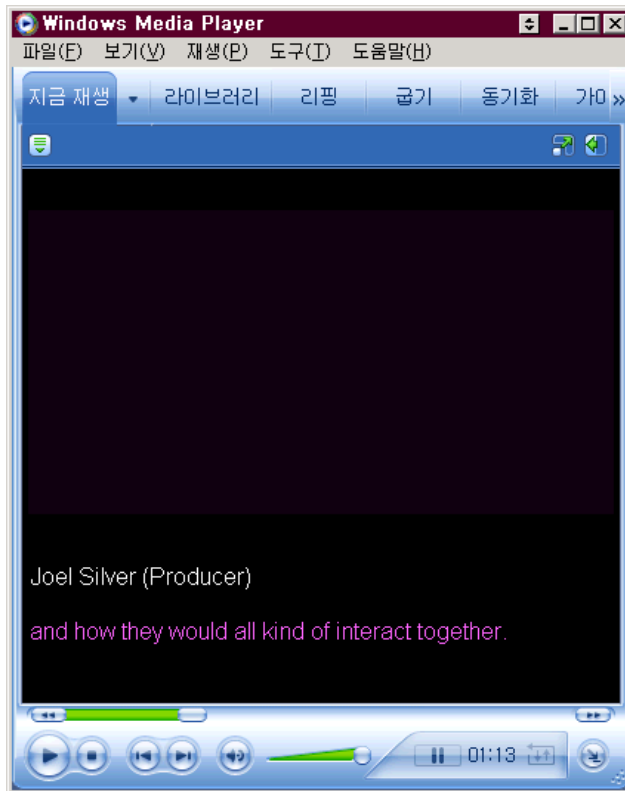
※ 출처: 반지의 제왕(왕의 귀환) 예고편 중  
(<http://www.captionkit.com/realplayer/1/10708137812265/presentation.ram>),  
2004.12.31

□ 멀티미디어 서비스를 제공시 접근성 준수방법에 대한 실사례로 사용자의 기호에 따라

- 캡션을 끄거나 켤 수 있으며, 내용을 별도로 추출하여 텍스트로 변환할 수 있어 검색 및 기타 보조기술과의 호환가능성을 높임.

## O (Good)

<그림 32> Windows Media Player에서 닫힌  
캡션의 예



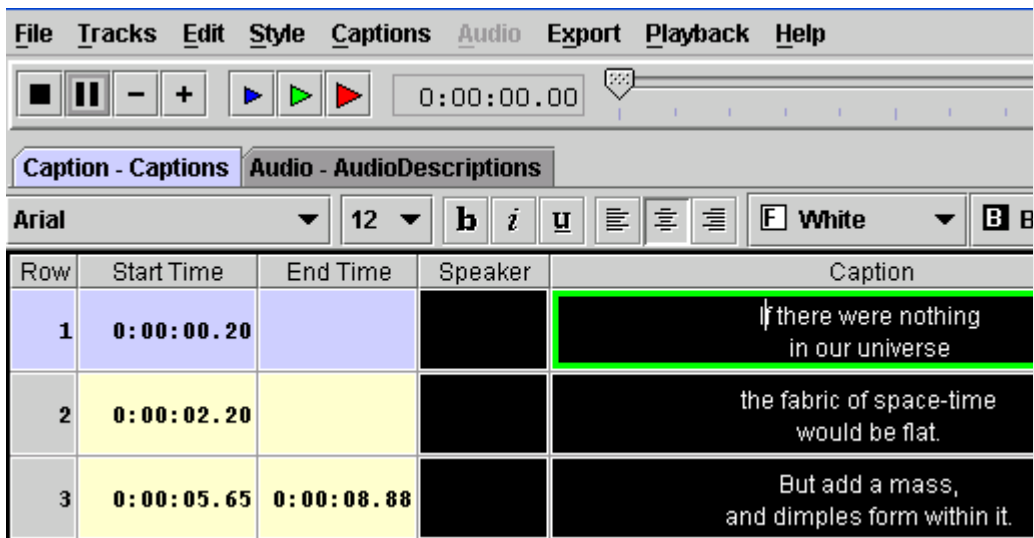
※ 출처: Inside the Matrix: Making of Matrix Reloaded,  
(<http://www.captionkit.com/mediaplayer/1/37917416024885/presentation.aspx>), 2005. 1. 2

□ 위와 같이 영화에서 닫힌 캡션을 제공함으로써 보다 많은 사람들이 정보에 접근할 수 있게 되는 것임.

- Real Player, Windows Media Player 등 다양한 제품에서 접근성을 고려한 멀티미디어 객체 제작이 가능하며,
- 저작 도구로 MAGPie, MakeSAMI 등이 있음.

## ※ 캡션 만드는 프로그램 MAGpie

<그림 33> MAGpie 프로그램을 이용한 캡션 제작 화면



- MAGpie는 미국 CPB/WGBH National Center for Accessible Media (NCAM)에서 만든 리치 미디어(rich media)용 캡션과 오디오 설명(audio description) 무료 저작 도구로,
  - 애플 킥타임 플레이어, 리얼 플레이어, 윈도우즈 미디어 플레이어 등 주요 미디어 플레이어를 모두 지원하며, 최신 버전은 플래시에 자막을 붙이는 기능도 지원함.
- 특히 W3C(World Wide Web)의 표준 멀티미디어 통합 언어인 SMIL (Synchronized Multimedia Integration Language) 과,

- Microsoft사의 독자적인 파일 형식인 SAMI (Synchronized Accessible Media Interchange) 파일을 쉽게 만들 수 있도록 도와줌.

□ 자세한 MAGpie의 사용법은 아래 홈페이지를 참조할 것

[\(http://ncam.wgbh.org/webaccess/magpie/\)](http://ncam.wgbh.org/webaccess/magpie/)

<그림 34> MAGpie를 이용해 캡션을  
넣은 비디오 화면

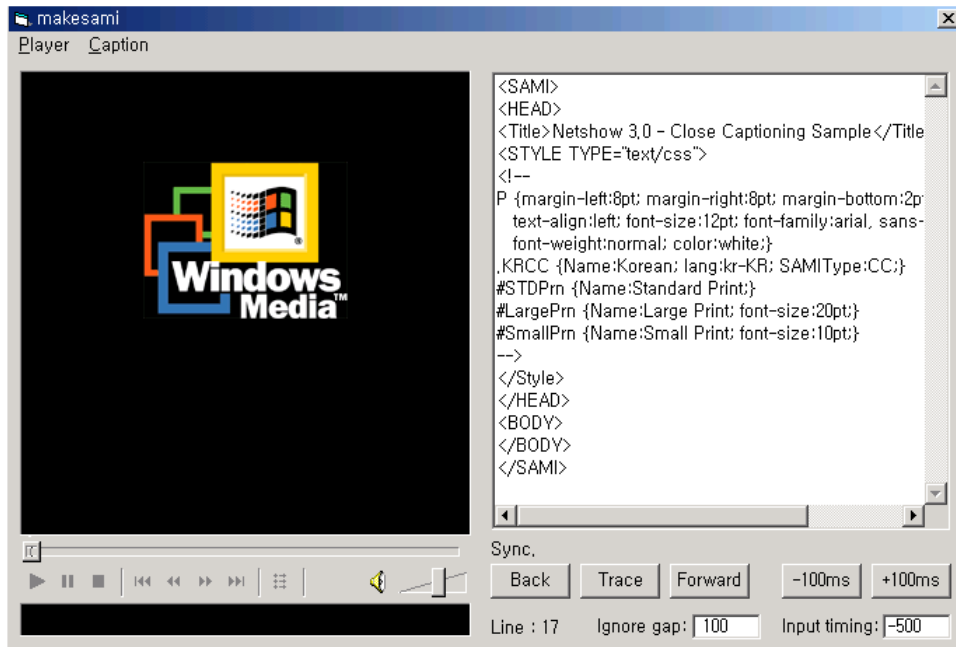


※ 출처: Keeping Web Accessibility in Mind Video  
(<http://www.webaim.org/info/asdvideo/>), 2005. 01. 02

#### ※ 캡션 만드는 프로그램 **MakeSAMI**

- MakeSAMI는 우리 나라에서 특히 많이 사용되는 Microsoft사의 Windows Media Player용 자막 파일인 SAMI 파일을 쉽게 제작하도록 도와주는 국내에서 만들어진 프로그램

<그림 35> MakeSAMI 프로그램 구동 모습



- 또한 MakeSAMI는 Real Player에서 자막에 사용되는 Real Text(.rt) 파일과 이것을 동영상 파일(.rm)과 결합시켜주는 표준 SMIL(.smi) 파일도 부가적으로 지원해 줌.

※ 자세한 내용은 캡션 월드

([http://captionworld.net/bbs/zboard.php?id=cf&page=1&sn1=&divpage=1&sn=off&ss=on&sc=on&select\\_arrange=headnum&desc=asc&no=13](http://captionworld.net/bbs/zboard.php?id=cf&page=1&sn1=&divpage=1&sn=off&ss=on&sc=on&select_arrange=headnum&desc=asc&no=13), 2005. 1. 2) 등 국내의 자막 제작 관련 홈페이지들을 검색해서 찾을 수 있음

### 3. 오디오 설명(Audio Description)

- 오디오 설명이란 캡션에 비해 덜 활용되는 기법 중의 하나

로, 시각적으로 나타나는 비디오 화면에 대해 해설자 (narrator)가 음성으로 모든 상황을 설명해주는 것을 말함.

- 오디오 설명에는 실제 비디오 상황에서 나오는 인물들의 대화가 모두 포함되며,
- 이와 더불어 해설자의 해설이 들어가서, 오디오 설명만으로 시각적인 비디오의 장면을 상상할 수 있도록 구성됨.

#### 4. 원고/대본(Transcripts)

□ 원고/대본은 비디오나 오디오의 음성 부분을 별도의 텍스트로 제공하는 것을 말하며,

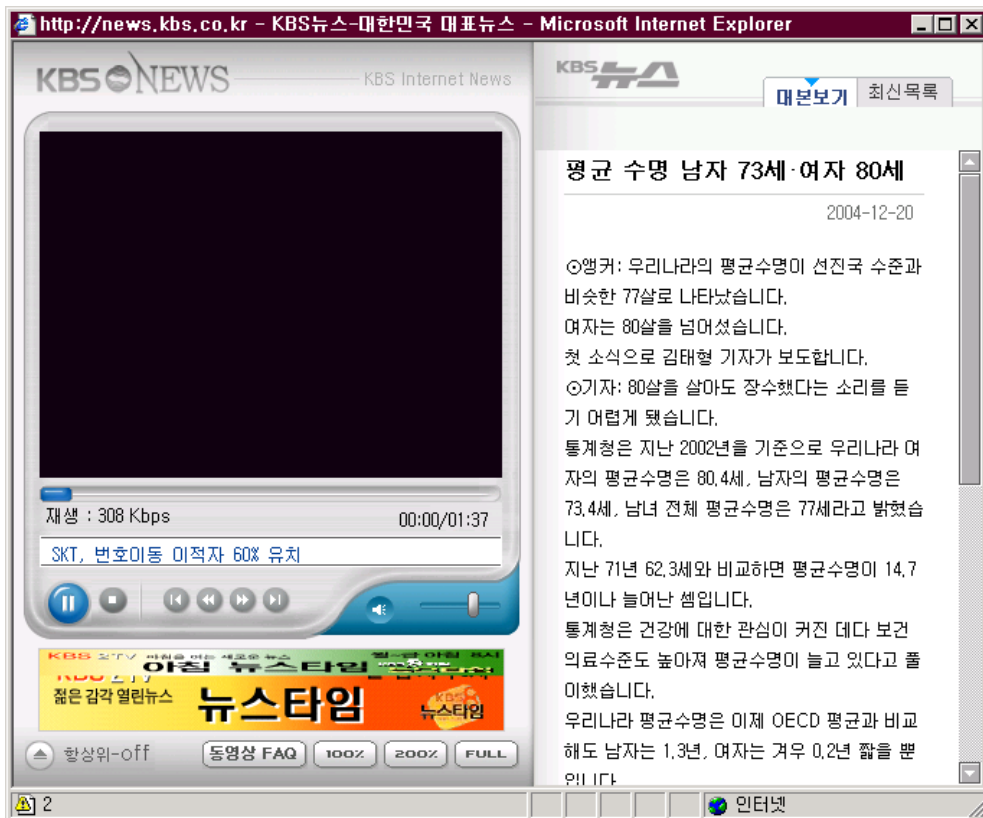
- 단순한 대사 외에 추가적인 상황 설명 등이 포함됨.

□ 캡션(특히 오픈 캡션)은 보통 비디오와 동시에 제공되어 읽는 속도가 느리거나 이해의 속도가 느린 사람들에게 접근에 제한을 줄 수도 있으므로, 원고를 같이 제공해주면 접근성을 훨씬 높일 수 있음.



## O (Good)

<그림 36> 동영상과 함께 원고(대본)를 제공한 사례



※ 출처 : KBS 뉴스 (<http://news.kbs.co.kr>), 2004. 12. 20

□ 동영상 뉴스의 경우, 캡션을 제공하는 것이 원칙임.

- 하지만, 위의 그림과 같이 캡션을 대체할 수 있는 텍스트 대본을 제공함으로써, 청각 장애인, 시각 장애인(텍스트를 점자로 변환 가능) 모두의 접근성을 높일 수 있음.

## ※ 미디어 형식의 선택

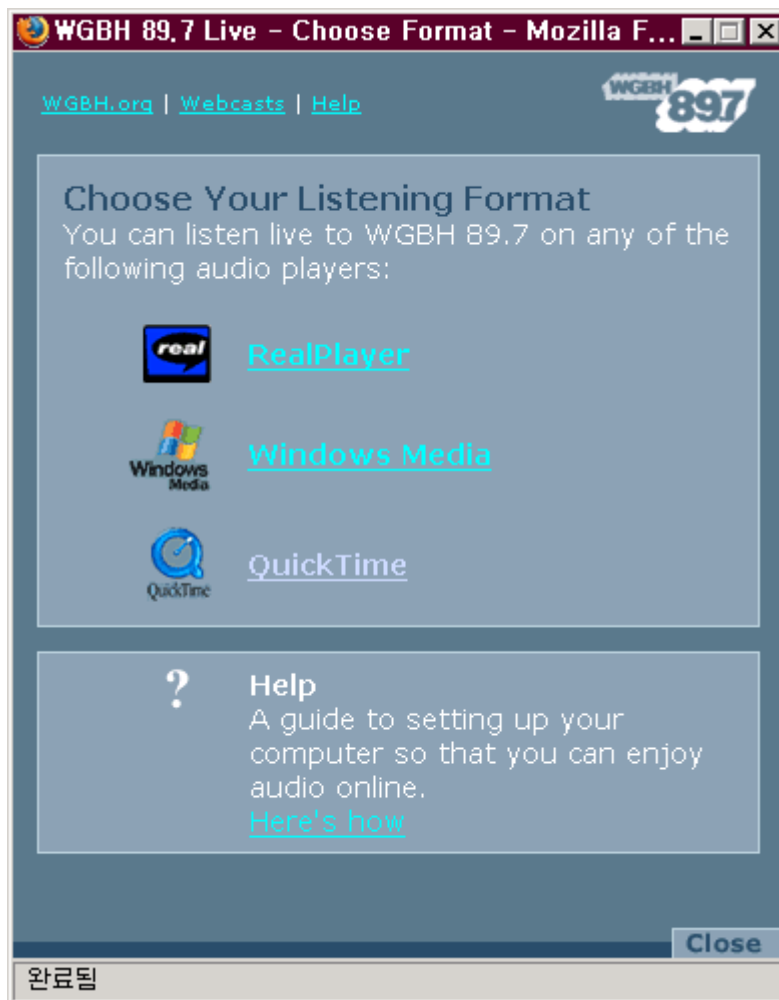
- 웹에서 동영상이나 오디오 스트리밍 서비스를 제공할 때에는 보편적인 서비스(universal service) 측면에서 되도록 많은 플랫폼에서 사용할 수 있도록 최소 두 가지 이상의 형식으로 제공하는 것이 바람직함
  - 현재 주요하게 쓰이는 Microsoft사의 Windows Media Player는 윈도우즈 플랫폼(구 버전인 6.4 제외)에서만 작동하므로,
    - 많은 사용자를 확보하고 있으나 플랫폼간 호환성은 가장 떨어지는 문제를 가지고 있음
  - Apple사의 Quick Time Player는 윈도우즈와 Mac OS 두 개의 플랫폼에서 작동하며, W3C의 표준 SMIL을 지원하지만 국내에서는 사용 빈도가 낮은 편임
  - Real Networks사의 Real Player는 윈도우즈와 Mac OS, Unix/Linux 플랫폼 등 주요 플랫폼과 W3C의 최신 SMIL 2.0을 지원하지만 국내에서는 사용 빈도가 낮은 편임
- 각 미디어 플레이어에 관한 추가 정보
  - Windows Media Player: Microsoft Windows Media (<http://www.microsoft.com/windows/windowsmedia/default.aspx>), 2005. 01. 02
  - Real Player: RealPlayer

([http://www.real.com/realplayer.html?country=US&language=EN&type=dlrhap\\_bb&pp=player&src=122604rpchoice\\_1\\_1\\_2\\_1\\_9\\_1](http://www.real.com/realplayer.html?country=US&language=EN&type=dlrhap_bb&pp=player&src=122604rpchoice_1_1_2_1_9_1)), 2005. 01. 02

- Quick Time Player: Apple - Quick Time  
(<http://www.apple.com/quicktime/>), 2005. 01. 02

## O (Good)

<그림 37> 미디어 형식을 사용자가 선택하도록 한 예



※ 출처: WGBH (<http://www.wgbh.org/>), 2005. 01. 02

- 사용자 PC, 접속 환경에 따라, 재생 가능한 미디어 형식이 제한될 수 있으므로,
  - 최소한 두 개 이상의 미디어 형식을 제공하고, 사용자가 접근 가능한 프로그램을 선택할 수 있도록 여러 개의 동영상을 제공하는 것이 바람직함.

#### ※ SMIL (Synchronized Multimedia Integration Language)

- SMIL('스마일'이라고 읽음)은 각종 웹 표준을 제정한 W3C (World Wide Web Consortium)에서 웹에서 비디오, 오디오, 텍스트와 같은 멀티미디어 자료를 효과적으로 조합하여 전달하기 위해 개발한 XML 기반의 새로운 언어로 현재 버전 2.0까지 출시되었음.
- SMIL은 XML 기반으로 되어 있으므로 일반 텍스트 편집기로 간단히 제작할 수 있으며, 처음부터 접근성을 고려하여 제작되어 몇 가지 접근성 특징을 갖추고 있음.
  - 정적인 요소에서 동적인 비디오에 이르기까지 다양한 요소에 대해 다양한 방식으로 대체 텍스트를 붙일 수 있음.
  - 사용자의 선호도와 친숙도, 장애 정도에 따라 멀티미디어 스타일과 레이아웃, 색깔, 많은 시스템 설정을 바꿀 수 있

음.

- HTML과 같은 링크와 탐색(navigation) 기능을 통해 시간에 따라 빠르게 변하는 콘텐츠라고 해도 사용자의 속도에 따라 원하는 콘텐츠로 이동할 수 있음.
- 화면 크기, 시간, 속도 등의 단위를 상대적으로 작성하면, 사용자의 환경 변화에 따라 상대적으로 프리젠테이션이 유연하게 바뀌는 장점이 있음.
- 표준 XML 문서이므로 플랫폼이나 장치에 상관없이 비교적 많은 멀티미디어 재생기에서 SMIL 표준을 지원함.

(Internet Explorer, RealOne Player, Quick Time Player, AMBULANT/X Open Source SMIL Player 등)

## IV. 색상

지침 3. 콘텐츠가 제공하는 모든 정보는 색상을 배제하더라도 인지할 수 있도록 구성되어야 한다.

① 콘텐츠가 제공하는 텍스트나 그래픽 정보는 색상을 제거하더라도 그 내용을 인지할 수 있어야 한다.

② 웹 페이지에서 보여주는 정보와 배경색 간에는 충분히 대비가 되어야 한다. 특히 웹 페이지의 내용을 고대비(**high contrast**) 모드로 표시했을 때에도 충분히 명암대비가 되도록 콘텐츠를 설계하여야 한다.

### 1. 색상을 이용한 정보 구분

□ 지침 3이 웹 콘텐츠 제작자의 색상 선택에 대한 제한 및 색상을 이용해 강조를 하지 말라는 것은 아니며,

○ 오직 색상으로만 정보를 전달하는 형태를 피하라는 것임.

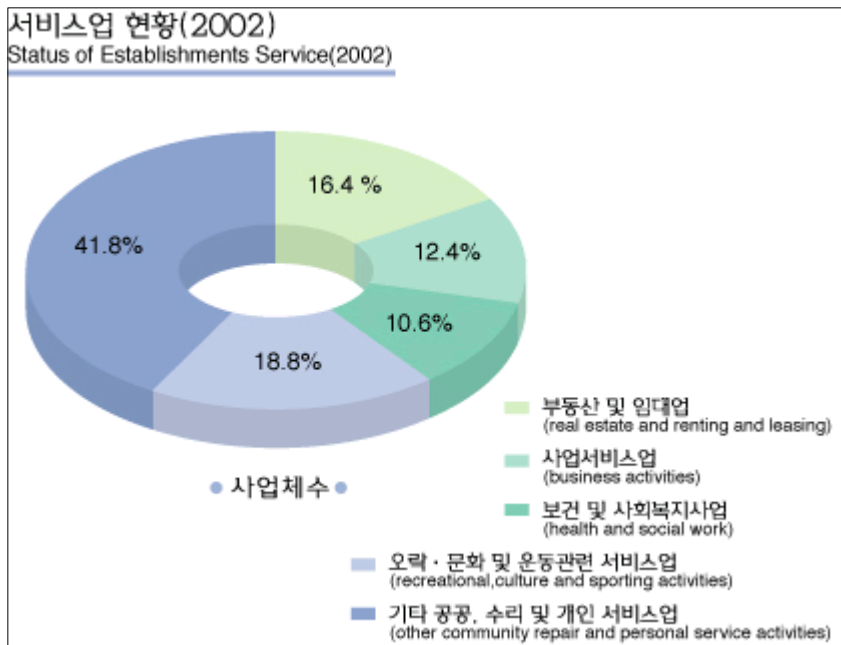
□ 웹 콘텐츠 제작자가 색상으로만 정보를 전달하려 한다면,

○ 특정한 색상을 구별할 수 없는 사용자(색맹, 색약 등)들과 흑백 디스플레이 사용자(PDA, 휴대용 웹 표시장치), 흑백 인쇄물을 보는 사용자에게는 무용지물이 될 우려가 있음

- 따라서, 색상으로만 정보를 구분하기보다는 색상 이외의 것  
-- 맥락, 마크업, 언어적인 설명, 구분 기호 등 -- 을 통해서  
도 정보가 구분되도록 웹 콘텐츠를 제작해야 함.

## X (Bad)

<그림 38> 색깔만으로 항목을 구분하기가 힘든 예



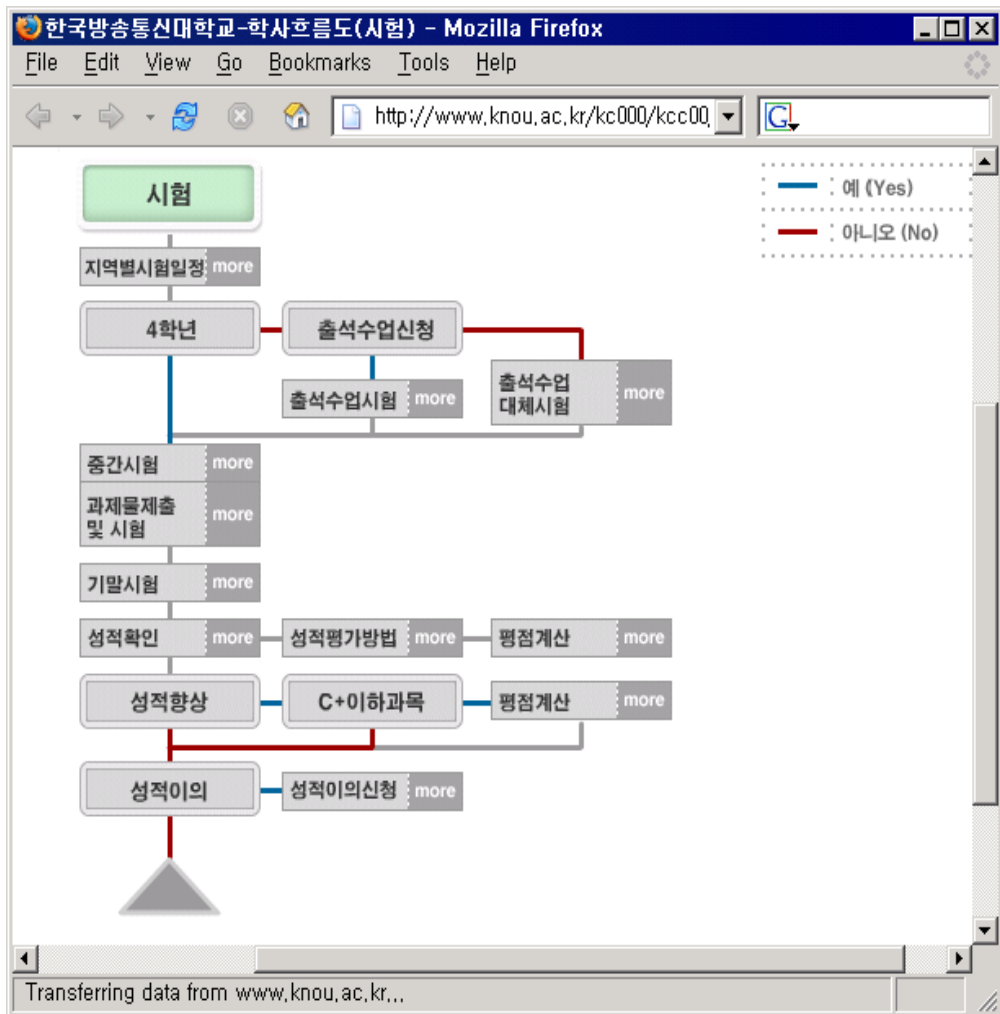
※ 출처 : 통계청

([http://www.nso.go.kr/newcms/graph/graph\\_view.html?content\\_id=%2051](http://www.nso.go.kr/newcms/graph/graph_view.html?content_id=%2051)), 2004. 8. 9

- 위의 사례에서처럼 색상만으로 정보를 제공함으로 인해 색상을 구분하지 못하면 서비스업 현황을 파악할 방법이 없음

***X (Bad)***

<그림 39> 색깔로만 ‘예’와 ‘아니오’를 구분한 사례



※ 출처: 한국방송통신대학교 ([http://www.knou.ac.kr/kc000/kcc00/kcc00\\_05.htm](http://www.knou.ac.kr/kc000/kcc00/kcc00_05.htm)),  
2004. 8. 10

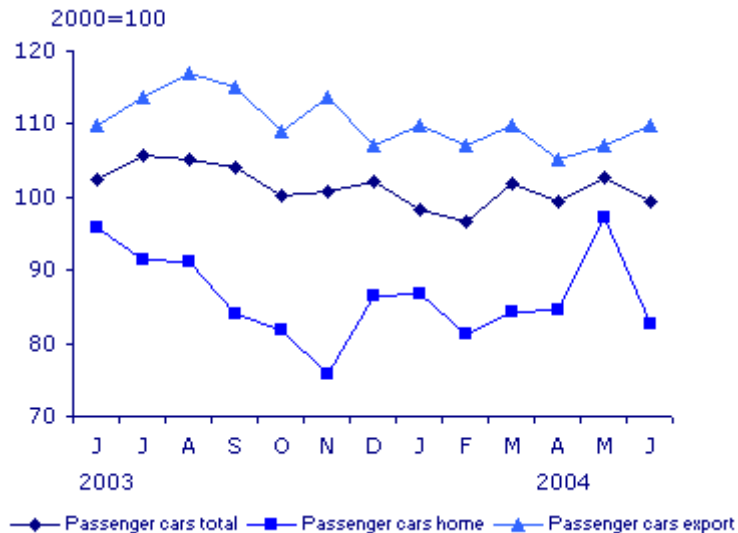
- ☐ 위의 그림에서는 흐름도(flow chart)에서 ‘예’와 ‘아니오’를 구분하는 선이 파랑색과 빨강색으로 표시되어 있어,



- 색깔 이외의 다른 정보로는 '예'와 '아니오' 선을 구분할 수가 없음.
- 또한, 위 그림을 흑백 프린터로 인쇄했을 때에도 흐름도의 '예'와 '아니오'를 구분할 수 없음.

## △ (Good)

<그림 40> 계열들을 색과 모양으로 구분한 그래프



※ 출처: 영국 National Statistics Online

(<http://www.statistics.gov.uk/cci/nugget.asp?id=376>), 2004. 8. 9

□ 위의 사례에서는 색상과 모양으로 상이한 정보를 제공하고 있으나, 색상과 모양이 유사하므로,

- 서로 다른 그래프의 계열들을 구분하기 어려움.
- 따라서, 선의 무늬(점선, 실선)를 추가로 다르게 표시한다거나, 그래프 위에 직접 계열명(Passenger cars total, Passenger cars home, Passenger cars export)을 표시해주는 것이 바람직함.

## X (Bad)

<그림 41> 색깔로만 버스 종류를 구분해놓은 사례

<div> <span>서울·강남 방면</span> <span>리무진일반</span> <span>리무진고급</span> <span> 시내버스</span> <span> 시외직행</span> </div>								
노선명	정차장번호	경유지	첫차	막차	운행간격(분)	소요시간(분)	요금(원)	운송회사 문의전화
가락시장 (609번)	5A, 11B	인천공항, 구반포, 강남버스터미널, 논현역, 강남역, 양재역, 도곡동, 대치동, 개포동, 삼성병원, 가락시장, 웰미리아파트, 수서역, 일원동	가락시장행 05:45	가락시장행 23:00	10~15	80~90	7,000	서울버스(주) 02-577-1343~5
			공항행 04:15	공항행 21:00				

※ 출처: 인천국제공항 (<http://www.airport.or.kr/guide/transportation/bus.jsp>),  
2004.10.12

- 위의 그림에서는 버스의 종류(리무진 일반, 리무진 고급, 시내 버스, 시외 직행)를 오로지 색깔로만 구분해 놓아,
  - 일반인들도 버스 종류를 구별하기 힘들고,
  - 색깔 정보를 볼 수 없는 시각장애인들은 전혀 버스의 종류를 알 수가 없음.

## X (Bad)

<그림 42> 필수 입력 필드를 색깔로만 표시한 잘못된 경우

Tell us who you are. (required fields in red)

Company:

Salutation:

First name:

Last name:

Job title:

Phone:  e.g.,

Web address:

E-mail:

※ 출처: JimThatcher.com (<http://www.jimthatcher.com/webcourse7.htm>), 2004. 10.

3

- 위의 그림에서와 같이 색깔(빨간색)로만 필수 입력 필드를 표시할 경우에는
  - 색깔을 인식할 수 없는 사용자들은 어떤 것이 필수 입력 필드인지 알 수가 없음.

## O (Good)

<그림 43> 필수 입력 필드를 색깔과 문자로 동시에 표시한 사례

Tell us who you are. (\* on **required fields**)

Company:

Salutation:

\* **First name:**

\* **Last name:**

Job title:

Phone:  e.g.,

Web address:

\* **E-mail**

※ 출처: JimThatcher.com (<http://www.jimthatcher.com/webcourse7.htm>), 2004.10.

3

□ 위의 그림에서는 다른 필드와 다르게 필수 입력 필드를 색깔(빨간색)과 특수 문자(\*)로 동시에 구분해 주어서

- 색깔을 지각할 수 없는 경우(색각이상자, 흑백 디스플레이 사용자, 화면 낭독 프로그램이나 점자 표시 장치를 사용하는 시각 장애인 등)에도 어떤 것이 필수 입력 필드인지 알 수 있음.

## X (Bad)

<그림 44> 글자의 굵기만으로 필수 기입사항을 표시한 사례

진한 글씨는 필수 기입사항입니다.

회원 아이디	<input type="text"/>	중복 확인
패스워드	<input type="text"/>	확인 <input type="text"/>
회원 이름	<input type="text"/>	
메일 주소	<input type="text"/>	
주민등록번호	<input type="text"/> - <input type="text"/>	
기본(자택) 우편번호	<input type="text"/> - <input type="text"/>	우편번호검색
기본(자택) 주소	<input type="text"/>	
기본(자택) 전화번호	<input type="text"/> - <input type="text"/> - <input type="text"/>	
회사이름	<input type="text"/>	
핸드폰 번호	<input type="text"/> - <input type="text"/> - <input type="text"/>	

※ 출처: KIEC 한국전자거래진흥원

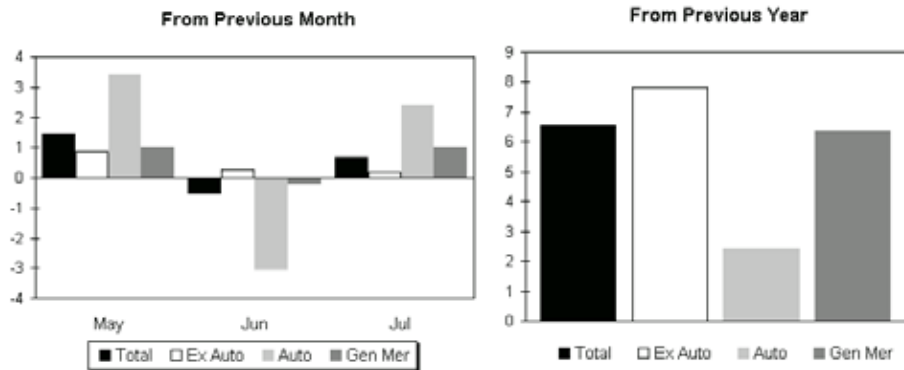
([http://www.kiec.or.kr/common/html/kiec\\_frm.asp](http://www.kiec.or.kr/common/html/kiec_frm.asp)), 2004. 11. 29

□ 위의 그림에서는 필수 입력 필드를 굵은 글자체로 표시하였는데,

- 색깔로 정보를 구분하는 것이 좋지 않은 것과 마찬가지로 글자의 굵기만으로 필수 기입 사항을 구분하는 것도 좋지 않은 방법임.
- 이 경우, 비시각적(예를 들면 화면 낭독 프로그램이 읽어서 알 수 있게)으로도 구분이 가능하도록 다른 구분 표시(예: ‘\*’, ‘필수’, ‘required’ 등)를 추가할 필요가 있음.

## △ (Good)

<그림 45> 명도(brightness)로 그래프의 막대를 구분한 사례



※ 출처: 미국 US Census Bureau (<http://www.census.gov/svsd/www/fullpub.html>), 2004.8.9

□ 위의 그림에서는, 막대들이 색깔이 아닌 무채색(흰색, 검정색, 회색 등)의 명도(brightness)로만 구분됨.

○ 다행히 그래프 내에서 계열들의 나열 순서(좌에서 우로)가 아래의 계열명(legend) 나열 순서와 일치하여 이해하기 쉬운 편이지만,

○ 그래프 위에 바로 계열명(Total, Ex Auto, Auto, Gen Mer)을 표시했으면 더 이해가 용이했을 것임.

## O (Good)

<그림 46> 색깔과 색이름을 같이 제시한 사례



※ 출처: 즐거운 학교 ([http://www.njoyschool.net/news/article\\_View.asp?no=336](http://www.njoyschool.net/news/article_View.asp?no=336)),  
2004. 8.21

□ 위의 그림에서는, 색깔을 고를 때에 고르고자 하는 색깔 원과 바로 같은 위치에 색이름이 제시되어,

- 색깔을 지각할 수 없는 경우에도 색 이름을 통해서 원하는 색을 선택할 수 있음.



## X (Bad)

<그림 47> 색깔만으로 필수 입력 항목을 표시한 예

※ 회원기본정보입력사항 (필수입력항목입니다)	
· 아이디	<input type="text"/> <input type="button" value="아이디 중복체크"/>
· 비밀번호	<input type="password"/>
· 비밀번호 확인	<input type="password"/>
· 이 름	<input type="text"/>
· 주민번호	<input type="text"/> - <input type="text"/>
· 이메일	<input type="text"/> @ <input type="text"/> ::email주소를 선택
· 우편번호	<input type="text"/> <input type="button" value="우편번호 찾기"/>
· 주소1	<input type="text"/>
· 주소2	<input type="text"/>
· 자택전화	<input type="text"/> - <input type="text"/> - <input type="text"/>
· 회사전화	<input type="text"/> - <input type="text"/> - <input type="text"/> 예) 02-777-7777
· 핸드폰	<input type="text"/> -선택 - <input type="text"/> - <input type="text"/> 예) 017-777

※ 출처: 한성고시학원 (<http://www.hsgosi.co.kr>), 2004. 11. 20

- 위 사례의 경우 화면 낭독 프로그램을 이용하는 장애인 또는 흑백 디스플레이를 사용하는 사용자는 어떤 항목이 필수 입력 항목인지 알 수 없게 됨으로,
  - 따라서 명확히 구분될 수 있는 기호를 병행하여 사용하는 것이 바람직함.

## O (Good)

<그림 48> 접근성을 고려해 필수 항목을 표시한 사례

※ 회원기본정보입력사항 (* 표시된 항목은 필수 입력 항목입니다)	
* 아이디	<input type="text"/> <span>아이디 중복체크</span> * 영문,숫자 포함
* 비밀번호	<input type="password"/> * 영문,숫자 포함
* 비밀번호확인	<input type="password"/>
* 이름	<input type="text"/>
* 주민번호	<input type="text"/> - <input type="text"/>
* 이메일	<input type="text"/> @ <span>email주소를 선택해 주세요</span>

- 위의 그림에서는, 필수 입력 항목 앞에 특수 문자(\*)를 동시에 제공하여 색상을 구분하기 힘든 사용자도 필수 입력 항목을 쉽게 알 수 있음.

## 2. 정보 전달을 위한 올바른 색의 사용

- 색깔만으로 정보를 제공하는 웹 콘텐츠를 제작해서는 안되며,
  - 접근성을 준수한 정보전달을 위해서는 선 모양(실선, 점선, 두께 등)이나 도형 내부의 패턴(빗금, 세로 무늬, 가로 무늬, 점 등)으로 같이 구분해주는 것이 중요함

□ 하지만, 어쩔 수 없이 색깔로 정보를 구분해야 하는 현미경 사진, 의학적인 사진, 지리에 따라 달라지는 기상 정보 등을 표시해야 하는 경우에는,

○ 매우 세심한 색깔의 선택이 반드시 필요함.

□ 웹에서는 심미적인 목적의 일반적인 그림이 아니고, 정보 전달을 목적으로 하는 문자, 일러스트레이션, 메뉴, 버튼, 차트, 지도, 그래프 등의 경우에는

○ 전경색(foreground)과 배경색(background)이 보다 명확히 구분될 수 있도록 배색에도 신경을 써야 함

□ 피해야 하는 색 조합

○ 명도 대비가 낮은 색 조합 : 노란색과 회색, 짙은 회색과 검정색, 짙은 파란색과 검정색, 흰색과 노란색, 빨강색과 검정색, 짙은 빨강색과 검정색, 빨강색과 짙은 파랑색 등

○ 녹색과 빨강색 : 가장 혼한 색각이상인 적록 색각 이상자들에게는 녹색과 빨강색의 구별이 잘 안 되므로, 빨강색 대신 자홍색이나 보라색을 사용해야 함

## ※ 색각이상이란?

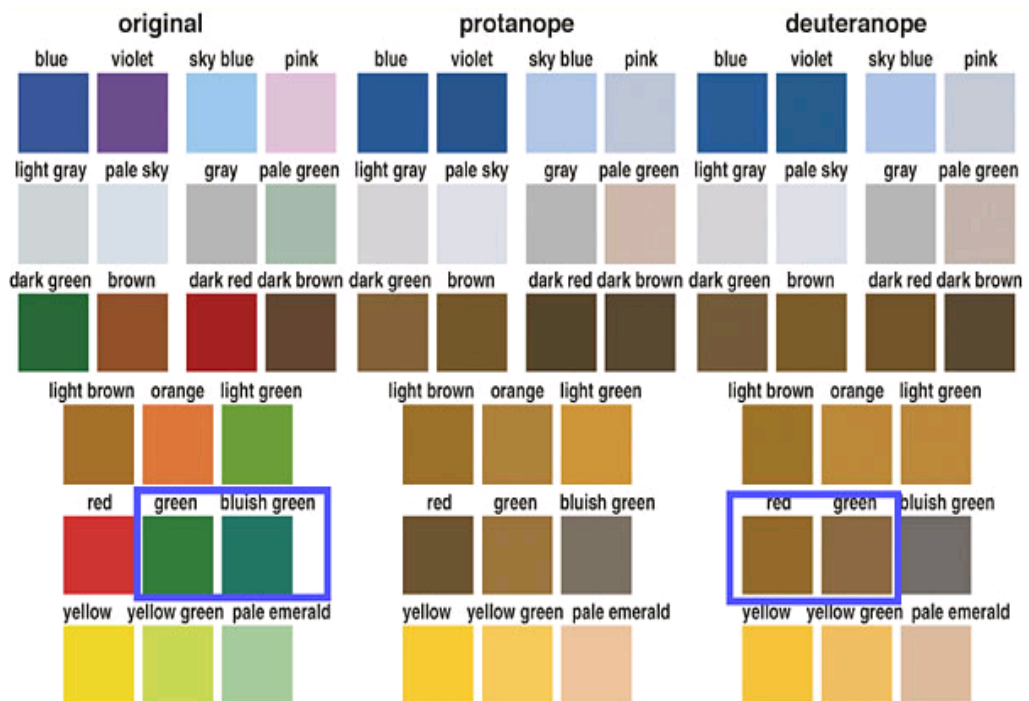
- 흔히 통칭하여 색맹이라고 부르는 용어는 사실 색각이상의 일부를 뜻하는 말이며,
  - 색각이상자라고 해서 모든 색을 보지 못하는 것은 아님.
  - 색각이상은 우리 눈의 추상체(cones)라 불리는 세포에서 특정 파장에 반응하는 색소가 없어서 일부 색을 구별하지 못하는 것으로,
    - 어떤 파장의 색소에 이상이 있느냐에 따라  
1색형색각(protanopia), 2색형색각(deteranopia),  
3색형색각(tritanopia)으로 나뉨.
- 색각이상은 성염색체를 통해 열성 유전되어 여성보다 남성에게서 더 많이 발견되며,
  - 코카서스 인종의 8%, 아시아 인종의 5%, 아프리카 인종의 4% 남성들이 적록 색각 이상을 가지고 있어
  - AB 혈액형의 발생 빈도보다 더 높음.
  - 가장 흔한 것이 빨강과 녹색을 구별하지 못하는 적록 색각이상이며, 매우 드물지만 파랑색과 노랑색을 구별하지 못하는 황청색각이상과 추상체 전체에 이상이 있어 색깔을 거의 구별하지 못하는 전색맹(전색약) 등이 있음.

<그림 49> 선천성 색각이상의 분류

### 선천성 색각이상의 분류

1색형색각	전색맹, 전색약		
2색형색각	적색맹 녹색맹	} 적록색맹	} 적록색각이상
	청황(황청)색맹		
이상3색형색각	적색약 녹색약	} 적록색약	
	청황(황청)색약		

<그림 50> 일반인, 1색각이상, 2색각이상에게 보이는 색상 시뮬레이션



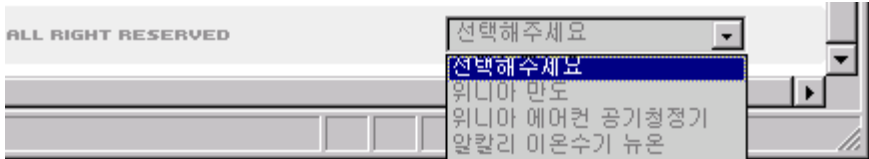
※ 출처: Barrier-free presentation for color blinds (<http://jfly.iam.u-tokyo.ac.jp/color/>), 2004. 8. 22

□ <그림 50>에서는 일반인과 1색형색각, 2색형색각에게 보이는 색상을 시뮬레이션 한 것임.

- 위의 세 줄은 색각이상자들에게 매우 유사하게 지각되어 피해야 하는 색 조합을 나타낸 것임.
- 즉, 대비되는 두 색깔이 초록, 빨강 요소만 차이가 있는 경우에 색각이상자들에게는 비슷하게 지각이 됨.
  - 예를 들면, 파랑색과 보라색(파랑 + 빨강)은 색각이상자들에게 구분하기 힘든 색깔 조합이 됨.
- 아래의 세 줄에는 세 개가 한 조가 된 색깔 조합이 나타나 있는데 일반인들이 대비를 뚜렷이 지각하는 색깔의 조합과 그렇지 않은 색깔의 조합이 색각이상자들에게는 거꾸로 지각되기도 하는 예를 보여주고 있음.
  - 예를 들면, 빨강, 초록, 청록색이 있을 때, 일반인들에게는 빨강과 초록은 잘 구별되고, 초록과 청록은 잘 구별이 안 되는 반면,
  - 색각이상자들에게는 빨강과 초록은 잘 구별되지 않고, 녹색과 청록은 확실하게 구분이 되어 보임.
  - 그 이유는 파랑색 파장에 반응하는 추상체 세포가 빨강, 초록 세포의 기능 저하에 대한 보상으로 더 민감해졌기 때문임.

## X (Bad)

<그림 51> 전경과 배경이 대비가 잘 되지 않아 가독성이 낮은 사례



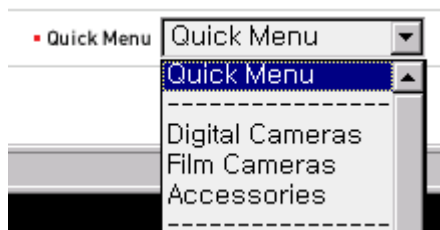
※ 출처: 2005년 발효과학 님채 (<http://www.dimchae.co.kr>), 2004.11.20

- 해당 웹사이트 하단에 있는 킥 메뉴를 보면 배경색과 글자 색이 유사한 색상으로 이루어져 있어 가독성을 떨어뜨려,
  - 색약자가 구분할 수 없으므로, “선택해주세요”, “위니아만도” 등의 목록을 나타내는 글자색을 진하게 수정해야 함.

## O (Good)

- 아래는 유사한 킥 메뉴를 사용하는 다른 홈페이지로 글자색을 검정색으로 처리하여 가독성을 높인 사례임.

<그림 52> 색상 대비가 잘 된 목록 상자



※ 출처: 삼성 카메라(<http://www.samsungcamera.com>), 2004. 11. 20

## V. 이미지 맵

지침 4. 이미지 맵 기법이 필요할 경우에는 클라이언트측 이미지 맵을 사용하거나 서버측 이미지 맵을 사용할 경우에는 동일한 기능을 하는 텍스트로 구성된 대체 콘텐츠를 제공해야 한다.

- ① 구성하려는 이미지 맵의 형태를 클라이언트측에서 기하학적으로 표현하기 어려운 경우가 아니면 서버측 이미지 맵을 사용하지 않는다.
- ② 서버측 이미지 맵 기법을 사용할 경우에는 이미지 맵의 영역에 대응하는 텍스트 링크를 별도로 제공해야 한다.
- ③ 클라이언트측 이미지 맵을 사용할 경우에는 이미지 맵의 영역에 대한 충분한 대체 텍스트를 제공하여야 한다.

### 1. 이미지 맵이란?

- 이미지 맵이란 한 개의 이미지의 일부에 한 개 또는 여러 개의 링크를 두는 기법을 말하는 것으로,
  - 용어가 지칭하는 것과는 달리 실제 맵(지도)은 아니지만,
  - 예컨대 흔히 볼 수 있는 지도 찾기의 그림에서 특정 지역



을 클릭하면 해당 지역의 정보가 있는 곳으로 이동하는 것이 그 사례임.

□ 사용자가 이미지 위에서 특정 영역을 선택(또는 클릭)했을 때,

- 그 해석(즉 처리)을 클라이언트(브라우저)에서 하느냐 서버에서 하느냐에 따라 클라이언트측 이미지 맵과 서버측 이미지맵으로 구분됨.

- 클라이언트측(client-side) 이미지 맵 : 클라이언트측 이미지 맵의 한 영역을 마우스로 활성화시키면, 브라우저가 그 좌표를 해석하며, 그 좌표가 포함된 영역과 연계된 링크를 선택하고 그것을 따라감.

- 서버측(server-side) 이미지 맵 : 서버측 이미지 맵의 한 영역을 마우스로 활성화시키면, `<a href=>`에서 지정한 서버 프로그램에 클릭한 곳의 좌표가 보내지고, 서버에 있는 프로그램이 특정한 동작을 수행함.

## 2. 클라이언트측 이미지 맵

□ 클라이언트측 이미지 맵은 일반적인 그림에 alt 속성을 이용한 대체 텍스트를 제공할 경우(지침 1 참고),

- 큰 어려움 없이 시각 장애인도 이용이 가능함.

□ `<area>` 요소를 사용한 예 : 이미지 전체와 각각의 영역

(area)에 대해 모두 alt 속성을 지정해야 함.

```
<P><IMG src="navbar1.gif" usemap="#map1" alt="메뉴
막대"></p>

<MAP name="map1">
 <AREA href="guide.html"
 alt="접속 지침"
 shape="rect"
 coords="0,0,118,28">
 <AREA href="search.html"
 alt="검색"
 shape="rect"
 coords="184,0,276,28">
 <AREA href="shortcut.html"
 alt="바로 가기"
 shape="circle"
 coords="184,200,60">
</MAP>
```

- 텍스트 링크를 같이 제공한 예 : <area> 대신에 <a>를 사용하면, 이미지 맵과 함께 별도의 텍스트 링크 모음이 제공되어 접근성이 더 높아짐

- 그러나 일부 브라우저(인터넷 익스플로러)에서 아직 지원하지 못함.

```
<p></p>
<MAP name="map1">
 <P>사이트 탐색
 <A href="guide.html" shape="rect"
coords="0,0,118,28">접속 지침 |
 <A href="shortcut.html" shape="rect"
coords="118,0,184,28">바로 가기 |
 <A href="search.html" shape="circle"
coords="184,200,60">검색 |
</MAP>
```

### 3. 서버측 이미지 맵

- 서버측 이미지 맵은 그림을 분할하는 영역의 모양이 너무 복잡해 불가피한 경우가 아니라면 사용하지 않는 것이 좋으며,
- 서버측 이미지 맵을 사용할 경우, 마우스와 같은 포인팅 장치가 아닌 입력 도구(예: 키보드)를 사용할 수 없어, 접근성이 떨어짐.

□ 또한, 서버측 이미지 맵은 링크에 대한 정보가 서버에 있기 때문에 화면 낭독 프로그램 등의 보조 기기가 그 내용을 해석할 수 없음.

- 따라서 이미지 맵을 사용할 경우, 클라이언트측 이미지 맵 쓰는 것을 권장하며,
- 부득이 서버측 이미지 맵을 쓸 경우에는 맵에서 사용되는 링크 목록을 별도의 텍스트로 제공하는 것이 바람직함.

□ 사용 예 : 여기에서 마우스로 특정한 좌표를 클릭하면(예를 들어  $x=10$ ,  $y=-20$ ), href에 지정된 서버 프로그램으로 좌표가 보내져 새로운 URI를 생성함.

- 아래의 경우 새로운 URI는  
["http://www.acme.com/cgi-bin/competition?10,20"](http://www.acme.com/cgi-bin/competition?10,20) 같은 형식이 됨.

```
<P>

```

## X (Bad)

<그림 53> 서버측 이미지 맵 사용 예



※ 출처: 000 교수 홈페이지 연구 분야 페이지

```

 <IMG src="../../../images/02_research/header.gif"
 border="0" alt="header.gif (6482 bytes)"
 usemap="#FrontPageMap1" ismap>

...중략...
<MAP name="FrontPageMap1">
 <AREA shape="rect" coords="453, 26, 557, 41"
 href="mailto:oooo@aw9500.knou.ac.kr">
 <AREA shape="rect" coords="309, 26, 444, 42"
 href="index.htm">
</MAP>
```

□ 위의 소스를 보면 A 태그 href 속성에 있는 주소로 사용자가 마우스로 클릭한 좌표 값을,

- "\_vti\_bin/shtml.dll/research3.htm/map1?x좌표,y좌표" 형식으로 서버 측에 전달하는 서버측 이미지 맵을 사용하고

있음.

□ 위의 사례에서는 단순히 두 개의 메뉴를 처리하기 위해 서버측 이미지맵을 사용하고 있는데,

○ 이미지 자체가 서버측 이미지맵을 쓸 정도로 기하학적인 형태가 아니므로 클라이언트측 이미지 맵을 사용하는 것이 바람직함.

○ 또한 서버측 이미지 맵(ismap)과 더불어 클라이언트측 이미지 맵의 형식(usemap)도 같이 사용하고 있어 클라이언트측 이미지 맵으로 작동하게 됨.

○ 따라서 서버측 이미지 맵은 사용할 필요가 없음.

## O (Good)

□ 위의 사례를 클라이언트 이미지 맵으로 수정한 소스임.

```
<IMG src="../../../images/02_research/header.gif"
border="0"
alt="선택 메뉴" usemap="#FrontPageMap1">
...중략...
<MAP name="FrontPageMap1">
 <AREA shape="rect" coords="453, 26, 557, 41"
```

```

href="mailto:0000@aw9500.knou.ac.kr" alt="email">
<AREA shape="rect" coords="309, 26, 444, 42"
href="index.htm" alt="home">
</MAP>

```

## X (Bad)

<그림 54> 대체 텍스트가 없는 클라이언트측 이미지 맵



※ 출처: 공정거래위원회 조직도

([http://www.ftc.go.kr/open\\_content/administrative/organization/intro\\_organization.php](http://www.ftc.go.kr/open_content/administrative/organization/intro_organization.php))  
, 2004. 11. 20

- 위의 그림에서는 클라이언트측 이미지 맵을 사용했으나, 영역별로 대체 텍스트를 붙이지 않아, 각 영역에 시각 장애인이 접근할 수가 없음.

```
<area shape=RECT coords=370,10,394,26
 href="buseo_dept2.php?code=D33">
<area shape=RECT coords=341,8,366,25
 href="buseo_dept2.php?code=D32">
<area shape=RECT coords=286,10,338,24
 href="buseo_dept2.php?code=D30">
```

## O (Good)

- 위의 클라이언트측 이미지 사례에 대체 텍스트를 첨부한 사례임.

```
<area shape=RECT
 coords="370,10,394,26"
 href="buseo_dept2.php?code=D33"
 alt="서무">
<area shape="RECT"
 coords="341,8,366,25"
 href="buseo_dept2.php?code=D32"
 alt="경리">
<area shape="RECT"
 coords="286,10,338,24"
 href="buseo_dept2.php?code=D30"
```



alt="총무과">

#### ※ 국내 현실

- 서버측 이미지맵은 서버 측에 불필요한 부하를 발생시키며 구현의 복잡성 때문에 근래에 와서는 거의 쓰이지 않고 있으나,
- 최근에 단순 이미지를 이용한 클라이언트측 이미지 맵이 아닌, 플래시나 애플릿을 이용한 이미지 맵의 등장으로 인해 이미지 맵과 관련한 새로운 접근성 문제가 발생하고 있음.

## VI. 프레임

지침 5. 콘텐츠를 구성하는 프레임의 수는 최소한으로 하며, 프레임을 사용할 경우에는 프레임별로 제목(title)을 붙여야 한다.

① 웹 콘텐츠에는 가급적 프레임을 사용하지 않아야 한다. 만일 프레임을 사용하는 경우에도 사용하는 프레임의 수를 최소한으로 줄여야 한다.

② 프레임을 사용할 경우에는 프레임 별로 서로 독특한(중복되지 않는) 제목을 부여하여 프레임을 식별할 수 있어야 한다.

□ 프레임에 제목을 넣어주는 이유는, 문서를 화면 낭독 프로그램(screen reader)를 통해 순차적으로 탐색하게 되는 시각장애인의 경우,

○ 프레임과 프레임을 이동할 때에 해당 프레임에 어떤 내용이 들어있는지 제목을 통해 쉽게 파악할 수 있게 되지만,

□ 프레임 제목이 없을 경우에는, 일반인들은 한눈에 쉽게 알 수 있는 프레임의 내용일지라도,

○ 시각장애인들은 프레임의 내용을 짐작할 수 없으며, 프레임의 내용을 모두 다 읽어보아야 하는 불편함이 생김

- 프레임뿐만 아니라 모든 웹 문서에는 예외없이 그 문서의 내용을 가장 정확히 표현할 수 있는 제목을 붙이는 것이 매우 중요함.

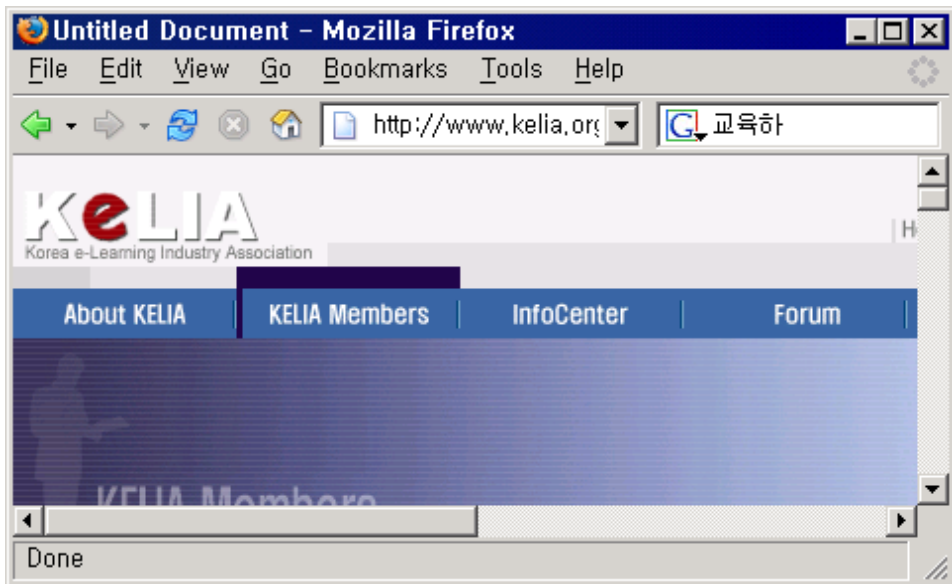
## 1. 문서의 제목 붙이기

- 모든 문서(팝업창, 프레임 문서, 내부 프레임(iframe) 문서 포함)에는 예외 없이, 그 문서의 속성을 가장 잘 대표하는 제목을 붙여야 함.
  - 제목은 <head></head> 태그 안쪽에 <title></title> 태그를 이용해 붙일 수 있으며,
  - 제목이 없거나 제목이 잘못 붙은 문서는 일반인이 웹을 탐색할 때에도 혼란을 주며,
    - 시각 장애인과 같이 비시각적으로 웹을 탐색하는 사람에게는 극심한 혼란을 초래할 수 있음
  - 제목이 없거나 잘못 매겨진 문서는 검색 엔진이 정확한 문서를 찾는 데에도 방해가 되며,
    - 검색을 성공적으로 마친 후에도 검색자가 자신이 원하던

정보 및 문서인지를 쉽게 확인할 수 없음.

## X (Bad)

<그림 55> 제목이 없는 문서(웹 페이지)



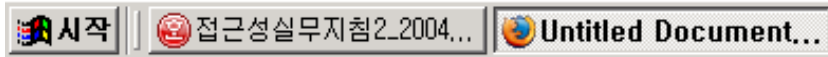
※ 출처: 한국이러닝산업협회 ([http://www.kelia.org/Kelia\\_Members/Kelia\\_Members\\_a.asp](http://www.kelia.org/Kelia_Members/Kelia_Members_a.asp)), 2004. 10. 31

□ 위의 그림처럼 문서에 제목이 없는 경우,

- 현재 열린 창의 제목줄에 'Untitled Document'라고만 표시되어 일반인들도 문서의 특성을 알기가 힘들며
- 시각 장애인들의 경우, 여러 개의 문서들 사이를 구분하여 이동하는 것이 불가능해짐.

## X (Bad)

<그림 56> 작업 표시줄에 나타난 제목이 없는 문서



- 위의 그림에서와 같이 제목이 없는 문서를 사용할 경우에는
  - 여러 개의 프로그램을 열어놓고 작업 표시줄에서 사용자가 원하는 프로그램 하나를 선택할 때에도 제목이 나타나지 않아 선택에 혼란을 줌.

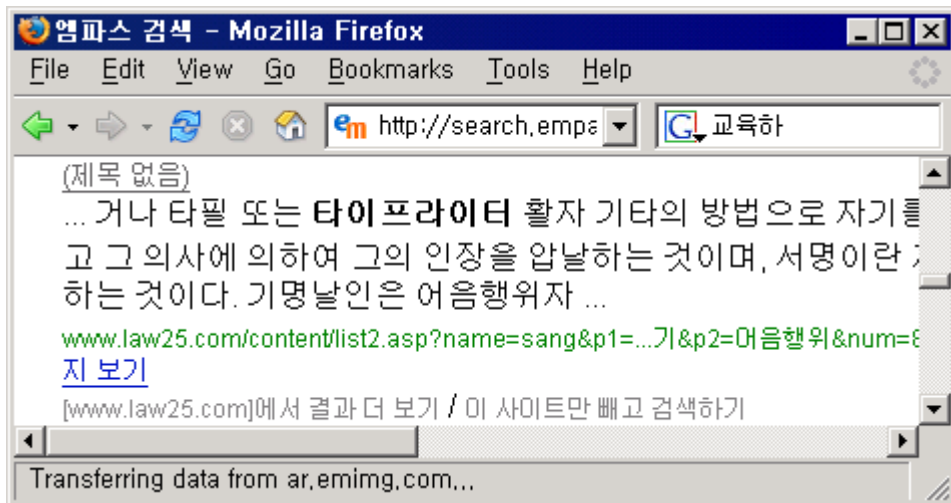
```
<html>
<head>
:
</head>
```

## O (Good)

```
<html>
<head>
 <title>한국이러닝산업협회 - 회원 소개</title>
</head>
```

## X (Bad)

<그림 57> 제목 없는 문서가 검색된 사례



※ 출처: 엠파스 검색 결과

(<http://search.empas.com/search/all.html?q=%C5%B8%C0%CC%C7%C1%B6%F3%C0%CC%C5%CD&qn=&m=B&n=15&ac=&rv=0&sd=0&v=33&w=cea1a2a3a4a5a6a7a8a9aaabacadaeafb0&cs=A&ss=A&l=0-5.0.0-0-1745-t&s=t&z=A&rq=&e=106>), 2004.

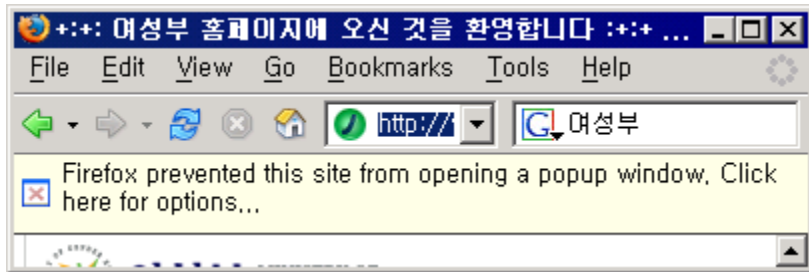
10. 31

□ 위의 그림에서와 같이 제목이 없는 문서는

- 검색 엔진의 검색 결과, 검색 순위에서 밀릴 뿐 아니라,
- 검색 결과를 보고 사용자가 해당 문서를 선택하는데 충분한 정보를 제공해주지 못함.

## X (Bad)

<그림 58> 제목에 이상한 문자를 사용한 사례



※ 출처: 여성부 (<http://www.moge.go.kr/>), 2004. 10. 31

- 위의 그림에서는 웹 페이지의 제목(여성부 홈페이지에 오신 것을 환영합니다)을 제공하고 있으나
  - 시각적 효과를 위해 제목에 이상한 문자(+, : 등)를 사용하여, 음성으로 읽었을 때 이해할 수 없는 내용이 함께 나타나는 문제가 발생함.
  - 따라서, 제목에는 시각적인 효과를 나타내는 문자(\*, +, :, # 등)는 사용하지 않는 것이 좋음.

## O (Good)

<그림 59> 세부 페이지마다 정확한 제목을 제공한 사례



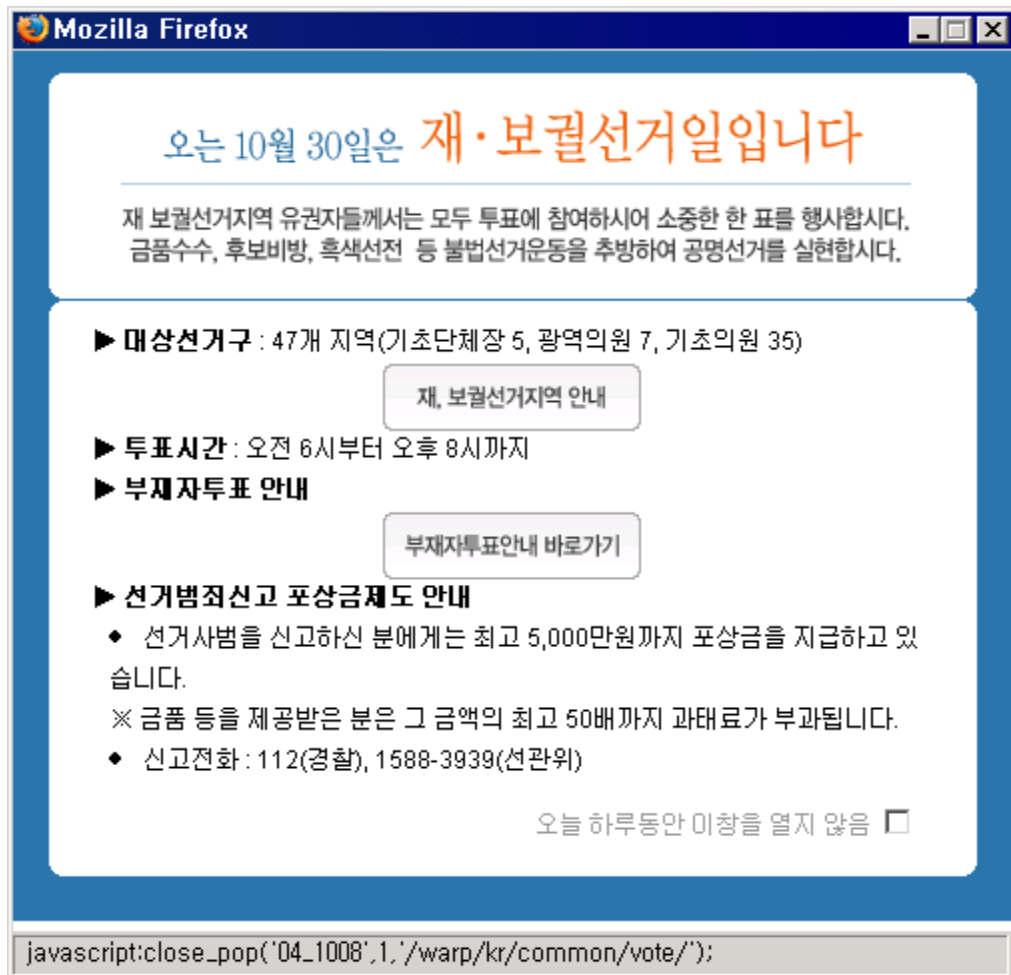
※ 출처: 미국 야후 (<http://www.yahoo.com>), 2004. 10. 31

- 위의 그림에서는 가장 첫 페이지(Yahoo!)와 첫 번째 세부 페이지(Yahoo! News), 그리고 두 번째 세부 페이지(Yahoo! News - World)의 제목을 명확히 제공해 주어
  - 제목만으로도 현재 사용자가 어떤 문서를 보고 있는지 쉽게 구분이 가능함.



## X (Bad)

<그림 60> 제목이 없는 팝업창



※ 출처: 행정자치부 ([http://www.mogaha.go.kr/kr/popup/04\\_1016/popup.html](http://www.mogaha.go.kr/kr/popup/04_1016/popup.html)),  
2004. 10. 31

□ 위의 그림에서는 원래의 창과는 별개로 공지 사항을 전달하기 위해 팝업 창을 사용하였으나

- 문서에 제목을 붙이지 않아서 시각 장애인들은 그 문서가 어떤 문서인지 끝까지 읽어보지 않고는 알 수가 없음.
- 팝업 창에 사용되는 문서나 프레임에 사용되는 프레임 문서 등 모든 문서에는 반드시 <title></title>을 사용하여 그 문서의 내용을 대표할 수 있는 정확한 제목을 붙여야 함.

## 2. 프레임 제목 붙이기

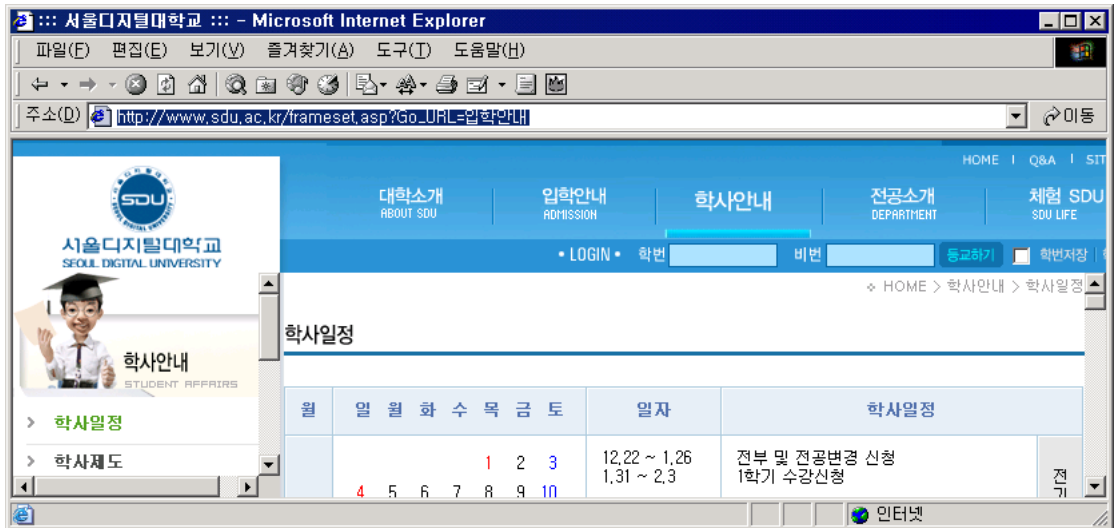
- 프레임을 사용할 때에는 여러 개의 문서를 모아 지정해놓은 프레임셋 문서와 각각의 개별 문서들로 이루어짐.
- 프레임셋 문서는 <frameset> 요소와 개별 문서를 지정하는 <frame> 요소로 이루어지며,
  - 각각의 <frame> 요소는 나중에 브라우저가 참조하기 위한 짧은 이름(name)과 더불어 프레임간의 이동을 쉽게 하기 위해,
  - 홈페이지 사용자들이 이해할 수 있는 유의미한 제목(title)을 반드시 제공해야 함.

### O (Good)

```
<frameset rows="20%,80%">
<frame name="top_frame" src="top_menu.html"
title="메뉴 프레임">
<frame name="bottom_frame" src="body_text.html"
title="본문 프레임">
</frameset>
```

## X (Bad)

<그림 61> 프레임 제목(title)을 붙이지 않은 경우



※ 출처: 서울디지털대학교 ([http://www.sdu.ac.kr/frameset.asp?Go\\_URL=입학안내](http://www.sdu.ac.kr/frameset.asp?Go_URL=입학안내)),  
2004. 8.21

```
<FRAMESET ROWS="99,*" BORDER="0" FRAMESPACING="0"
FRAMEBORDER="NO">
<FRAME SRC="/guest_Top.asp" NAME="frame_top"
SCROLLING="no" NORESIZE MARGINWIDTH="0"
MARGINHEIGHT="0" FRAMEBORDER="NO">
<FRAMESET COLS="200,*" BORDER="0" FRAMESPACING="0"
FRAMEBORDER="NO">
<FRAME
SRC="/include/Menu_Script/g02_Entrance_Menu.asp"
NAME="frame_left" MARGINWIDTH="0" MARGINHEIGHT="0"
FRAMEBORDER="NO" SCROLLING="AUTO">
```

```
<FRAME SRC="/02_Admission/main.asp"
NAME="frame_right" MARGINWIDTH="0" MARGINHEIGHT="0"
FRAMEBORDER="NO" NORESIZE>
</FRAMESET>
</FRAMESET>
```

- 위의 사례는 프레임에 제목(title)을 붙이지 않은 사례임.
- 프레임셋 문서에서 프레임들을 나눌 때에는 프레임 소스 하나마다 name="" 과 title=""을 지정해주어야 함.
  - name은 나중에 문서를 불러올 위치(target)를 지정하는 것으로 frame\_left 등과 같이 (띄어쓰기 없는) 영어 및 숫자 문자열로 지정함. (CDATA 형식: <http://www.w3.org/TR/html401/types.html#type-cdata>)
  - title은 사용자가 프레임들을 구분하기 쉽도록 안내해주는 것으로 “좌측 메뉴 프레임”과 같이 이해하기 쉬운 말로 작성함.
- 아래는 위의 문서를 title을 붙여 수정한 사례임.

## O (Good)

```
<FRAMESET ROWS="99, *" FRAMEBORDER="0">
<FRAME SRC="/guest_Top.asp" NAME="frame_top"
SCROLLING="no" NORESIZE MARGINWIDTH="0"
MARGINHEIGHT="0" FRAMEBORDER="NO" title="로고
```

들어가는 프레임">

```
<FRAMESET COLS="200,*" BORDER="0" FRAMESPACING="0"
FRAMEBORDER="NO">
```

```
<FRAME
```

```
SRC="/include/Menu_Script/g02_Entrance_Menu.asp"
```

```
NAME="frame_left" MARGINWIDTH="0" MARGINHEIGHT="0"
```

```
FRAMEBORDER="0" SCROLLING="AUTO" title="좌측 메뉴
프레임">
```

```
<FRAME SRC="/02_Admission/main.asp"
```

```
NAME="frame_right" MARGINWIDTH="0" MARGINHEIGHT="0"
```

```
FRAMEBORDER="0" NORESIZE title="본문 내용 프레임">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

### 3. 흔히 발견되는 프레임 오류

- 프레임셋 문서의 경우, <frameset> 요소가 <body> 요소를 대체하므로 <frameset>과 <body>를 모두 사용하는 것은 문법에 어긋나는 것이며,
  - 프레임을 사용할 수 없을 경우에 대안적으로 보여줄 내용은 <noframes> 요소 안에 배치하되, <noframes> 자체는 <frameset> 요소 안쪽에 배치(nested)되어야 함

- <frameset> 요소의 경우, border라는 속성값은 쓸 수 없으며 대신 <frame> 요소에 대해서는
  - frameborder="1" 또는 frameborder="0"을 사용하여 프레임의 경계선 속성을 지정해야 함.

## **X (Bad)**

```
<html>
<head>
<title>대법원 홈페이지에 오신 걸 환영합니다.</title>
<meta http-equiv="Content-Type" content="text/html;
charset=euc-kr">
</head>

<frameset rows="75,1*" frameborder="NO" border="0"
framespacing="0">
 <frame name="menu" scrolling="NO" noresize
src="k_menu.html">
 <frame name="body" src="kc/kc_c100.html">
</frameset>
<noframes>
 <body bgcolor="#FFFFFF">
 </body>
</noframes>
</html>
```

※ 출처 : 대법원 홈페이지 ([http://www.scourt.go.kr/kc\\_p.html](http://www.scourt.go.kr/kc_p.html)), 2004. 10. 31

□ 위의 대법원 홈페이지에서

- 프레임셋 문서의 제목이 '대법원 홈페이지에 오신 걸 환영합니다.'라고만 되어 있어, 다른 페이지와 차별성이 나타나지 않아 홈페이지의 이용자에게 혼란을 줄 우려가 있으므로,
- 해당 페이지를 정확하게 표현할 수 있도록 수정해야 함.
  - 예를 들면 '대법원 홈페이지 - 법원 소식'과 같이 대법원 홈페이지 내에서 어떤 부분에 있다는 것을 알려주어야 함.
- 또한 이 문서의 형식이 프레임셋 표준을 따른다는 것을 <!DOCTYPE...>을 사용하여 맨 앞부분에 선언해주어야 함.
- <frameset> 요소의 속성으로 border와 framespacing 이라는 속성은 잘못된 것으로,
  - 이 속성들은 <frame> 요소로 옮겨져야 하며 각각 frameborder와 marginwidth / marginheight로 대체되어야 함.
  - frameborder 속성값은 0 또는 1만 가능함.
- <frame> 제목은 title 속성을 사용하며, 일반 홈페이지 이용자들이 이해하기 쉬운 용어로 지정해야 함.



- <noframes>는 <frameset> 안쪽에 배치되어야 하며,
  - <noframes> 요소에 <body>만 있고 아무것도 없을 경우에는 프레임을 지원하지 못하는 브라우저를 사용하는 사람들이 아무런 정보를 얻지 못하는 결과를 초래하므로,
  - 별도의 페이지를 안내해주거나, 최소한의 안내 문구, 예를 들면, “프레임을 볼 수 있는 브라우저가 필요합니다.”라도 넣어주어야 할 것임.

## O (Good)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html lang="ko">
<head>
<title>대법원 - 법원 소식</title>
<meta http-equiv="Content-Type" content="text/html;
charset=euc-kr">
</head>
<frameset rows="75,1*">
<frame name="menu" title="메뉴 부분" scrolling="NO"
noresize src="k_menu.html" frameborder="0"
marginwidth="0" marginheight="0">
<frame name="body" title="본문 부분"
```

```

src="kc/kc_c100.html" frameborder="0"
marginwidth="0" marginheight="0">
 <noframes>
 <p>프레임을 지원하지 않는 브라우저는 프레임 없는 페이지를
보십시오.</p>
 </noframes>
</frameset>
</html>

```

## **X (Bad)**

```

<frameset rows="93,*" border="0" framemargin="0"
framespacing="0" frameborder="0">
<frame name="_topmenu" src="_wnTop.jsp"
leftmargin="0" topmargin="0" scrolling="no"
noresize/>
 <frameset cols="160,*">
 <frame name="_leftmenu" src="_wnLeft.jsp"
leftmargin="0" marginwidth="0" marginheight="0"
topmargin="0" scrolling="auto" frameborder="NO">
 <frame name="_body" src="/MainHome">
 </frameset>
 </frameset>
</frameset>

```

※ 출처: 워크넷 (<http://www.work.go.kr/>), 2004. 10. 3

## O (Good)

```
<frameset rows="93,*">
<frame name="_topmenu" title="위쪽 메뉴 부분"
src="_wnTop.jsp" marginwidth="0" marginheight="0"
scrolling="no" noresize>
<frameset cols="160,*">
<frame name="_leftmenu" title="왼쪽 메뉴 부분"
src="_wnLeft.jsp" marginwidth="0" marginheight="0"
scrolling="auto" frameborder="0">
<frame name="_body" title="본문" src="/MainHome">
 </frameset>
<noframes>
<p>프레임 없는 버전의
문서를 보십시오.</p>
</noframes>
</frameset>
```

## VII. 깜박임, 움직임

지침 6. 콘텐츠는 스크린의 깜박거림을 피할 수 있도록 구성되어야 한다.

① 웹 콘텐츠에는 애니메이션 등과 같이 깜박거리는 주파수가 3 Hz에서 49 Hz 사이인 콘텐츠 요소들을 포함하지 않아야 한다.

② 만일 위의 ①에서 명시한 요구조건을 만족할 수 없는 웹 콘텐츠는 깜박거림이 있는 웹 페이지로 이동하기 전에 이 페이지에 깜박거림이 있음을 사전에 사용자에게 경고해주어야 한다. 또한, 스크린의 깜빡거림이 배제된 대체 페이지를 별도로 제공하여야 한다.

□ 움직이는 그림 (animated gif)나 애플릿 및 스크립트를 사용한 움직임, 플래시를 사용한 깜박임 등이 모두 대상이 되며,

○ <blink>, <marquee> 등과 같이 텍스트에 적용되는 깜박임이나 움직임과 관련된 태그의 사용도 문제가 됨.

□ 이들은 모두 html/xhtml 표준에 없는 비표준 요소일 뿐 아니라, 브라우저 호환성과 접근성 측면에서도 사용하지 않는 것이 바람직함.

□ 이러한 깜박임과 움직임을 사용할 경우, 깜박거림의 주파수 범위가 3hz에서 49hz이며 광과민성 발작의 원인이 되므로,

- 따라서 최소한 사이트 메인 페이지에서 깜박거리는 요소가 없도록 디자인하고,
  - 특정 페이지에 깜박거리는 요소가 있을 때에는 해당 페이지로 가는 링크에 경고를 표시하고
  - 깜박거림이 없는 대체 페이지와 이 페이지로 이동할 수 있는 링크를 제공하는 것이 필요함.
- 일반인에게도, 지나치게 동시에 여러 곳에서 깜박임이 있거나, 깜박거림의 주파수가 위의 범위에 들어갔을 때에는 혼란스러움을 느끼게 하며,
  - 깜박임이 원래 의도했던 주의 집중의 효과를 떨어뜨릴 수 있으므로, 깜박이는 요소의 사용을 신중하게 검토할 필요가 있음.

## X (Bad)

<그림 62> 깜박거림이 심한 국내의 사이트



※ 출처: 서울방송 (<http://www.sbs.co.kr/>), 2004. 12. 26

- 국내 대부분의 상업적 사이트가 광고, 뉴스, 메뉴, 하이라이트 섹션 등 거의 모든 부분(사각형으로 표시한 부분)이 깜박거리도록 설계되어 있고,
- 주요 메뉴를 선택할 때에 사용자의 미세한 마우스 움직임에도 크게 움직이도록 설계되어 있어 시각적으로 혼란을 줄 뿐만 아니라
  - 마우스 조작이 서툰 사용자들의 접근성을 떨어뜨리고 있음.

## O (Good)

<그림 63> 깜박임이 전혀 없는 영국의 BBC 방송 사이트



※ 출처: 영국 BBC 방송 (<http://www.bbc.co.uk/>), 2004. 12. 26

□ 국내 방송사 홈페이지와 대조적으로 영국 BBC 방송사 홈페이지의 경우, 깜박이는 그림이나 움직이는 메뉴, 움직이는 텍스트 등을 전혀 사용하지 않아










○ 차분한 느낌을 줄 뿐만 아니라, 더 많은 사용자들이 쉽게 이용할 수 있도록 배려하고 있음.

## X (Bad)

<그림 64> 대부분의 사진이 동시에 깜박거리 매우 혼란스러운 사례

Photo Focus

더 많은 물품보기

 <p><b>6,400원</b> 수정사 때</p>	 <p><b>9,900원</b> 엘리 정품</p>	 <p><b>7,900원</b> 캡싸다 일본수</p>	 <p><b>18,700원</b> 끼끼 / 4가</p>	 <p><b>22,700원</b> 섹시고양이 /</p>
 <p><b>3,300원</b> 2개1세트!</p>	 <p><b>5,900원</b> 마리이야기 /</p>	 <p><b>30,700원</b> 더블제이</p>	 <p><b>19,800원</b> MOONEE</p>	 <p><b>24,800원</b> 첼리 2차</p>

※ 출처: (주)옥션 - 포토 포커스 (<http://www.auction.co.kr>), 2004. 11. 22

- 국내 대형 전자상거래 사이트인 옥션의 홈페이지 물품보기 항목을 보면 다양한 속도의 깜박이는 GIF 파일들이 빈번하게 사용되고 있음.
- 이는 홈페이지 방문객의 시선을 끌기 위한 것으로 보이나 광과민성 환자나 주의집중에 어려움을 느끼는 사람들에게는 정보 접근에 문제를 야기할 가능성이 큼.



## X (Bad)

<그림 65> 국내 사이트들에서 자주 쓰이는  
흘러가는 뉴스의 경우에도 주의가 필요함



※ 출처: 서울시 홈페이지 (<http://www.seoul.go.kr>), 2004. 11. 22

- 위의 경우 마우스를 뉴스 문장에 갖다 대면 멈추게 처리하여 사용자에게 의한 최소한의 제어장치는 마련해 두었으나,
  - 마우스 사용이 능숙하지 않은 사용자의 경우 원하는 뉴스 기사를 정확히 선택하기 어렵고,
  - 키보드 제어장치가 없으므로 마우스를 사용할 수 없는 장애를 가진 사람들에게는 심각한 문제를 야기함
- 시각 장애인용 화면낭독 프로그램의 경우, 변하는 텍스트가 계속 생길 경우 해당 부분만을 계속 반복적으로 읽어줌으로,
  - 결국 홈페이지를 전혀 이용할 수 없게 될 수 있으므로,
  - 꼭 필요한 경우가 아니라면 흘러가는 뉴스보다는 지정된 개수의 최신의 뉴스만을 고정적으로 보여주는 것이 바람직함.

## VIII.장치 독립성

지침 7. 키보드(또는 키보드 인터페이스)만으로도 웹 콘텐츠가 제공하는 모든 기능을 수행할 수 있어야 한다.

① 웹 콘텐츠는 키보드 또는 장애를 극복하도록 도와주는 여러 가지 입력 장치를 보조기구로 사용하는 경우에도 콘텐츠가 제공하는 모든 기능을 사용할 수 있어야 한다.

- 동 지침의 실천적 목표는 마우스를 사용할 수 없는 시각 장애인들과 마우스를 사용하기 힘든 지체장애를 가진 사람들을 배려하기 위한 것으로,
  - 특히 마우스를 클릭하는 등의 이벤트 발생 시 이벤트 처리기가 스크립트를 불러오게 되는데
  - 이 때 마우스와 키보드 등 특정한 장치와 연관된 이벤트 처리기보다는 장치 독립적인(device-independent) 이벤트 처리기를 사용하는 것이 바람직함.
- 마우스 뿐만 아니라 키보드로도 스크립트가 접근 가능하게 하려면 대응하는 키보드 이벤트 처리기를 추가하여야 하며,
  - 마우스의 더블 클릭의 경우, 대응하는 키보드의 이벤트가

없으므로 사용시 주의가 필요함.

- 기하학적인 도형으로 정의가 안 되는 영역이 있는 경우를 제외하고는 서버측 이미지 맵(server-side image maps) 대신에,
  - 반드시 클라이언트측 이미지 맵(client-side image maps)을 사용해야 함.(지침 4 참고)
- 키보드를 통해 순차적으로(예를 들면 Tab 키를 이용) 또는 직접적으로(단축키를 이용) 웹 페이지 내의 요소들을 탐색할 수 있어야 함.

## 1. 장치독립적인 이벤트 발생기 사용

- 사용자 상호작용 관점의 이벤트가 아닌 응용 프로그램 관점의 이벤트를 사용해야 함
- 예를 들어 HTML 4.01에서는 "onfocus", "onblur", "onselect"와 같은 것들이 응용 프로그램 관점에서의 이벤트 속성으로,
  - 이것들은 특정 입력 장치에 연계된 이벤트가 아니므로, 장치 독립적인(device independent) 이벤트라 할 수 있음

- 또한, "onmousedown", "onmouseup", "onclick", "ondblclick" 등은 마우스를 이용한 사용자 상호작용 관점의 이벤트 발생기로,
  - 마우스가 없거나 쓸 수 없는 사용자에게 문제가 발생함.
- 마우스와 관련된 이벤트를 꼭 써야 할 경우에는 그것에 대응하는 키보드 이벤트를 같이 넣어주는 것이 바람직함.
  - "onmousedown" 사용시는 "onkeydown"을 병행
  - "onmouseup"을 사용시는 "onkeyup"을 병행
  - "onclick"을 사용시는 "onkeypress"를 병행
    - 사실상 현존하는 대부분의 브라우저는 onclick 이벤트에 대해서 onkeypress를 동시에 지원해주므로 현실적으로는 onkeypress를 명시할 필요성은 줄어들고 있음.
- 마우스 좌표에 의존하는 이벤트를 쓰는 것은 장치 독립적인 입력 관점에서 바람직하지 않음.

## X (Bad)

<그림 66> 마우스로만 조작 가능한 메뉴

지원정책	인력지원	관련자
자금/금융지원 >	▪ 산업기능요원제도 운영	1건
인력지원 >	▪ 외국인 산업연수제도 운영	1건
창업벤처지원 >	▪ 생산현장직무기피요인 해소사업	13건
기술지원 >	▪ 해외고급기술인력 도입 사업	1건
판로/수출 >	▪ 대학생 중소기업체험활동	3건
정보화지원 >	▪ 경영기술지원단 운영	3건
소기업/여성기업 >	▪ 중소기업 인력 실태 조사	9건
제조물책임(PL) >	▪ 청년채용 패키지 사업	1건
재래시장지원/기타 >		

※ 출처: 중소기업청(<http://www.smba.go.kr>) 홈페이지

```
<a href="sub_frame2.jsp?page=/portal/
portal.policy.PolicyServlet?yl=01"
onMouseOut="MM_swapImageResore()"
onMouseOver="MM_swapImage('Image36','','images/y_le
fte_02.gif',1);MM_showHideLayers('Layer1
...중략
```

□ 위의 그림에서는 지원 정책 메뉴를 마우스로만 선택하도록 되어 있음.

○ 즉, 각각의 메뉴 링크에 onMouseOut, onMouseOver 이벤트를 사용하여 키보드를 이용한 접근은 제한됨.

- onFocus, onBlur를 추가하여 작성하면 이론적으로 최소의 키보드 접근이 가능하게 됨.
- 현실적으로는 키보드 접근성을 보장하기 위해 탭(Tab) 키로 이동시 탐색 순서, 마우스 좌표 위주의 설계 등 몇몇 가지를 더 고려해야 사실상 키보드 접근이 가능해짐.

## O (Good)

```
<a href="sub_frame2.jsp?page=/portal/portal.policy.PolicyServlet?yl=01"
onMouseOut="MM_swapImgRestore()"
onBlur="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Image36','','images/y_leftfte_02.gif',1);MM_showHideLayers('Layer1....중략
onFocus="MM_swapImage('Image36','','images/y_leftfte_02.gif',1);MM_showHideLayers('Layer1....중략
```

## 2. 키보드 탐색

- 시각 장애인, 지체 장애인(뇌병변 장애 포함) 등 많은 사용자들이 마우스와 같은 위치 지시 장치(pointing device)를 사용할 수 없으며,

- 특수 환경(운전 중 등)에서는 일반인도 마우스를 사용하기

힘든 경우가 발생함.

- 따라서, 마우스가 아닌 키보드만으로도 메뉴 선택과 링크 선택 등 웹 페이지 내의 상호작용이 가능해야 함.
- 페이지 내의 선택 가능한 요소들 사이를 이동할 때에는 Tab 키(전진), Shift + Tab 키(후진)를 이용함.
  - Tab 키를 이용해 전진 및 후진 이동할 때에 논리적으로 타당한 순서대로 이동하도록 제작되어야 함.
- 텍스트 필드의 특정한 값을 조사하여, 오류 발생시 강제로 포커스(focus)를 특정한 양식(form)으로 이동시킬 경우, 키보드를 이용한 입력이 불가능해질 수 있으므로 주의해야 함.

## (1) tabindex 속성

- 보통은 소스에서 작성된 순서에 따라 요소들 사이를 Tab키를 이용해 이동할 수 있음.
- 하지만, 소스상의 순서와 다르게 이동하는 것이 논리적으로 더 타당할 때 또는 명확히 이동 순서를 웹 콘텐츠 제작자가

지정하고자 하는 경우에는,

- Tab 키를 이용해 요소(element)들을 이동할 때에 각 요소에 `tabindex` 값을 넣어주어 요소들간의 이동 순서를 웹 콘텐츠 제작자가 정함.
- 하지만, 모든 요소에 `tabindex` 값을 넣을 수는 없고,
  - A(앵커), AREA(이미지 맵의 영역 표시), BUTTON(버튼), INPUT(입력 양식), OBJECT(객체 표시), SELECT(목록 상자 양식), TEXTAREA(텍스트 입력 양식)에 대해서만 `tabindex`를 넣을 수 있음.
- 사례 : `tabindex`가 중복될 경우, 소스에서 먼저 출현하는 것이 앞선 순서를 배정받음. 아래의 예에서는 BUTTON, INPUT filed1, INPUT filed2, INPUT submit, A 순으로 포커스가 이동함.

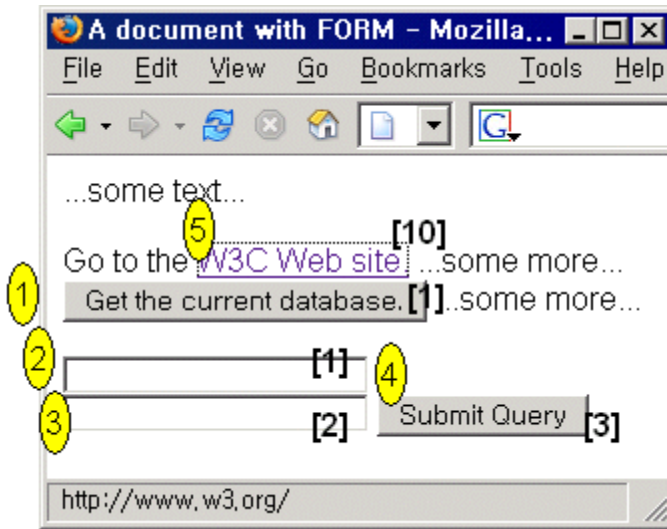
## O (Good)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>복잡한 tabindex 사용 예</TITLE>
</HEAD>
<BODY>
```



```
...some text...
<P>Go to the
W3C Web
site.
...some more...
<BUTTON type="button" name="get-database"
 tabindex="1" onclick="get-database">
 Get the current database.
</BUTTON>
...some more...
<FORM action="..." method="post">
<P>
<INPUT tabindex="1" type="text" name="field1">
<INPUT tabindex="2" type="text" name="field2">
<INPUT tabindex="3" type="submit" name="submit">
</P>
</FORM>
</BODY>
</HTML>
```

<그림 67> 복잡한 tabindex 사용 예



※ 출처: W3C의 HTML 4.01 명세

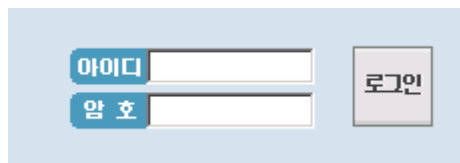
(<http://www.w3.org/TR/html401/interact/forms.html#edef-tabindex>), 2004.11.8

- 위의 그림에서 노란색 동그라미로 표시된 번호는 탭 키의 이동 순서이며, [ ] 안에 표시된 숫자는 소스에서 지정한 tabindex 값임.
- 요소의 출현 순서보다 tabindex가 더 우선적으로 적용되지만, 그 값이 동일하면 요소의 출현 순서에 따라 이동 순서가 결정됨.
- 즉, [Get the current database] 버튼과 그 밑에 있는 텍스트 입력 상자는 동일하게 tabindex="1"을 가지고 있지만 [Get the current database] 버튼이 더 먼저 출현하므로 키보드로 이동할 때에 가장 먼저 포커스를 받게 됨.

## O (Good)

```
<input type=image src="/img/b_login.gif"
style="border:0" tabindex='3' alt='로그인하기'>
<!--중간생략-->
<input type=text name=pUserId size=12 maxlength=15
onFocus="this.select()" tabindex='1'>
<!--중간생략-->
<input type=password name=pPassWd size=12
maxlength=15 onFocus="this.select()" tabindex='2'>
```

<그림 68> 입력 순서를 tabindex  
값을 통해 바르게 지정한 예



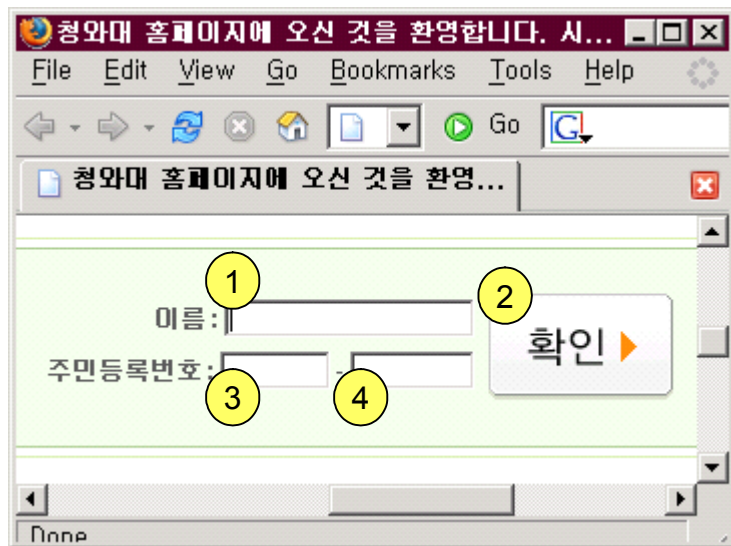
※ 출처: 정보통신 사이버대학(<http://www.ituniv.or.kr/bin/common/login2.php>), 2004. 10. 31

- 위의 사례에서, 소스 내에서는 로그인 버튼 → 아이디 입력창 → 암호 입력창 순으로 각 요소가 출현하지만,
  - **tabindex**를 이용하여 아이디 입력창 → 암호 입력창 → 로그인 버튼 순으로 탭키가 이동하도록 설정해놓아 사용자들의 편의성을 높여주고 있음.

## X (Bad)

- 아래의 그림에서는 키보드로 이동할 때에 이름 → 확인 버튼 → 주민등록번호 앞자리 → 주민등록번호 뒷자리 순으로 포커스가 이동하여
  - 이름을 입력한 다음 바로 주민등록번호를 입력할 수 없고, '확인' 버튼을 건너 뛴 다음에 주민등록번호를 입력 가능함.
  - 따라서 마우스가 아닌 키보드 사용자에게 혼란을 가져오는 잘못된 사례임.

<그림 69> 탭키 순서가 지정되지 않아 불편을 주는 사례



※ 출처: 청와대 홈페이지 회원 가입 (<http://www.cwd.go.kr/cwd/kr/mem/prechek.php>), 2004.  
10. 31

## (2) accesskey 속성

□ 중요한 요소에는 `accesskey` 속성을 이용해 직접 접근 가능한 단축키를 넣어줌으로써 탭키를 이용한 순차적인 접근에 대해서 보완할 수 있음.

○ 단축키를 넣을 수 있는 요소로는 `A`, `AREA`, `BUTTON`, `INPUT`, `LABEL`, `LEGEND`, `TEXTAREA`가 있음.

□ 보통 Windows 기반의 PC에서는 `Alt+단축키`를, Mac OS에서는 `Cmd+단축키`를 누르면 단축키가 실행됨.

□ 아래의 예에서는 단축키 `U`를 누르면 '이름'이라는 `LABEL` 요소가 활성화되어, 바로 텍스트 입력 필드를 활성화시켜 바로 이름을 입력할 수 있게 됨.

```
<FORM action="..." method="post">
<P>
<LABEL for="fuser" accesskey="U">이름</LABEL>
<INPUT type="text" name="user" id="fuser">
</P>
</FORM>
```

※ 출처:HTML 4.01 Specification

(<http://www.w3.org/TR/html401/interact/forms.html#edef-accesskey>), 2004. 11. 10

□ 아래의 예와 같이 앵커(A)에 단축키를 넣어주면,

- 링크에 포커스가 이동하거나(인터넷 익스플로러의 경우) 또는 링크가 가리키는 문서로 직접 이동할 수 있음(모질라 계열 브라우저의 경우).

```
<A
accesskey="C" rel="contents"
href="http://someplace.com/specification/contents.h
tml">목차
```

□ accesskey 사용시 주의

- accesskey는 브라우저에서 기본적으로 제공하는 단축키와 충돌을 일으킬 수 있으므로, 사용시 주의가 요구됨.
  - 특히 사용자의 시스템과 사용자가 사용하는 브라우저, OS, 프로그램 종류에 따라 어떤 단축키가 사용될 지 알 수 없으므로,
- 꼭 필요한 경우가 아니면, tabindex를 활용하고, accesskey 사용을 자제하는 것이 안전한 방법임.

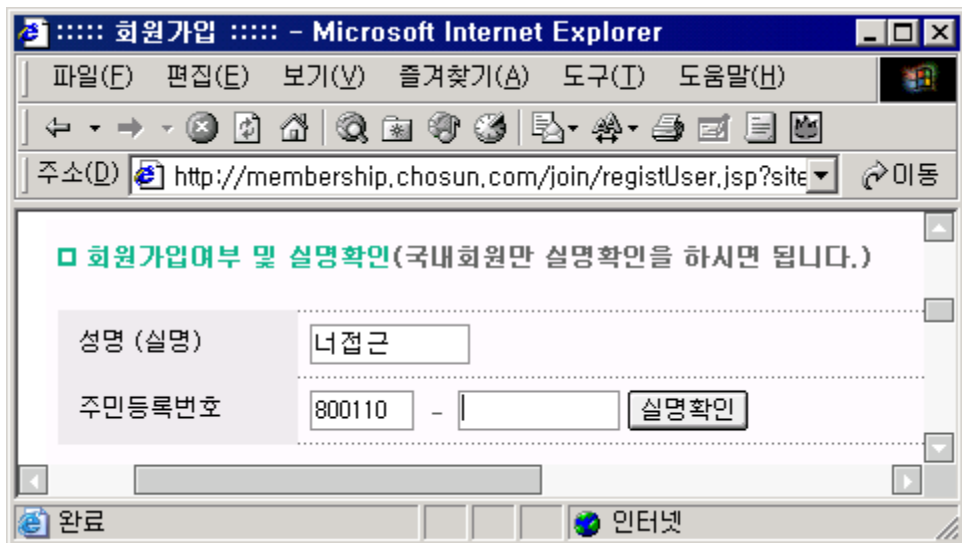
### (3) 자동 초점 이동

## X (Bad)

□ 아래의 그림에서는 <그림 82>과는 달리, 키보드를 이용할 때에도 성명 → 주민등록번호 → 실명확인 버튼 순으로 이동이 가능하지만,

- 주민등록번호 앞자리를 입력했을 때에 자동으로 뒷자리로 넘어가도록 해놓아, 실수로 잘못 입력했을 경우, 키보드로는 다시 앞자리로 이동(Shift+Tab)할 수 없는 문제가 발생함.

<그림 70> 회원 가입시 흔히 사용되는 주민등록번호 입력의 문제점



※ 출처: 조선닷컴 회원 가입(<http://membership.chosun.com/join/registUser.jsp>), 2004. 10.

31

□ 아래는 <그림 83>에 대한 HTML 소스로, onKeydown 이벤트가 발생했을 때에 자동으로 Enter\_Key라는 함수를 호출하여 주민등록번호 뒷자리(reg2)로 이동하도록 되어있음.

## ***X (Bad)***

```
// 자동으로 포커스를 이동하는 자바스크립트 함수

function Enter_key(toform) {

 if(document.m_body.hansin.value == 'true')
 document.m_body.hansin.value = 'false';

 if(event.keyCode == 13){
 toform.focus();
 }
}

<!-- 중간 생략-->

<!--주민등록번호 앞자리-->

```





```
onKeyDown='Enter_key(this.form.real_auth);'>
```

#### (4) 키보드를 통한 메뉴 선택

### O (Good)

<그림 71> 키보드로 이동과 선택이 가능한 2단계 메뉴의 예



※ 출처: 미국 Apple사 (<http://www.apple.com>), 2004. 10. 19

□ 위의 사례에서는 Tab키로 이동시 상위 메뉴를 왼쪽에서부터 순서대로 이동한 다음, 하위 메뉴로 이동함.

○ 따라서 마우스를 사용하지 않더라도 일반적으로 생각하는 논리적인 순서에 따라 키보드를 이용해 메뉴를 이동하고 탐색할 수 있음.

### △ (Good)

□ 아래 사례는 <그림 84>처럼 키보드로도 메뉴 이동이 가능하도록 잘 제작되었으나,

○ 플래시로 메뉴를 제작하여 플래시가 작동되지 않을 경우

(예를 들면 텍스트 브라우저 사용자나 플래시 플러그인을 설치하지 못한 사용자) 메뉴에 접근할 수 없는 문제가 있음.

□ 또한, 키보드로 메뉴 이동시 현재 선택된 메뉴를 시각적으로 확인(사각형 테두리)할 수 있도록 콘텐츠를 제작함.

○ 하지만, 현재까지는 인터넷 익스플로러 외의 다른 브라우저에서는 작동하지 않는 문제점이 있음.

<그림 72> 경우에 따라 키보드로 이동과 선택이 가능한 2단계 메뉴



※ 출처: 미국 Macromedia사 (<http://www.macromedia.com>), 2004. 10. 20

□ 위의 그림을 보면, 플래시로 제작하였지만 Tab 키를 통해 선택된 상위 메뉴의 하위 메뉴까지 논리적인 순서로 이동 가능하며,

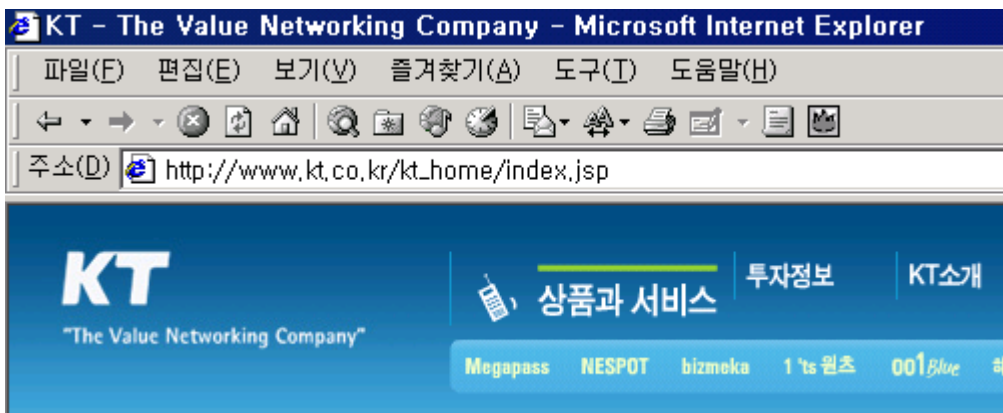
○ 선택된 부분에는 시각적인 테두리가 주어져 현재 위치 (Products → Products Home)를 알기 쉬움. 단, 특정한 브라우저에서만 작동함.

## X (Bad)

□ 아래의 사례는 2004년 현재 우리 나라 홈페이지에서 주 메뉴 구성에 자주 사용하는 기법으로,

- 주 메뉴를 접근성을 고려하지 않은 플래시로 제작하여, 이로 인해 다음과 같은 문제가 발생함.
  - 키보드로는 메뉴 선택과 이동이 전혀 불가능함.
  - 마우스를 이용할 경우에도, 매우 세밀하게 마우스 포인터를 움직이지 않으면 메뉴 선택이 어려움. (노인, 지체 장애인, 손 떨림이 있는 사용자 등)

<그림 73> 키보드 접근성을 고려하지 않은 플래시 메뉴



※ 출처: KT ([http://www.kt.co.kr/kt\\_home/index.jsp](http://www.kt.co.kr/kt_home/index.jsp)), 2004. 10. 19

- 위의 그림을 보면, Apple사와 같이 2단계의 메뉴 체계로 되어 있으나,

- 모든 메뉴를 플래시로 제작하였으며, 마우스가 아니면 전혀 메뉴 접근을 할 수 없음.
- 마우스의 움직임에 따라 메뉴의 위치, 크기, 모양이 계속 바뀌므로 미세한 마우스 동작 실수에 의해 메뉴 선택을 잘못할 가능성도 매우 큼.

## IX. 링크를 통한 이동

지침 8. 웹 콘텐츠는 반복적인 네비게이션 링크(repetitive navigation link)를 뛰어넘어 페이지의 핵심부분으로 직접 이동할 수 있도록 구성하여야 한다.

① 웹 콘텐츠 상에 반복적 네비게이션 링크 객체가 포함되어 있으며 이 링크 객체가 콘텐츠의 핵심부분보다 먼저 읽어주도록 구성된 경우에 이들 링크들의 읽기를 생략하고 직접 콘텐츠의 메인 부분으로 직접 이동할 수 있는 링크를 제공하여야 한다.

② 하나의 긴 문장으로 구성된 웹 페이지는 이 페이지의 문장을 여러 개의 논리적인 절로 구분하고 각 절의 색인을 콘텐츠의 첫 부분에 포함시켜 원하는 절의 시작 부분으로 직접 이동할 수 있도록 웹 페이지를 구성한다.

③ 모든 웹 사이트는 텍스트 또는 대체 텍스트가 포함된 텍스트 아닌 콘텐츠로 구성된 사이트 맵을 제공하여야 한다.

□ 사이트 구조상 동일한 상단 메뉴나 좌측 메뉴가 페이지마다 되풀이 될 경우에는,

○ 이를 화면낭독 프로그램이 반복적으로 읽지 않고서 바로 중요한 내용으로 건너뛸 수 있도록 링크를 넣어주어야 함

□ 즉, 각 페이지의 메뉴 시작 부분에 해당 페이지의 핵심 콘텐츠로 바로 이동할 수 있는 링크를 제공하면 메뉴를 순서

대로 읽지 않고도 핵심 콘텐츠에 직접 접근할 수 있게 됨.

□ 전체 사이트의 구조를 한 눈에 볼 수 있는 사이트 맵을 제공하는 것처럼,

○ 한 페이지 내에서도 전체 내용의 목차를 먼저 보여준 다음 해당 내용들을 보여주도록 구성하는 것이 매우 중요함.

## 1. 건너뛰기 링크

### O (Good)

<그림 74> 보이지 않는 건너뛰기 링크가 제공되는 사례

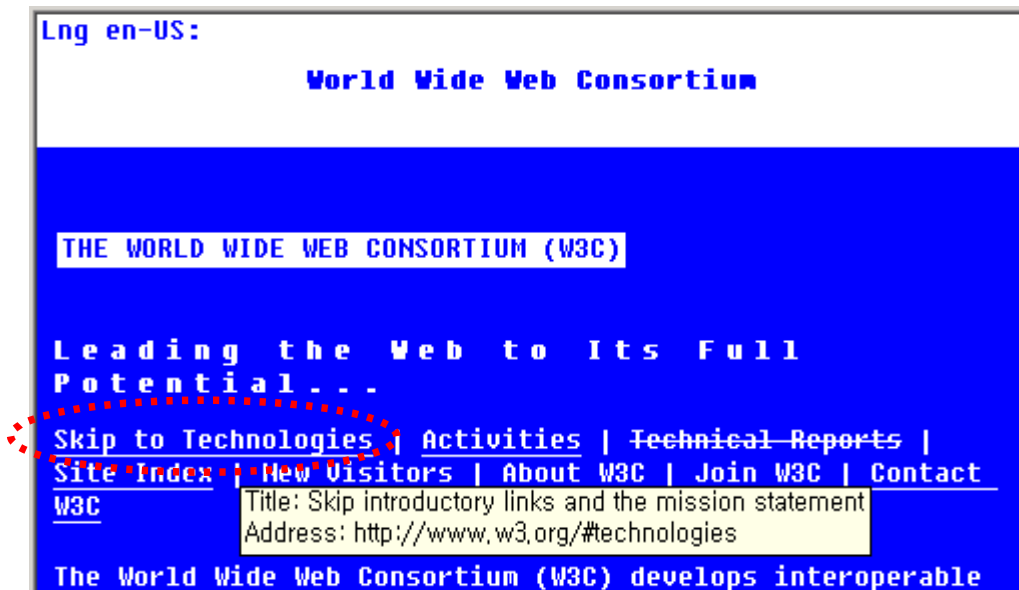


※ 출처: World Wide Web Consortium (<http://www.w3.org/>), 2004. 12. 26

□ 위의 <그림 87>에는 7개의 선택 항목(Activities, Technical Reports,..., Contact W3C)을 거치지 않고 핵심 내용으로 이동할 수 있도록 콘텐츠를 제작함.

- 즉, 'Activities' 링크 앞에 'Skip to Technologies'라는 링크를 제공하고 있음.
- 이 링크는 보통의 시각적인 브라우저에서는 눈에 보이지 않아 아무런 영향을 주지 않으나,
- <그림 88>과 같이 텍스트 전용 브라우저나, 시각 장애인용 화면 낭독 프로그램을 사용할 경우에는 모든 항목을 읽지 않고 핵심 콘텐츠로 바로 이동할 수 있는 장점이 있음.

<그림 75> 위의 사이트를 텍스트 전용 브라우저로 본 경우



※ 출처: World Wide Web Consortium (<http://www.w3.org/>), 2004. 12. 26

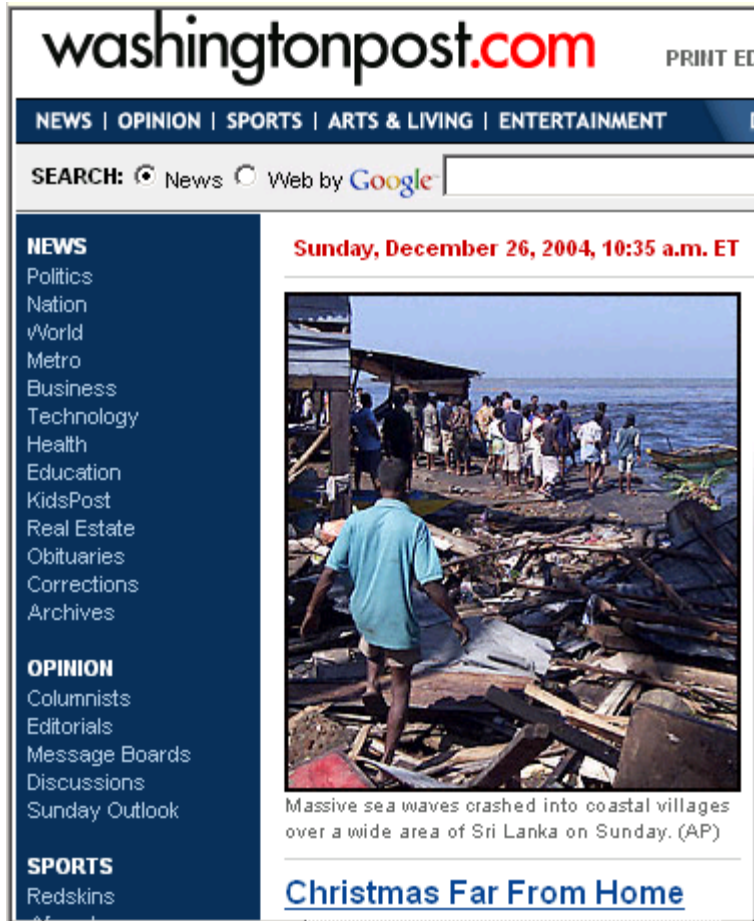


```
<a class="bannerLink"
title="Skip introductory links and the mission
statement" href="#technologies" shape="rect">Skip
to Technologies
...중략...
<a name="technologies" id="technologies"
shape="rect">W3C A to Z
```

- 위의 소스에서 'Skip to Technologies' 링크를 선택할 경우에는 같은 페이지 내의 다른 부분을 건너뛰고 '#technologies'라는 부분으로 이동하도록 콘텐츠를 제작함.

## X (Bad)

<그림 76> 너무 많은 메뉴를 건너뛰는 방법이 없는 경우



※ 출처: 와싱턴 포스트 (<http://www.washingtonpost.com>), 2004. 12. 26

□ 위의 그림에서와 같이 신문사 사이트는 전형적으로 왼쪽에 매우 많은 메뉴 항목들을 가지고 있으므로, 오늘의 주요 기사에 접근하려면 모든 메뉴를 거쳐야 함.

○ 따라서 반드시 많은 메뉴를 건너뛰고 바로 오늘의 주요

기사를 읽을 수 있는 링크를 제공해야 함.

## 2. 사이트 맵

### X (Bad)

<그림 77> 사이트 맵이 없는 홈페이지



※ 출처: 농림수산 정보센터 패밀리 사이트 아피스포털 (<http://www.affis.net>),  
2004.11.20

□ 위의 홈페이지는 상당히 많은 부분과 복잡한 메뉴로 이루어졌으나,

- 첫 페이지와 하위 페이지 모두에 사이트 맵이 제공되지 않아, 전체적으로 사이트의 구조를 파악하기가 어려움.
- 장애인들의 경우, 사이트 맵이 복잡한 웹을 탐색하는 데에 매우 큰 도움을 주므로 사이트 맵은 매우 쉽게 접근성을 높일 수 있는 방법임.

## O (Good)

<그림 78> 위계적으로 잘 구성해놓은 사이트 맵.



※ 출처: Accessify.com (<http://www.accessify.com/site-map.asp>), 2004. 12. 18

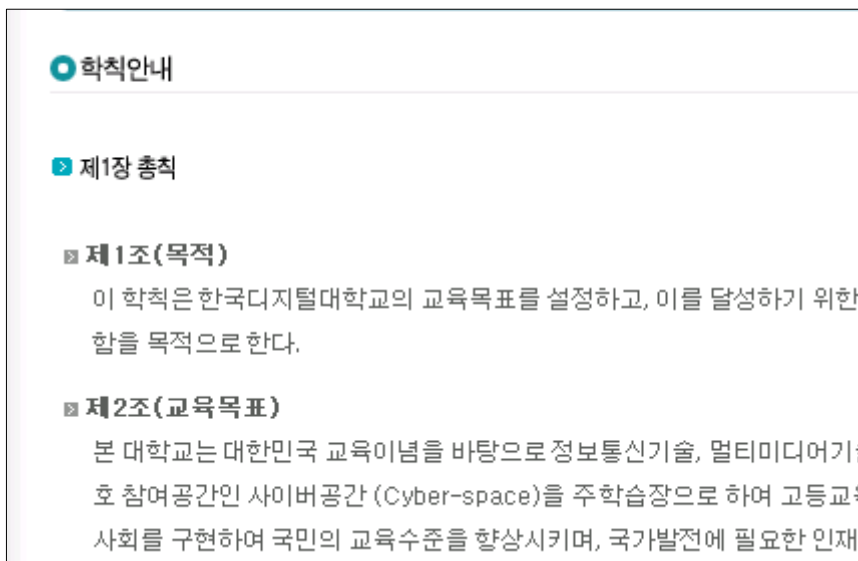
- 위의 사례에서는 사이트맵을 위계적으로 큰 메뉴, 작은 메뉴를 잘 구분하여 제공하고 있을 뿐만 아니라,
  - 위계적으로 찾아가기 힘들 경우, 바로 알파벳 순으로 원하는 페이지를 찾을 수 있도록 알파벳 순의 사이트맵을 함께 제공하고 있어,
  - 전체적인 구조를 시각적으로 한꺼번에 파악하기 힘든 장애인들에 대해서도 쉽게 원하는 페이지를 찾을 수 있도록 배려하고 있음.

### 3. 페이지 내 목차

- 한 페이지 내에서도 시각장애인의 경우에는 원하는 곳을 찾으려면 모든 문서를 순차적으로 읽어내려 가야 하므로
- 한 페이지 내에서도 전체 내용의 목차를 먼저 보여주고,
- 또한, 목차를 클릭하면 해당 내용으로 직접 이동하도록 구성해야 함.

#### **X (Bad)**

<그림 79> 목차 바로 가기가 필요한 경우



※ 출처: 한국디지털대학교 학칙 안내

([http://www.koreadu.ac.kr/kdu\\_front/center/grad\\_provision\\_1.html](http://www.koreadu.ac.kr/kdu_front/center/grad_provision_1.html)), 2004.12.26

□ 위의 그림에서 학칙은 여러 개의 장과 여러 개의 조로 이루어져 있음.

○ 이와 같이 여러 개의 조항으로 이루어진 글의 경우 맨 위쪽에 전체 조항을 열람할 수 있는 목차와 바로 가기 링크가 있어야 함.

○ 그렇지 않으면, 장애인들의 경우, 원하는 조항을 바로 갈 수 없고, 모든 조항을 순차적으로 읽어내려가야 하는 불편함이 생김.

## O (Good)

<그림 80> 복잡한 페이지에 먼저 제시된 목차

### Diseases & Conditions

Please note that documents in PDF format require [Adobe's Acrobat Reader](#)

#### Primary Sources for Diseases and Conditions:

- MEDLINEplus [About](#) | [Alphabetical Listing](#)
- Healthfinder [About](#) | [Alphabetical Listing](#)

#### Related Topics on this Page

• <a href="#">Bioethics</a>	• <a href="#">Organ Donation and Transplantation</a>
• <a href="#">Clinical Trials</a>	• <a href="#">Other General Sources of Information on Diseases and Conditions</a>
• <a href="#">Complementary and Alternative Medicine</a>	• <a href="#">Procedures and Therapies</a>
• <a href="#">Coping/Support</a>	• <a href="#">Screening and Prevention</a>
• <a href="#">Drinking and Alcohol</a>	
• <a href="#">Genetics</a>	

※ 출처: 미국 Department of Health and Human Services  
(<http://www.hhs.gov/diseases/index.shtml>), 2004. 12. 18

- 위의 그림에서와 같이 페이지의 내용이 매우 길고 여러 개의 부분으로 나뉘는 경우,
  - 맨 위쪽에 각 부분으로 이동할 수 있는 목차를 먼저 보여줌으로써 전체 내용을 쉽게 파악할 수 있고, 원하는 부분으로 쉽게 이동이 가능함.

## X. 시간 제한

지침 9. 실시간 이벤트나 제한된 시간에 수행하여야 하는 활동 등은 사용자가 시간에 구매받지 않고 읽거나, 상호작용을 하거나 응답할 수 있어야 한다.

① 웹 콘텐츠의 시간 제약 조건이 최소한 아래의 항목 중 하나는 만족하여야 한다.

- 1) 사용자가 시간제한 기능을 동작하지 않도록 할 수 있거나,
- 2) 일반적으로 사람들이 선호하는 시간의 10배 이상으로 제한 시간을 늘릴 수 있거나,
- 3) 주어진 시간이 종료하기 전에 경고를 발하며, 제한 시간을 늘리기 위하여 최소한 10초 이상 시간이 주어질 수 있어야 한다.

② 그러나 아래의 항목은 예외로 한다.

- 1) 경매 등과 같이 시간 제약이 있으나 다른 대안이 없는 경우거나,
- 2) 실시간 게임이나 시간이 정해진 범위에서 작업해야 하는 테스트 등과 같이 시간 제약이 필수적인 경우

③ 팝업 창은 가급적 사용을 피해야 한다. 그러나 꼭 사용해야 하는 경우에는 팝업 창이 열리기 전에 사용자에게 경고를 할 수 있어야 한다. 또한 팝업 창이 열리더라도 포커스가 새로 열린 팝업 창으로 이동해서는 안 된다.



## 1. 페이지 자동 이동

### (1) 메타 리프레시(meta refresh)

- ☐ 다음과 같은 경우 특정한 페이지를 방문했을 때 다른 페이지로 자동으로 바뀌도록 할 필요가 생길 수 있음.
  - 더 이상 그 페이지 주소가 유효하지 않아서, 유효한 새 주소로 이동시킬 때
  - 그 페이지에서 어떤 작업을 처리하고 나서, 다른 페이지로 이동시킬 때
  - 웹서버에서 지정한 시작 파일이 아닌 다른 파일에서 시작되도록 하고 싶을 때
- ☐ 위와 같은 경우에 일반적으로 아래와 같이 meta "refresh" 기법을 사용함.

### **X (Bad)**

```
<!-- 10초 후에 URL에서 지정된 페이지로 자동으로 이동 -->
<META HTTP-EQUIV=Refresh CONTENT="10;
URL=http://www.htmlhelp.com/">
```

□ 그러나 이 방법은 다음과 같은 이유로 사용하지 않는 것이 바람직함.

- 어떤 브라우저에서는 이런 식의 페이지 이동을 지원하지 않으며, 장애인용 화면낭독 프로그램, 텍스트 기반의 브라우저에서 이런 기능을 지원할 수 없음.
- 통상적으로 사용되는 그래픽 기반의 브라우저(인터넷 익스플로러, 모질라, 파이어폭스, 사파리, 오페라 등)에서도 무단으로 성인 사이트로 이동하는 것을 막기 위해 사용자가 메타 리프레시 기능 옵션을 꺼둘 수 있음.
- 이렇게 페이지 이동을 한 후에는, [←뒤로 가기] 버튼을 통해 페이지를 되돌아갈 수가 없고, 사용자는 아무리 뒤로 가기 버튼을 눌러도 현재 페이지에서 갇혀 버림.
- 이제까지 방문했던 페이지들의 기록(history)이 엉망이 됨.
- 의미적으로 새 페이지가 구 페이지를 대체한다는 정보를 아무것도 가지고 있지 못하므로 검색 엔진의 검색 로봇이 해당 페이지가 새 페이지로 옮겨졌다는 정보를 찾아내지 못함.
- 어떤 검색 엔진에서는 리프레시를 사용한 페이지는 의도적으로 검색 결과에서 빼버림.

- 이전 페이지의 내용을 미처 다 읽지 못했는데 사용자의 의도와 무관하게 새 페이지로 내용이 바뀌어버려 특히 느리게 읽는 사람(예: 외국인 등)에게 불편을 초래할 수 있음.
- 특정 시간 내에 어떤 일을 다 끝마치지 않으면 새 페이지로 이동해버리겠다는 것은
  - 장애인, 노인과 같이 컴퓨터 작업 속도가 일반인보다 훨씬 느린 사람들에게는 인터넷을 쓸 수 없게 만드는 결정적 요인이 됨.

## **X (Bad)**

<그림 81> 자동으로 다른 페이지로 이동하도록 잘못 제작된 페이지의 예



※ 출처: 원진에이티 (<http://www.wonjinmall.com/>), 2004. 12. 27

- 7초 후에 자동으로 페이지가 다른 주소로 넘어가도록 되어 있어, 페이지가 넘어간 후에는 뒤로 가기 버튼(←)을 이용해 원래의 페이지나, 그 이전 페이지로 돌아갈 수 없음.

## **(2) HTTP redirects**

- meta 태그를 이용한 리프레시는 웹 문서와 브라우저 사이

에서 리프레시가 일어나는 반면,

- HTTP Redirect의 경우는 서버 자체에서 이미 다른 주소를 참조하도록 설정이 되어 있어, 처음부터 브라우저는 최종 주소만을 받게 됨.

□ 또한 HTTP redirects를 쓰면 서버가 브라우저에게 주소가 바뀐 이유에 대해 더 의미 있는 정보를 HTTP 상태 코드 (status code)로 전달해주어,

- 문법 검사기, 검색 엔진, 브라우저 등이 그것을 기반으로 서로 다른 행동을 취할 수 있도록 도움을 줌
  - 영구적인 이동 (HTTP 301)
  - 일시적인 이동 (HTTP 307)
  - 정의되지 않은 이동 (HTTP 302) 등

□ 서버측 redirect를 지정하는 방법

- Apache 웹 서버의 경우 httpd.conf 파일에 redirect directive를 넣어줌

## O (Good)

```
Redirect permanent /one http://example.com/two
Redirect 303 /three http://example.com/other
```

- IIS 웹서버의 경우 Internet Information Services 스냅인 (Snap-In)에서 원하는 웹사이트나 디렉토리를 선택한 후 Property Sheet를 열고,
  - Home Directory, Virtual Directory 또는 Directory 탭을 선택한 다음, A Redirection to a URL을 선택함
  - Redirect to 상자에 이동하고자 하는 주소의 URL을 넣어 주면 됨
  - 예를 들어 /Catalog 디렉토리에 있는 파일에 대한 요청이 들어오면 그것을 /NewCatalog 디렉토리로 보내고자 할 때에, /NewCatalog라고 입력함.
- 웹 서버 관리자 권한이 없을 경우, 서버측 스크립트 언어인 서버측 스크립트 언어를 이용해 HTTP 헤더에 값을 보낼 수 있으며, PHP의 경우에는 아래와 같이 작성해야 함.

## O (Good)

```
<?php
header("Location: http://www.example.com/");
?>
```

### (3) 단순한 링크를 통한 이동

□ 시간 제약을 걸어놓고 사용자의 반응을 기다리거나, 사용자가 미처 반응하기 전에 브라우저가 다음 동작을 취하는 것은,

- 반응이 느린 사용자(장애인, 노인, 초심자 등)의 사용 편의성을 크게 떨어뜨리므로, 단순하게 정적인 링크를 사용하는 것이 가장 좋은 대안이 될 수 있음

## **O (Good)**

□ 아래의 예는 주소가 바뀌었다는 사실을 직접적으로 알려주고, 사용자가 그 바뀐 주소에 걸린 링크를 선택하여 이동할 수 있도록 한 예임.

이 페이지는 2004년 1월 1일 이후부터 `<a href="http://www.new.com/new_page">`새로운 주소`</a>`로 옮겼습니다. 책갈피/즐거찾기를 해놓으셨다면 수정해주십시오.

## 2. 빠른 내용의 전환

- 최신 뉴스를 전하기 위해 수 초마다 내용이 갱신되거나, 좌우 또는 상하로 내용이 흐르도록 할 경우,
  - 글 읽는 속도가 느리거나(정신지체 장애인, 어린이 등) 마우스 조작이 어려운 사람(노인, 지체 장애인, 컴퓨터 초보자 등)은 원하는 뉴스나 뉴스 기사를 선택할 수 없음
- 특히, 화면낭독 프로그램을 사용해 홈페이지를 보는 시각장애인의 경우 페이지의 특정한 부분이 계속 반복적으로 업데이트 되어
  - 포커스가 다른 곳으로 넘어가지 못하고 한 곳에 갇혀버릴 수 있는 문제가 발생함.

## X (Bad)

<그림 82> 수초마다 자동 갱신되는 뉴스



※ 출처: 네이버 (<http://www.naver.com>) , 2004. 12. 28

- 위의 그림에서 뉴스의 아랫 부분은 자동으로 수 초마다 갱신되도록 해놓아,
- 비시각적인 탐색을 하는 장애인들은 해당 부분으로 포커스가 반복적으로 이동해버려, 다른 부분으로 이동이 불가능해질 수도 있음.

□ 따라서, 아주 부득이한 경우가 아니라면 다음과 같은 대안적인 방법을 사용하는 것이 바람직함.

- 최신의 뉴스 기사 몇 개만을 고정적으로 보여주되, 사용자가 직접 해당 페이지를 다시 불러오거나(page reload),



- 다른 사용자가 들어왔을 때에는 다른 내용의 뉴스 기사가 나오도록 프로그래밍하여,
  - 실제로는 고정된 페이지를 보여주되, 원하는 뉴스를 모두 보여주는 효과를 간접적으로 얻을 수 있음
- 주어진 공간이 적어서 많은 뉴스를 보여주지 못할 경우에는,
  - 단순히 전체 보기(more...) 링크를 통해 뉴스 전체를 볼 수 있는 화면으로 이동하게 할 수 있음.
- 굳이 한 화면에서 좁은 공간에 더 많은 내용을 보여주고자 할 경우에는,
  - <iframe>을 사용하여, 사용자가 직접 위아래로 스크롤해가면서 뉴스 기사를 선택할 수 있으며,
  - 이러한 경우 <iframe> 요소에는 반드시 title="최신 뉴스"와 같은 식으로 이름을 붙여서 현재의 iframe 구역이 어떤 목적으로 설정되었다는 것을 알려야 함

### 3. 새 창, 팝업 창

#### (1) 링크에 연결된 새 창

□ 사용자가 특정한 링크를 선택하여 다른 곳으로 이동할 때  
에, 다음과 같은 이유로 꼭 필요한 경우가 아니면 새 창보  
다는 현재의 창에 새로운 주소가 나타나는 것이 바람직함

- 새 창에 새로운 주소가 열릴 경우, 브라우저의 방문 기록  
(history)이 남지 않아서 뒤로 가기(←)와 앞으로 가기(→)  
등 탐색에 자주 사용하는 버튼을 사용할 수 없음

- 시각 장애인의 경우, 소리를 통해 웹을 탐색해야 하므로  
한 번에 한 곳에만 주의를 집중할 수 있으며,

- 새 창이 열리는 것은 동시에 두 개 이상의 내용을 머릿  
속에 기억하고 그것들 사이를 왔다갔다 해야 함으로,

- 인지적인 부하가 급격하게 증가하여 웹에서 길을 잃고  
사용자를 혼란스럽게 만드는 원인이 됨.

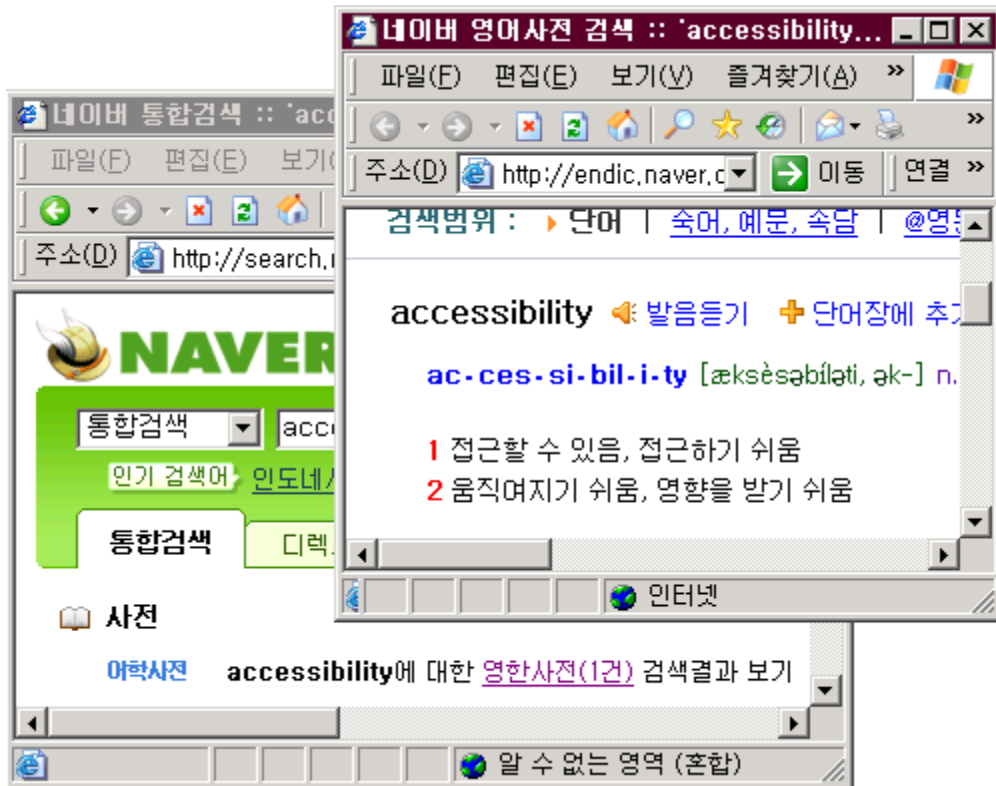
- 특히 두 개 이상의 창이 열릴 때에 창 제목(즉, 문서의 제  
목)을 정확히 부여하지 않았을 경우에는

- 일반인과 시각 장애인 등 모두에게 혼란을 줄 수 있음

□ 따라서, 부득이하게 새 창을 열어야 할 경우에는 미리 해당  
링크에 대한 설명이나 title 속성을 통해 경고를 해주는 것  
이 바람직함.

## X (Bad)

<그림 83> 경고 없이 무조건 새 창으로 뜨는 링크



```
<!-- 아무런 경고 없이 기본 링크가 새 창으로 열림 -->
<a target="_blank"
href="http://endic.naver.com/search.naver?where=end
ic&mode=srch_ke&query=accessibility"
onClick="return
goOtherCR(this, 'u='+escape(this.href)+'&r=1&i=_dumm
y_&f=nx&a=dict_en&q='+headerfooter_query_encoded);"
>영한사전(1건)
```

※ 출처: 네이버 검색 결과

(<http://search.naver.com/search.naver?where=nexearch&query=accessibility&frm=t1>), 2004.

12. 28

□ 위의 사례와 같이 국내 대다수 사이트들이 일반적인 링크에 대해 새 창이 뜨도록 해놓아 장애인들에게 불편을 주고 있음.

□ 이 경우에는 새 창이 아닌 링크로 교체하거나, 또는 아래와 같이 최소한 새 창으로 링크가 열릴 것이라는 사실을 미리 알려주어야 함.

```
<!-- 새 창으로 열린다는 사실을 미리 경고해주도록 바꾼 예 -->
<a target="_blank"
href="http://endic.naver.com/search.naver?where=endic&mode=srch_ke&query=accessibility"
onClick="return
goOtherCR(this, 'u='+escape(this.href)+'&r=1&i=_dummy_&f=nx&a=dict_en&q='+headerfooter_query_encoded);"
title="검색 결과 보기: 새 창으로 열림">영한사전(1건) (새
창)
```

## O (Good)

<그림 84> 새 창으로 열릴 것을 미리 알려준 사례.



<!-- 기본 링크는 같은 창에서 열림 -->

```
<a class="yschttl"
href="http://rds.yahoo.com/S=2766679/K=accessibilit
y/v=2/SID=w/TID=PRVT_83/l=WS1/R=1/IPC=kr/SHE=0/H=1/
SIG=11n9hrj6i/EXP=1104332492/*-http%3A//outlook.mic
rosoft.com/enable">HOME, Microsoft
Accessibility
```

<!-- 새 창으로 열고 싶을 경우 별도의 링크가 제공됨 -->

```
<a
href="http://rds.yahoo.com/S=2766679/K=accessibilit
y/v=2/SID=w/TID=PRVT_83/l=WS1/R=1/IPC=kr/SHE=0/H=1/
NW=1/SIG=11n9hrj6i/EXP=1104332492/*-http%3A//outloo
```

```
k.microsoft.com/enable" target="_blank">
```

※ 출처: 미국 야후 검색 결과

(<http://search.yahoo.com/search?p=accessibility&fr=FP-tab-web-t&toggle=1&ei=UTF-8>), 2004.  
12. 28

- 위의 그림에서와 같이 국내 사이트와 대조적으로 미국의 사이트들은 링크가 같은 창으로 열리는 것이 기본값이며,
  - 특별히 사용자가 새 창으로 열고 싶을 경우 별도의 아이콘 링크(빨강색 동그라미 부분)를 마련해 놓았음.
  - 뿐만 아니라 새 창으로 열리는 그림에는 충실하게 새 창으로 링크가 열릴 것이라는 안내를 <img>의 alt 속성을 이용해 하고 있음.

## (2) 원하지 않는 팝업 창

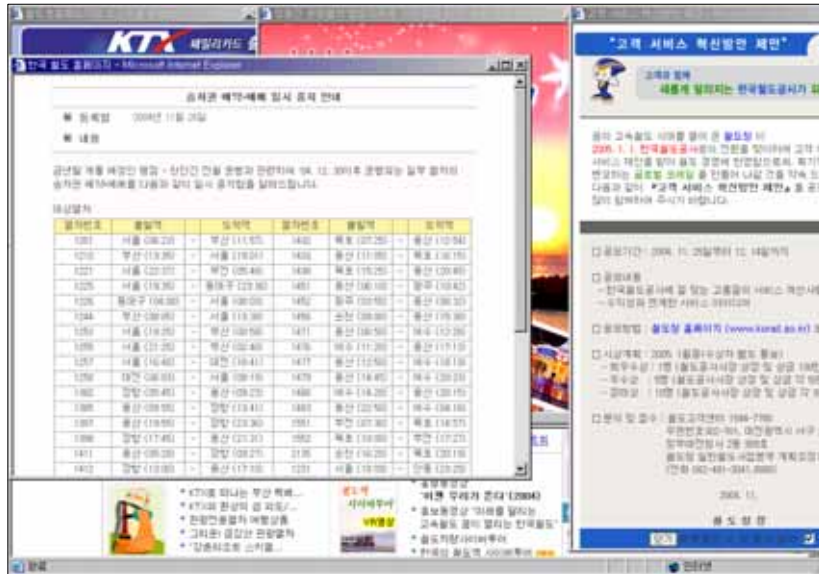
- 링크가 새 창으로 열리는 것보다 사용자들이 원하지 않았는데 팝업(Pop-up) 창이 열리는 것은 더 나쁜 영향을 줌.
  - 특히 시각 장애인들의 경우 원하지 않는 팝업 창이 여러 개 떴다는 사실을 모르고 엉뚱한 창에서 계속 웹 탐색을 진행하다 포기하는 경우가 생길 수 있음.
- 요즈음 많은 브라우저들(예: Mozilla, Mozilla Firefox,

Opera, Safari, Avant Browser, Konqueror 등과 같이 기본적으로 팝업 차단 기능이 내장된 브라우저나 Windows XP Service Pack 2나 구글 툴바와 같은 팝업 차단 소프트웨어를 설치한 Internet Explorer 등)에서 원하지 않는 팝업 창을 차단하도록 기본값이 설정되어 있어,

- 팝업 창을 이용해 일반 사용자들의 주목을 끌 수 있는 가능성도 점점 줄어들고 있음
- 많은 사용자들이 원하지 않는 팝업 창(예: 팝업 광고 등)에 대해 불편함을 느끼고,
  - 무의식적으로 팝업 창을 닫아버리는 습관으로 인해 중요한 메시지 전달을 팝업 창으로 하는 것은 실질적 효과가 없음.

**X (Bad)**

<그림 85> 여러 개의 원하지 않는 팝업 창이 뜨는 사례



※ 출처: 철도청 (<http://www.korail.go.kr/>), 2004. 11. 27

□ 위의 그림에서는 첫 페이지에 원하지 않는 팝업 창이 무려 4개씩이나 등장하여 큰 혼란을 줌.

○ 더욱이 팝업 창마다 (문서의) 제목을 정확히 붙여놓지 않아 문제가 심각해짐.

□ 위의 그림에서와 같이 모든 사용자에게 꼭 알려야 할 만큼 중요하고 긴급한 내용이라면 팝업을 무분별하게 띄우지 말고,



- 본문 페이지의 사용성과 편의성을 고려한 디자인을 통해 어떻게 사용자의 주목을 끌 수 있을지 고민해야 할 것임.

## XI. 데이터 테이블

지침 10. 데이터 테이블은 테이블을 구성하는 데이터 셀의 내용에 대한 정보가 충분히 전달될 수 있어야 한다.

① 데이터 테이블은 테이블의 제목이나 테이블의 내용을 요약하여 제공해야 한다.

② 데이터 테이블은 데이터 셀 별로 대응되는 모든 헤더를 확인할 수 있도록 구성하여야 한다.

### 1. 표(table)의 종류

□ 데이터 테이블(data table)이란 표로 나타내기에 적당한 논리적인 정보들(예를 들면 연도별 예산표 또는 일주일 식단표 등)을 머리글(header) 부분과 일반 내용 부분으로 나누어 표현한 표를 말하며, 배치용 테이블(layout table)과 대조됨.

○ 데이터 테이블은 시각적으로 2차원의 격자 형식으로 표현되지만, 음성으로는 내용을 풀어서 표현되며,

○ 이 때 일반 내용 칸을 읽기 전에 그것의 머리글(header)을 짚지어서 읽어줌.

- 배치용 테이블(layout table)이란 논리적인 정보를 나타내기 위해서가 아니라
  - 단순히 시각적으로 텍스트나 그래픽을 원하는 위치에 배치하기 위해 HTML의 <table> 요소를 사용한 경우를 지칭함.
  - 접근성 측면에서 배치용 테이블을 남용하는 것은 바람직하지 않으며,
    - 보통은 스타일 시트(CSS : Cascading Style Sheet)의 위치 지정 기능(positioning)으로 대체할 수 있음.

## 2. HTML에서 데이터 테이블의 바른 사용법

- summary 속성
  - summary 속성은 비시각적인 웹 표시 장치인 화면 낭독 프로그램이나 점자 표시장치 등에게 유용한 표의 구조, 요약 정보를 제공하는 것으로,
    - summary 속성은 시각적인 브라우저에는 아무런 표시가 되지 않음.
- caption 요소

- **caption** 요소는 표의 제목을 나타내는 것으로, 시각적인 브라우저에서 표의 위 또는 아래에 표시되며, 그 위치와 모양은 스타일 시트(CSS)를 통해 조정 가능함.

#### □ 머릿글(header) 요소

- 표에서 제목줄이나 제목열에 해당하는 정보를 표시하기 위해서 사용되는 요소를 머릿글 요소라고 하며, **<thead>**, **<tfoot>**, **<th>** 등이 있음.
- 모든 테이블의 가로 줄(row)은 **<thead>**, **<tbody>**, **<tfoot>** 요소를 사용하여 크게 세 개의 그룹으로 나눌 수 있음.

```
<TABLE>
<THEAD>
 <TR> ...머릿글 정보...
</THEAD>
<TFOOT>
 <TR> ...꼬릿말 정보...
</TFOOT>
<TBODY>
 <TR> ...첫 번째 데이터 블록의 첫 번째 줄...
 <TR> ...첫 번째 데이터 블록의 두 번째 줄...
</TBODY>
```

```

<TBODY>
 <TR> ...두 번째 데이터 블록의 첫 번째 줄...
 <TR> ...두 번째 데이터 블록의 두 번째 줄...
</TBODY>
</TABLE>

```

□ 표의 머릿글(헤더)과 내용 칸(content cell)을 올바르게 짝짓기

- 표의 머릿글과 실제 데이터를 담고 있는 내용은 짝짓기가 바르게 되어야
  - 표를 표시할 수 없는 상황에서 표를 펼치거나, 화면 낭독 프로그램 등에서 표를 순서대로 읽어줄 때에도
  - 표가 제공하고자 하는 정보를 정확히 이해할 수 있음.
- 짝짓는 방법은 크게 두 가지이며,
  - 첫 번째는 머릿글에 고유한 id 속성값을 준 다음, 그것과 연관되는 내용에도 headers라는 속성을 통해 똑같은 값을 갖게 함으로써 연관성이 있다는 것을 나타낼 수 있음.
  - 두 번째는 머릿글이 영향력을 미치는 범위를 scope 속성을 이용해 표시해주는 방법임.

< 개인별 시험 성적표 >

구분	중간 고사	기말 고사
김철수	50	60
박영희	70	80

## Bad (X)

```
<table>
 <tr><td>구분<td>중간 고사<td>기말 고사
 <tr><td>김철수<td>50<td>60
 <tr><td>박영희<td>70<td>80
</table>
```

- 위의 표를 화면 읽기 프로그램으로 읽었을 때의 예

구분, 중간 고사, 기말 고사, 김철수, 50, 60, 박영희, 70, 80

## Good (O)

```
<!-- scope를 이용해 머릿글과 내용을 짝지은 예 -->

<table summary="중간 고사와 기말 고사 성적을 개인별로
보여주는 표">
 <caption>개인별 시험 성적표</caption>
 <thead>
```

```

 <tr> <th>구분
 <th scope="col">중간 고사
 <th scope="col">기말 고사
 </thead>
 <tbody>
 <tr> <th scope="row">김철수
 <td>50
 <td>60
 <tr> <th scope="row">박영희
 <td>70
 <td>80
 </tbody>
</table>

```

```

<!-- id, headers 속성을 이용해 머릿글과 내용을 짝지은 예
-->

```

```

<table summary="중간 고사와 기말 고사 성적을 개인별로
보여주는 표">

```

```

 <caption>개인별 시험 성적표</caption>

```

```

 <thead>

```

```

 <tr> <th>구분

```

```

 <th id="midterm">중간 고사

```

```

 <th id="final">기말 고사

```

```

</thead>
<tbody>
 <tr> <th id="kim">김철수
 <td headers="midterm kim">50
 <td headers="final kim">60
 <tr> <th id="park">박영희
 <td headers="midterm park">70
 <td headers="final park">80
</tbody>
</table>

```

○ 화면 읽기 프로그램으로 읽었을 때의 예

표제목 : 개인별 시험 성적표

표 머릿글 : 구분, 중간 고사, 기말 고사

표 내용 :

김철수 중간 고사 50, 김철수 기말 고사 60

박영희 중간 고사 70, 박영희 기말 고사 80

## Good (O)

- ☐ 아래 표는 머릿글 부분에 id 속성값(예: checkpoint, status 등)을 준 다음,



- 실제 내용 칸에서는 그 값을 headers에 다시 지정해줌으로써 내용 칸이 어떤 머릿글과 짝지어지는지 명확히 표시해주고 있음.

<그림 86> 머릿글과 내용을 적절하게 짝지은 표의 예

IBM Web accessibility checklist - version 3.5			
Checkpoint		Yes No Planned N/A	Comments
1	<u>Images and animations.</u> Use the alt="text" attribute to provide text equivalents for images. Use alt="" for images that do not convey important information or convey redundant information.		
2	<u>Image maps.</u> Use client-side image maps and alternative text for image map hot spots. If a server-side map is needed, provide equivalent		

※출처: IBM Accessibility Center

(<http://www-306.ibm.com/able/guidelines/web/accessweb.html>), 2004.10.3

```
<!-- id, headers 속성을 이용한 머리글과 내용 짝짓기 예-->

<table summary="IBM Web accessibility checklist"
border="0" cellpadding="2" cellspacing="2">
<tbody>
<tr>
 <td colspan="4" height="18" class="tdblue">IBM Web
accessibility checklist - version 3.5</td>
</tr>
<tr bgcolor="#99ccff">
```

```

 <th id="checkpoint" colspan="2">Checkpoint</th>
 <th id="status"><small>Yes
No
Planned
N/A</small></th>
 <th id="cmts">Comments</th>
</tr>
<tr bgcolor="#eeeeee">
 <th id="cp1" width="4%" valign="top">1</th>
 <td headers="checkpoint cp1" width="46%">Images and animations.
Use the alt="text"
attribute to provide text equivalents for images.
Use alt="" for
images that do not convey important information or
convey redundant information.
 </td>
 <td headers="status cp1"> </td>
 <td headers="cmts cp1" width="40%"> </td>
</tr><!-- 뒷 부분 생략 -->

```

## O (Good)

<그림 87> 머릿글이 미치는 범위(scope)를 명시한 표

2001년 2학기 개설 과목

학교명	교과목명	학교명	교과목명
강원대학교	웹서버 구축 II	전남대학교	인터넷 보안
	VLSI CAD		전자상거래 보안
미화여자대학교	GIS 이론 및 응용	한국방송통신대학교	웹서버 구축 I
전남대학교	네트워크 관리	호서대학교	리눅스 실습
합 계 : 8과목			

※ 출처: 정보통신 사이버대학 (<http://www.ituniv.or.kr/html/info03-1.html?pMode=con>), 2004.

10. 3

```
<!-- scope를 이용한 머릿글과 내용 칸 짝짓기 예 -->

<table border="0" cellpadding="0" cellspacing="0"
width="535">
<caption>2001년 2학기 개설 과목</caption>
<tbody>
 <tr>
 <th scope="col">학교명</th>
 <th scope="col">교과목명</th>
 <th scope="col">학교명</th>
 <th scope="col">교과목명</th>
 </tr>
 <tr>
 <td rowspan="2" class="univ">강원대학교</td>
 <td class="course">웹서버 구축 II</td>
```

```

 <td rowspan="2" class="univ">전남대학교</td>
 <td class="course">인터넷 보안</td>
 </tr>
 <tr>
 <td class="course">VLSI CAD</td>
 <td class="course">전자상거래 보안</td>
 </tr>

 <!-- 중간 생략됨. -->
 </tbody>
</table>

```

□ 위의 표는 머릿글(학교명, 교과목명 등)의 <th> 요소에 대해 scope라는 속성값을 넣어줌으로써,

- 머릿글이 어떤 내용에까지 범위가 미치는지를 명확히 지정하고 있으며, caption 요소를 이용해 표의 제목을 제공한다.




## X (Bad)

<그림 88> 표를 이용한 데이터의 표현

서울.경기 광역예보		
오늘 (15일)	내일 (16일)	모레 (17일)
서 - 북서 다소 강	남서 - 서	남서 - 서
		

※ 출처: 기상청(<http://www.hma.go.kr>), 2004. 11. 26

<그림 89> 짝짓기가 안 된 머릿글과 내용

<td><td>		
서울.경기 광역예보		
</table><table summary="">		
<td> 오늘 (15일)	<td> 내일 (16일)	<td> 모레 (17일)
<td> 서 - 북서	<td> 남서 - 서	<td> 남서 - 서
<td> 	<td> 	<td> 
<td> 맑은 후 밤늦 게 구름 많아짐	<td> 구름 많음	<td> 구름 많음
</table>		

- 위의 표에서 시각 장애인은 어떤 것이 15일의 날씨, 어떤 것이 17일의 날씨인지 알기 힘들.

## O (Good)

<그림 90> 표를 바르게 수정한 예

<table summary="3일간 대체로 구름 많음">		
<th scope="col" id="d01"> 오늘 (15일)	<th scope="col" id="d02"> 내일 (16일)	<th scope="col" id="d03"> 모레 (17일)
<td headers="d01"> 서 - 북서 다소 강	<td headers="d02"> 남서 - 서	<td headers="d03"> 남서 - 서
<td headers="d01"> 구름조금	<td headers="d02"> 구름많음	<td headers="d03"> 구름많음
</table>		

- 위의 날씨를 나타내는 표에서는 오늘, 내일, 모레와 같은 날짜를 나타내는 줄(행)이 머릿글이므로 <td> 대신에 <th> 요소를 사용하여 수정하는 것이 좋음.
- 각 머릿글 셀 <th>에는 해당 머릿글이 세로(column)로 적용되는지 가로로(row) 적용되는지에 따라 scope="col" 또는 scope="row"와 같은 속성을 넣어줌.
  - 또한 세로로(column) 나열된 셀들이 가장 위의 머릿글 날짜에 해당되는 날씨라는 것을 확실하게 표현하기 위해
  - 머릿글에는 id를 부여하고 밑에 나오는 열(column)에 나열된 셀들은 공통의 headers 값을 공유하고 있음
  - 마지막으로 데이터를 나타내는 표에는 표의 요약용 summary라는 속성값으로 넣어주는 것이 좋음

## XII. 콘텐츠의 배치

검사항목 11. 콘텐츠의 모양이나 배치는 논리적으로 이해하기 쉽게 구성하여야 한다.

① 문서의 모양이나 콘텐츠의 배치를 위해서는 스타일 시트(style sheet)를 사용하여야 한다.

② 배치용 테이블을 사용하여 콘텐츠의 모양이나 배치를 할 경우에는 테이블을 구성하는 모든 셀들을 왼쪽상단에서 오른쪽 하단에 이르는 순서대로 늘어놓았을 때에도 그 내용을 충분히 이해할 수 있도록 구성해야 한다.

### 1. 스타일 시트와 테이블을 이용한 배치 비교

#### ☐ 테이블을 이용한 배치(layout)

- 테이블은 하나의 개체로 취급되어, 테이블 내의 모든 요소들이 로드되기 전까지는 내용물이 나타나지 않으므로 일반적으로 브라우저에서 느리게 표시되며,
- 내용물의 배치를 위해 테이블 안에 중복적으로 여러 테이블을 넣는 것은 코드를 이해하기 어렵게 만들고,
  - 또한 개발 후에도 유지보수 애로점이 있음.

- 테이블을 이용하여 내용물의 배치를 미세하게 조정하기 위해 특정한 픽셀 크기의 투명한 그림을 넣기도 하는데,
- 이로 인해 더욱 속도가 느려지거나, 무의미한 그림으로 인한 코드의 혼란, 절대적인(absolute) 크기 단위 사용으로 인한 디자인의 적응성 결여 등의 문제가 생길 수 있음.

#### □ 스타일 시트를 이용한 배치

- 스타일 시트를 이용해 내용물을 배치할 경우,
- 내용물의 구조와 내용을 건드리지 않은 채로 배치 위치나 표현 방식만 바꿀 수 있어 구조와 표현을 효과적으로 분리할 수 있음.
- 스타일 시트를 이용해 내용을 배치하면 일반적으로 코드의 양이 작아지고, 구조가 간결해지며,
- 다른 사람이 보아도 이해하기가 쉬워지고, 후에 내용을 수정 보완하거나 또는 내용을 그대로 놔둔 채로 모양이나 배치만 바꾸는 것이 매우 쉬운 장점이 있음.

#### ※ 스타일 시트를 이용한 배치에 관한 참고 자료

- CSS 2.1의 Visual formatting model  
(<http://www.w3.org/TR/CSS21/visuren.html>), 2004. 12. 26



- ☐ Visual formatting model의 한국어 번역본  
(<http://trio.co.kr/webrefer/css2/visuren.html>), 2004. 12. 26
- ☐ Glish.com : CSS Layout techniques  
(<http://glish.com/css/home.asp>), 2004. 12. 26
- ☐ The Layout Reservoir  
(<http://www.bluerobot.com/web/layouts/>), 2004. 12. 26
- ☐ Little boxes  
([http://www.thenoodleincident.com/tutorials/box\\_lesson/boxes.html](http://www.thenoodleincident.com/tutorials/box_lesson/boxes.html)), 2004. 12. 26
- ☐ Position is Everything  
(<http://www.positioniseverything.net/>), 2005. 03. 01

※ 스타일 시트를 이용해 내용물을 배치한 사이트의 예

- ☐ World Wide Web Consortium (<http://www.w3.org/>), 2004. 12. 10
- ☐ SitePoint.com (<http://www.sitepoint.com>), 2004. 12. 10
- ☐ The Web Standards Project  
(<http://www.webstandards.org/>), 2004. 12. 26
- ☐ CSS Zen Garden: The Beauty in CSS design  
(<http://www.csszengarden.com/>), 2004. 12. 26

- ☐ ESPN.com (<http://espn.go.com/>), 2004. 12. 26
- ☐ Yahoo! (<http://www.yahoo.com/>), 2004. 12. 26
- ☐ Opera Web Browser (<http://www.opera.com/>), 2004. 12. 26
- ☐ 한글 Mozilla 프로젝트 (<http://www.mozilla.or.kr/>), 2004. 12. 26
- ☐ Jeffrey Zeldman Presents The Daily Report (<http://www.zeldman.com/>), 2004. 12. 30
- ☐ Wired News (<http://www.wired.com/>), 2004. 12. 30
- ☐ MSN (<http://www.msn.com/>), 2005. 02. 02
- ☐ ABC News (<http://abcnews.go.com/>), 2005. 01. 15

※ 스타일 시트를 이용한 내용과 구조의 분리 예:

### **CSS ZenGarden**

- ☐ CSS Zen Garden은 디자이너들로 하여금 HTML 본문은 전혀 건드리지 않고 오로지 CSS(Cascading Style Sheet)만을 활용하여 디자인을 극적으로 바꿀 수 있다는 것을 보여줌
  - CSS만을 활용함으로써 아래의 예와 같이 HTML 원본 문서는 100% 동일한 문서를 전혀 다른 모양으로 표현이 가

능함.

- 이렇게 함으로써 원본 문서는 구조적으로 견고하고 표준 문법에 맞으며, 장애인의 접근성이 매우 높은 문서가 되고,
  - CSS는 오로지 시각적인 디자인을 입히는 역할만을 하게 됨.
- 아래 문서들은 모두 CSS 표준, XHTML 표준, 미국의 재활법 508조의 규정, W3C WAI(Web Accessibility Initiative)의 웹 콘텐츠 접근성 적합성의 최상위 수준인 AAA를 모두 만족하면서도 시각적으로는 다양한 모습을 보여줌
- 특히, CSS의 배치(layout) 기능을 이용하여 내용물의 배치가 시각적으로는 상당히 달라지지만,
  - 스타일 시트를 제거했을 때에는 일정한 구조적인 순서로 표시되므로, 화면 낭독 프로그램 등을 사용하는 시각 장애인에게도 순서의 혼란을 주지 않음.

<그림 91> CSS Zen Garden: 스타일 시트를 제거했을 때



※ 출처: css Zen Garden (<http://www.csszengarden.com>), 2004. 12. 27

<그림 92> CSS Zen Garden: 기본 스타일 적용



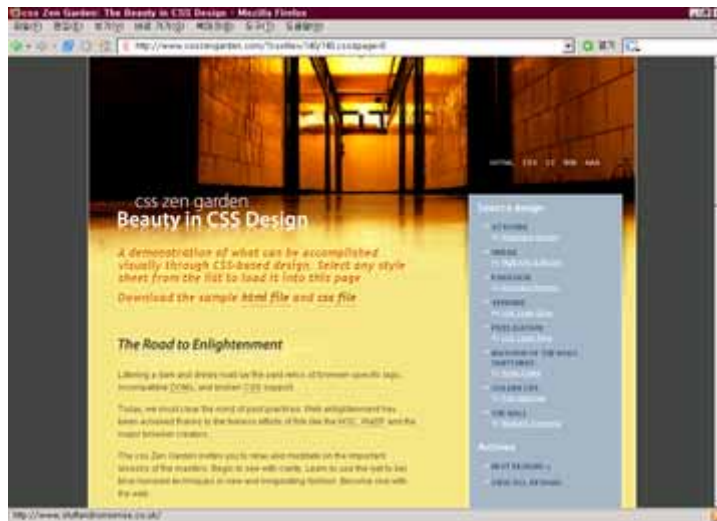
<그림 93> CSS Zen Garden: Paravion 스타일 적용



<그림 94> CSS Zen Garden: Golden Cut 스타일 적용



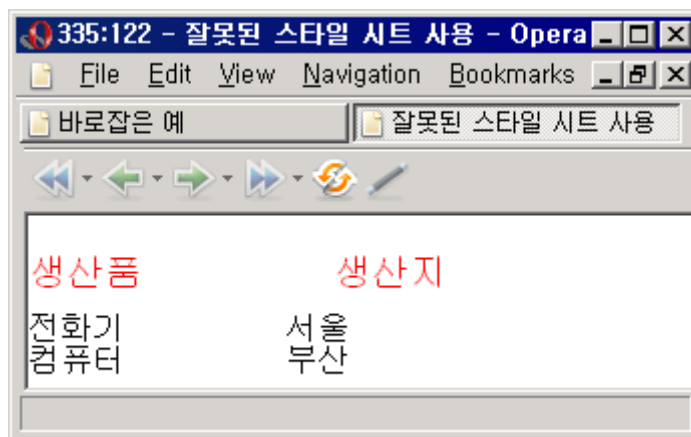
<그림 95> CSS Zen Garden: The hall 스타일 적용



## 2. 스타일 시트의 바른 사용

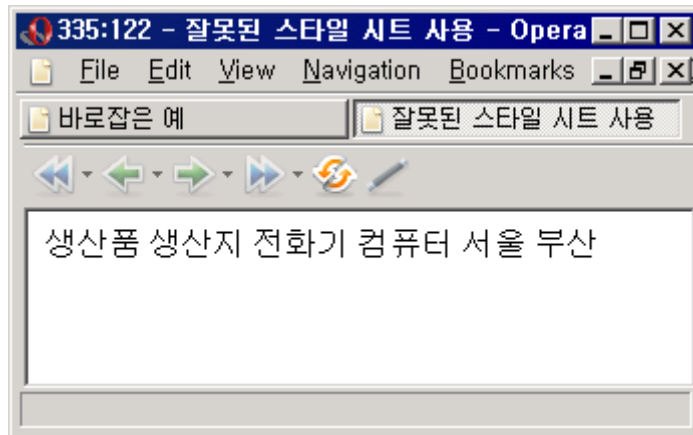
**X (Bad)**

<그림 96> 잘못된 스타일 시트의 사용



- 스타일 시트를 이용해서 내용물의 위치를 지정할 때에는 스타일 시트를 제거했을 때에도 논리적으로 이해가 가능해야 함.

<그림 97> 스타일 시트를 빼면 순서가 어긋나는 경우



- 스타일 시트를 제거했을 때 어떤 것이 생산품이고 어떤 것이 생산지인지 구별할 수 없음.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8">
<style type="text/css">
.menu1 { position: absolute; top: 1em; left: 0;
color: red; background-color: white }
```

```

.menu2 { position: absolute; top: 1em; left: 8em;
color: red; background-color: white }

.item1 { position: absolute; top: 3em; left: 0;
margin: 0}

.item2 { position: absolute; top: 4em; left: 0;
margin: 0}

.item3 { position: absolute; top: 3em; left: 8em;
margin: 0}

.item4 { position: absolute; top: 4em; left: 8em;
margin: 0}

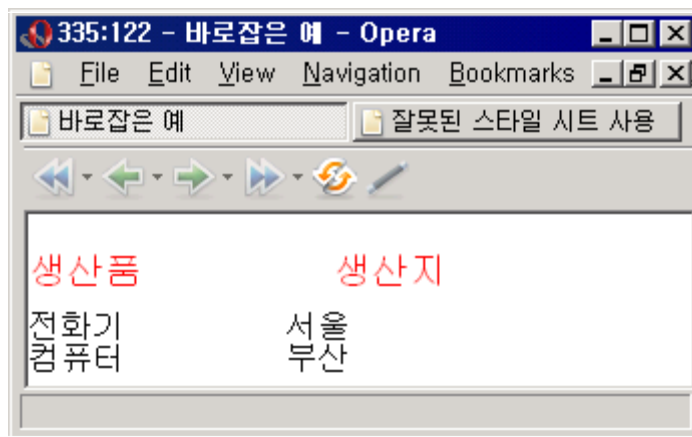
</style>
<TITLE>잘못된 스타일 시트 사용</TITLE>
</head>
<body>
생산품
생산지
전화기
컴퓨터
서울
부산
</body>
</HTML>

```



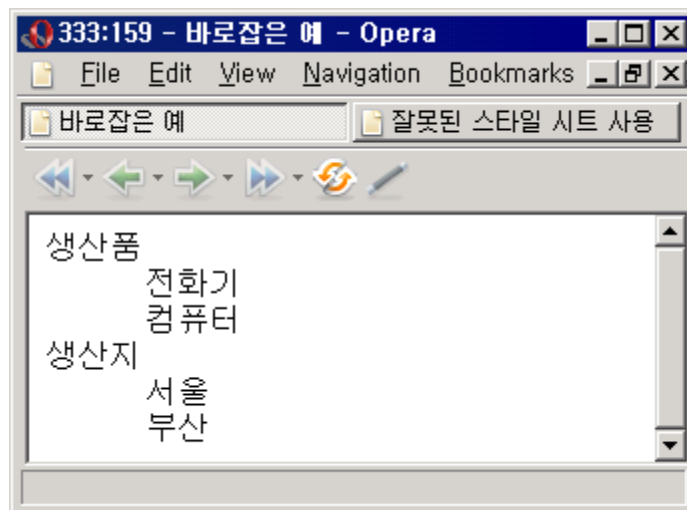
## O (Good)

<그림 98> 스타일 시트를 바로잡은 예



- 스타일 시트를 써서 내용을 배치했을 때에 스타일 시트가 제거되었을 때 내용이 논리적으로 이해가 가능한지 주의해야 함.

<그림 99> 스타일을 빼도 논리적 이해가 가능한 경우



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">

<HTML>

<head>

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

<head>

<style type="text/css">
.menu1 { position: absolute; top: 1em; left: 0;
font-size: 120%; color: red; background-color: white
}
.menu2 { position: absolute; top: 1em; left: 8em;
font-size: 120%; color: red; background-color: white
}
.item1 { position: absolute; top: 3em; left: 0em;
margin: 0}
.item2 { position: absolute; top: 4em; left: 0em;
margin: 0}
.item3 { position: absolute; top: 3em; left: 8em;
margin: 0}
.item4 { position: absolute; top: 4em; left: 8em;
margin: 0}
</style>
```

```

<TITLE>바로잡은 예</TITLE>
</head>
<body>
<dl>
<dt class="menu1">생산품</dt>
 <dd class="item1">전화기</dd>
 <dd class="item2">컴퓨터</dd>
<dt class="menu2">생산지</dt>
 <dd class="item3">서울</dd>
 <dd class="item4">부산</dd>
</dt>
</dl>
</body>
</HTML>

```

□ 그림 112과 같이 생산품, 생산지가 여러 개가 있어서 복잡한 짝짓기가 필요한 데이터를 표현하기 위한 가장 좋은 방법은 아래와 같이 표(table)를 이용하는 것임.

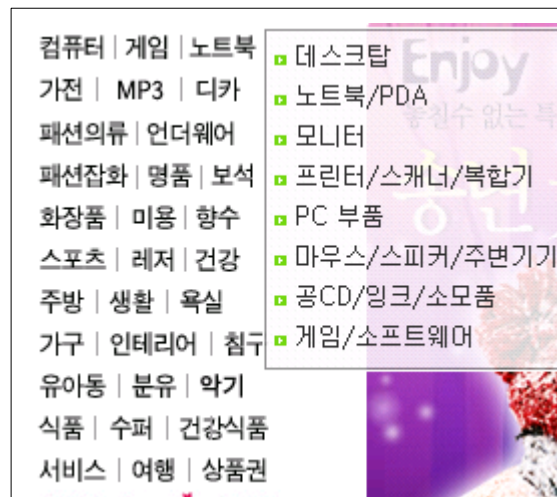
- 단, 표를 이용할 때에 생산품, 생산지와 같은 제목에 해당하는 부분은 반드시 <th> 요소를 사용해야 함.
- 이렇게 함으로써 시각 장애인용 화면 낭독 프로그램에서도 “생산품, 전화기, 생산지, 서울, 생산품, 컴퓨터, 생산지, 부산”과 같은 순서로 바르게 읽어주어 표의 내용을 이해할 수 있게 됨.

## O (Good)

```
<table summary="각 생산품별 생산지를 보여주는 표">
<caption>생산품과 생산지</caption>
<tr>
 <th scope="col">생산품
 <th scope="col">생산지
<tr>
 <td>전화기
 <td>서울
<tr>
 <td>컴퓨터
 <td>부산
</table>
```

## X (Bad)

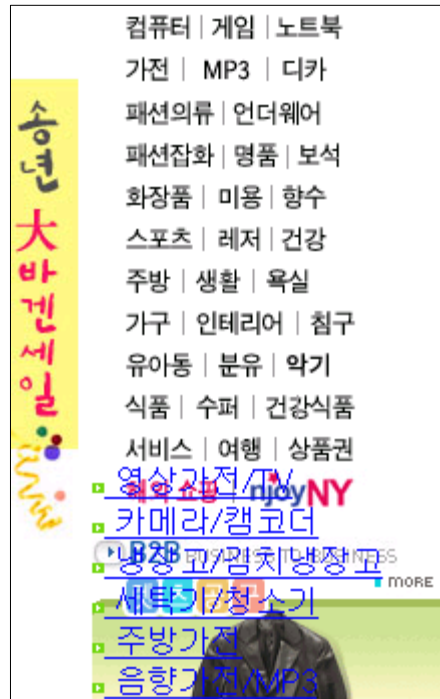
<그림 100> 잘못된 스타일 시트 사용 예



※ 출처: KTmall (<http://www.ktmall.com>), 2004. 11. 26

## X (Bad)

<그림 101> 스타일 시트를 빼면 어긋나는 메뉴



- 위의 전자상거래 사이트에서는 왼쪽의 상품 카테고리 부분에 마우스를 갖다 대면 하위 상품 카테고리가 보이는 방식을 사용하고 있음.

- 이 때 하위 메뉴의 위치는 스타일 시트를 이용해 절대적인 위치값을 갖도록 해놓아 하위 메뉴의 접근이 어려움.

- 따라서 스타일 시트를 제거했을 때나, 스타일 시트를 지원할 수 없는 상황에도 내용물이 정상적으로 표시되고,

- 사용자가 이해할 수 있도록 시각적으로뿐 아니라, 논리/구조적 마크업 (H1, H2, UL, DL, LI 등)을 사용하여 문서를 재구성해야 함.

### 3. 배치용 테이블(layout table)

- 배치용 테이블에는 summary, caption, thead, tbody, tfoot, th 등 의미적인 마크업을 붙이지 않는 것이 좋음
- 배치용 테이블의 경우, 테이블 안의 내용을 순서대로 펼쳤을 때(linearized) 타당하게 이해할 수 있어야 함.

#### **X (Bad)**

<그림 102> 양식 입력시 테이블을 잘못 사용한 예

Household Income Range:	Industry in which you work:
<input type="text" value="Select One"/>	<input type="text" value="Select One"/>
Job Title:	Job Function:
<input type="text" value="Select One"/>	<input type="text" value="Select One"/>

※ 출처: New York Times (<http://www.nytimes.com/auth/login?URI=http://>), 2004. 11. 01

- 위의 그림과 같이 테이블을 써서 양식(form) 입력을 받을 경우, 양식의 논리적인 순서에 주의해야 함.

- 아래 그림과 같이 테이블이 해제되었을 때에 논리적인 순서가 흐트러질 수 있음.

## **X (Bad)**

<그림 103> 테이블을 없애면 순서가 뒤바뀐 예

Household Income Range:	
Industry in which you work:	
—Select One—	▼
—Select One—	▼
Job Title:	
Job Function:	
—Select One—	▼
—Select One—	▼

- <그림 115>은 양식을 입력받을 때에 양식의 레이블(label)과 실제 양식이 테이블의 칸(cell)을 이용하여 상하로 배치되어 있음.
- 그러나 이 경우, 화면 음성 낭독 프로그램이나 테이블을 지원하지 못하는 텍스트 브라우저와 같이 테이블을 순차적으로 읽는 방식의 프로그램에서 테이블을 풀어 헤치면,
  - Household Income Range, Industry in which you work와 같이 연속적으로 레이블이 나오고, 해당 양식은 밑에 한꺼번에 몰려서 나오는 문제가 생김.
- 따라서, 양식을 입력받을 때에 테이블을 사용할 때에는 테이블을 왼쪽, 오른쪽 칸 순으로 펼쳤을 때에도 논리적으로



타당한지 점검해보아야 함.

- 더욱 바람직한 방법은 <label> 요소를 사용하여 양식의 레이블과 양식을 짝지어 주는 것으로 ‘XIII. 온라인 양식 (form)’에 자세한 설명이 있음.

## O (Good)

<그림 104> 테이블 대신 CSS로만 내용물 배치를 한 사례



※ 출처: 재정경제부 함께 풀어가는 종합 투자 계획 (<http://jumpkorea.mofe.go.kr>), 2004.  
12. 20

O (Good)

<그림 105> CSS를 제거했을 때 모습



- 위 사례의 첫 번째 그림에서는 테이블을 사용하지 않고 CSS(Cascading Style Sheet)로만 내용물을 배치하여 테이블 사용에 따른 복잡함을 없앤 경우로,
  - 두 번째 그림에서와 같이 문서의 스타일 시트나 디자인 요소를 제거해도 잘 짜여진 논리적인 구조는 그대로 남아 있음을 알 수 있음.

## XIII. 온라인 양식(form)

지침 12. 온라인 양식을 포함하는 콘텐츠는 양식 작성에 필요한 정보, 양식 구성 요소, 필요한 기능, 작성 후 제출 과정 등 양식과 관련한 모든 정보를 제공해야 한다.

① 온라인 양식을 구성하는 모든 양식 제어요소, 예를 들면, 에디트 박스(edit box), 라디오 버튼(radio button), 체크 박스(check box) 등은 레이블과 해당 양식 제어요소간의 표시 순서가 일정하여야 한다.

② 탭(tab)키를 이용하여 양식 제어요소 간을 이동할 경우에 그 순서가 왼쪽 위에서 오른쪽 아래 부분으로 순차적인 이동이 가능하여야 한다.

③ 양식을 구성할 때에 보조기술을 사용하면 접근이 가능한 경우에만 스크립트, 애플릿, 플러그인 또는 다른 프로그램 요소를 사용한다. 만일 이것이 불가능할 경우에는 스크립트, 애플릿(applet), 플러그인(plug-in) 등 프로그램 요소의 기능을 정지시켜도 온라인 양식을 작성할 수 있어야 한다.

### 1. 양식의 접근성

- 웹에서는 양식을 통해 사용자로부터 다양한 방법으로 값을 입력받아, 그 결과를 웹 서버에 전송하고 그 결과에 따라서 다른 페이지를 보여줄 수 있게 되므로, 가장 기본적인 상호작용 방법으로 활용됨.

- 일반적으로 장애인들에게는 양식에 자료를 입력하는 것이 보통의 웹을 탐색하는 것보다 훨씬 어려운 일이며,
  - 양식의 이해와 자료 입력에 비장애인보다 훨씬 많은 시간을 소요하게 되므로,
- 웹 페이지에 양식을 삽입할 때에는,
  - 양식의 현재 위치, 빠른 이동, 정확한 맥락 제공, 오류 방지, 작업의 취소, 충분한 시간적 여유 제공 등 장애인들의 접근성을 향상시키기 위한 문제를 신중하게 고려해야 함.

## 2. 각 컨트롤에 정확한 이름 붙이기(LABEL)

- 컨트롤에 정확한 이름표(label)를 붙여줌으로써 시각장애인 처럼 컨트롤과 그 컨트롤의 제목을 짚짓기 힘든 사람들에게 양식을 쉽게 사용할 수 있도록 함.
  - 이름표는 해당하는 컨트롤과 시각적으로 잘 짚짓기가 되어야 하는 것은 물론이지만 이것만으로는 부족하며,
  - 반드시 `<label for="ID값">` 요소를 사용하여 해당하는 컨트롤과 명시적으로 짚짓기를 해주어야 하며, 이것을 명시

적인 레이블링(explicit labeling)이라고 함.

□ 이렇게 명시적으로 레이블링을 함으로써 시각적으로 양식을 보는 비장애인들에게도 이득을 줄 수 있음.

- 즉, 직접 컨트롤을 선택하지 않아도 단지 레이블만을 선택하는 것만으로도 그것과 짝지어진 컨트롤(예: 라디오 버튼, 텍스트 필드 등)을 선택할 수 있는 장점이 있음.

## O (Good)

<그림 106> 라디오 버튼과 레이블을 짝지은 예



```
<label for="rb_showposts_0"><input type="radio"
name="showposts" value="0" id="rb_showposts_0"
checked="checked" />Threads</label>
```

```
<label for="rb_showposts_1"><input type="radio"
name="showposts" value="1" id="rb_showposts_1"
/>Posts</label>
```

□ 위의 예에서는 <label> 요소가 <input>이라는 컨트롤을 완전히 둘러싸는 형태로,

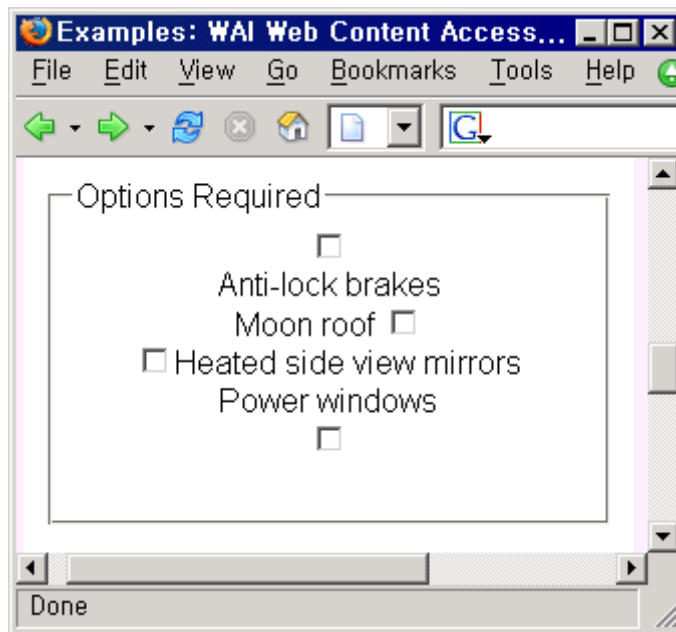
- 라디오 버튼을 직접 누르지 않고 Posts라는 레이블만 눌

러도 해당 라디오 버튼을 선택하는 것과 동일한 결과를 가져옴.

- W3C 표준에서는 위와 같이 하기보다는, 단지 label을 정확히 지정해주는 것이 좋다고 권장하고 있음.

### △ (Good)

<그림 107> 시각적 결함을 <label>로 보완한 예



```
<FIELDSET>
```

```
<LEGEND align="top">
```

```
Options Required
</LEGEND>

<INPUT TYPE="CHECKBOX" NAME="opt1" ID="opt1"
VALUE="abs">

<LABEL for="opt1">Anti-lock breaks</LABEL>
<LABEL for="opt2">Moon roof </LABEL>

<INPUT TYPE="CHECKBOX" NAME="opt2" ID="opt2"
VALUE="mr">

<INPUT TYPE="CHECKBOX" NAME="opt3" ID="opt3"
VALUE="hm">
<LABEL for="opt3">Heated mirrors</LABEL>
<LABEL for="opt4">Power windows</LABEL>

<INPUT TYPE="CHECKBOX" NAME="opt4" ID="opt4"
VALUE="pw">
</FIELDSET>
```

※ 출처: WAI WCAG Curriculum (<http://www.w3.org/WAI/wcag-curric/label.htm>), 2004. 11.  
10

- 위의 예에서는 시각적으로 체크 상자가 먼저 나오거나 또는 체크 상자의 제목이 먼저 나오거나 하여 일관성이 없게 양식이 작성되었지만,



- 다행히 <label> 요소를 이용하여 체크 상자와 그 제목을 명시적으로 짝지어놓았기 때문에 장애인들이 사용하는 데에 큰 문제는 없음.

## X (Bad)

<그림 108> 레이블을 명시하지 않아 장애인에게 선택을 어렵게 한 예

<input type="checkbox"/> 스포츠(구기)	<input type="checkbox"/> 등산	<input type="checkbox"/> 낚시	<input type="checkbox"/> 경기관람
<input type="checkbox"/> 독서	<input type="checkbox"/> 미술	<input type="checkbox"/> 바둑, 장기	<input type="checkbox"/> 사진
<input type="checkbox"/> 수집,스크랩	<input type="checkbox"/> 여행(산책)	<input type="checkbox"/> 영화감상	<input type="checkbox"/> 요리
<input type="checkbox"/> 음악감상	<input type="checkbox"/> 인터넷	<input type="checkbox"/> 연극관람	<input type="checkbox"/> 미술(감상)
<input type="checkbox"/> 악기연주	<input type="checkbox"/> 레포츠	<input type="checkbox"/> 골프	<input type="checkbox"/> 볼링
<input type="checkbox"/> 스키	<input type="checkbox"/> 스쿠버, 수상스키	<input type="checkbox"/> 수영	<input type="checkbox"/> 승마
<input type="checkbox"/> 테니스	<input type="checkbox"/> 꽃꽂이	<input type="checkbox"/> 분재	<input type="checkbox"/> 서예
<input type="checkbox"/> 애완동물기르기	<input type="checkbox"/> PC게임	<input type="checkbox"/> 기타	

※ 출처: 인터넷 우체국 회원 가입 페이지(<http://www.epost.go.kr/>), 2004. 11. 26

```
<input name="hobby" value="01" style="border: 0pt none ;" onclick="chkHobbyCnt(this.form, this)" type="checkbox"> 스포츠 (구기)
```

- 위의 예에서는, 시각적으로는 어떤 체크박스가 어떤 항목과 연관되는지 어느 정도 구분이 되지만,

- 이것을 음성으로 연속적으로 들을 경우, 해당 체크박스가 앞의 것과 연관되는지 뒤의 것과 연관되는지 알기 어려움.

- 따라서, 아래와 같이 각각의 체크박스에 서로 다른 id를 부

여하고, 그것과 짝지워지는 텍스트에는 <label>을 붙임으로써 접근성이 높아짐.

## O (Good)

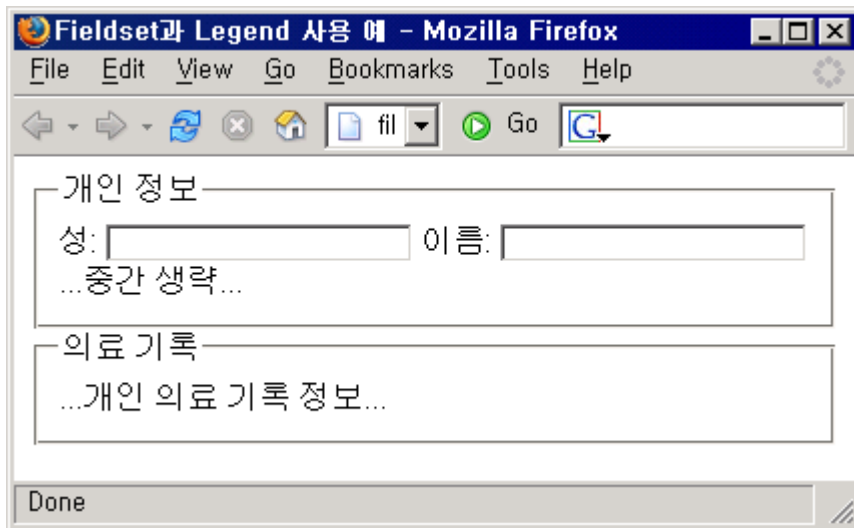
```
<input name="hobby" id="sports" value="01"
style="border: 0pt none ;"
onclick="chkHobbyCnt(this.form, this)"
type="checkbox"> <label
for="sports">스포츠 (구기)</label>
```

### 3. 유사한 컨트롤의 묶음 (FIELDSET)

- 회원 가입, 구독 신청 등과 같이 많은 정보를 양식을 통해 입력받을 때에는 적절하게 비슷한 컨트롤들을 묶어줌으로써
  - 순차적으로 양식을 입력해야 하는 장애인들에게는 현재의 맥락을 쉽게 파악하고, 양식 입력의 오류를 줄일 수 있게 함.
- 유사한 컨트롤들은 fieldset 요소를 통해 묶어주고, legend 요소를 이용해 제목을 붙임.

## O (Good)

<그림 109> <fieldset>을 사용하여 유사 컨트롤을 하나로 묶은 예



□ 위 예에서는 개인의 신상 정보와 개인의 의료 기록에 대한 정보를 입력하는 부분으로 나눈 다음,

- 각 부분에 속하는 컨트롤들을 fieldset으로 묶어 구분하고, legend를 통해 구분된 영역에 제목을 붙임.

```
<form action="http://example.com/adduser"
method="post">
 <fieldset>
 <legend>개인 정보</legend>
 <label for="lastname">성:</label>
 <input type="text" id="lastname" tabindex="1" />
```

```

<label for="firstname">이름:</label>
<input type="text" id="firstname" tabindex="2"
/>

...중간 생략...

</fieldset>

<fieldset>
 <legend>의료 기록</legend>
 ...개인 의료 기록 정보...
</fieldset>
</form>

```

#### 4. 유사한 목록의 묶음 (OPTGROUP)

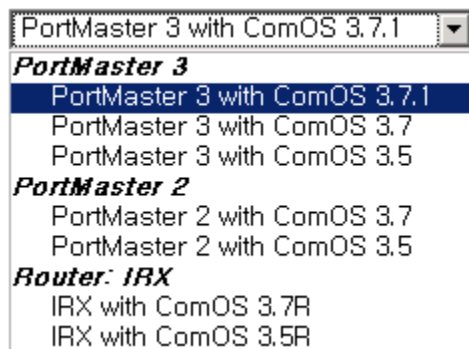
- SELECT 요소를 이용해 목록 상자(list box)를 만들 때에 목록이 많아지면, 한꺼번에 목록을 볼 수 없는 시각장애인들은 전체 목록의 개수도 알 수 없고,
  - 전체를 조감할 수 없으므로 현재까지 들은 목록 중에 하나가 최상의 선택인지 결정하기가 어려움.
- 일반인들도 지나치게 목록이 많아지면 정확히 원하는 항목을 선택하기가 어려워지며,
  - 이런 어려움을 부분적으로 해결해주는 것이 유사한 항목

을 그룹으로 묶어주는 `optgroup` 요소임.

- 아래의 예에서는 7개의 항목을 `optgroup` 요소를 사용하여 세 개의 범주로 분류한 것이며, 선택을 구조적으로 쉽게 할 수 있도록 함.

## O (Good)

<그림 110> 목록상자 내의 유사 항목을  
<optgroup>으로 묶은 예



```
<SELECT name="ComOS">
 <OPTGROUP label="PortMaster 3">
 <OPTION label="3.7.1" value="pm3_3.7.1">
PortMaster 3 with ComOS 3.7.1
 <OPTION label="3.7" value="pm3_3.7"> PortMaster 3
with ComOS 3.7
 <OPTION label="3.5" value="pm3_3.5"> PortMaster 3
with ComOS 3.5
 </OPTGROUP>
```

```

<OPTGROUP label="PortMaster 2">
 <OPTION label="3.7" value="pm2_3.7">PortMaster 2
with ComOS 3.7
 <OPTION label="3.5" value="pm2_3.5">PortMaster 2
with ComOS 3.5
</OPTGROUP>
<OPTGROUP label="Router: IRX">
 <OPTION label="3.7R" value="IRX_3.7R">IRX with
ComOS 3.7R
 <OPTION label="3.5R" value="IRX_3.5R"> IRX with
ComOS 3.5R
</OPTGROUP>
</SELECT>

```

※ 출처: WAI WCAG Curriculum에서 약간 변형  
<http://www.w3.org/WAI/wcag-curric/optgrp.htm>, 2004. 11. 10

## 5. 양식의 키보드 접근성

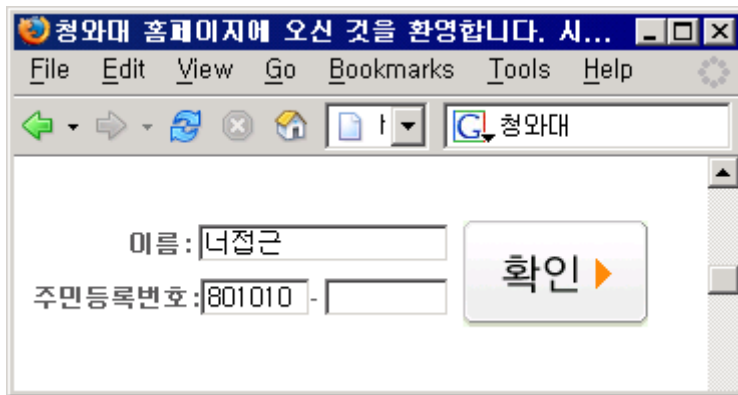
□ 마우스를 사용할 수 없는 시각 장애인, 지체 장애인들을 위해 양식의 값을 채워넣거나, 이동하거나, 실수를 정정하거나, 최종 제출하는 모든 관련 행위가 키보드를 통해 이루어질 수 있어야 함.

□ 키보드를 통해서 양식을 탐색할 수 있게 함으로써,

- 비장애인들에게도 더 정확하고 빠르게 양식의 값을 입력하고, 오류를 줄이고, 제출할 수 있게 되어 이득을 줌.

## X (Bad)

<그림 111> 잘못된 양식 사용 예



※ 출처: 청와대 홈페이지 회원 가입

(<http://www.cwd.kr/cwd/kr/mem/precek.php>), 2004. 10. 31

```
<!-- 앞부분 생략... 이름 필드-->
이름 : <input name="name"
type="text" size="16" maxlength="40" hname="이름"
required maxbyte="40">
<!-- 중간 생략... 확인 버튼-->

<!-- 중간 생략... 주민등록번호 입력 필드-->
주민등록번호 :<INPUT maxLength=6
```

```
size=6 name=resident_code_1 hname="주민등록번호1"
required maxbyte="6" value=''>
-
<INPUT maxLength=7 size=7 name=resident_code_2
hname="주민등록번호2" required maxbyte="7" value=''>
```

- 위의 예에서 잘못된 부분은 label이 없는 것으로,
  - 텍스트 필드에는 적절한 식별자(id)를 붙이고, 그 식별자에 맞는 label을 <label for="id"> 형식으로 반드시 붙여야 함.
- 주민등록번호 입력에 대해서는 자세한 설명이 필요
  - 주민등록번호의 경우, 앞자리와 뒷자리를 한꺼번에 입력해야 하는지, 아니면 나누어서 입력해야 하는지
    - 시각장애인은 알 수가 없으므로, 앞부분과 뒷부분이 나뉘어져 있다는 사실을 숨겨진 텍스트(<span style="display:none">)를 통해 알려주거나,
    - 각각의 텍스트 필드에 title="텍스트 필드 설명"을 삽입해 줌으로써 해당 텍스트 필드에 무슨 값을 넣어야 하는지 확실하게 알 수 있게 됨.
- 잘못된 속성 사용 예 : <input> 요소의 속성에 hname,



required, maxbyte라는 속성은 HTML 표준 문법에 없으며, 잘못된 것임.

#### □ 확인 버튼의 유형 지정 필요

- 통상적으로 확인 버튼으로 이동하지 않아도 이름과 주민등록번호를 입력 후에 키보드의 [Enter] 키만을 누름으로써 데이터가 제출(submit)되도록 콘텐츠를 제작해야 함.
- 이미지에 단순한 하이퍼링크를 거는 것(<a><img ...></a>)은 그것이 양식에 입력된 값들을 최종 제출(submit)하는 버튼과는 다르므로,
  - 그 링크를 마우스로 누르거나 키보드로 포커스를 준 다음 [Enter]를 누르는 것과 같이 명시적으로 그 링크를 활성화시켜주지 않으면 제출이 되지 않는 문제가 발생함.
- 따라서 제출(submit) 버튼은 <input type="submit" ...>과 같이 명확히 유형을 지정해주거나,
  - 또는 이미지를 버튼으로 사용할 경우에도 <input type="image" ...>와 같이 지정하여야만 값을 입력하는 도중에, 또는 모든 값을 입력한 후에 [Enter] 키를 누름으로써 양식의 값을 제출이 가능함.

#### □ 키보드 이동 순서 지정

○ 아울러 키보드를 이용해 정확한 순서대로 값을 입력할 수 있도록 `tabindex="값"`을 넣어주어야 하며,

- 이러한 경우, `tabindex`의 값은 반드시 1, 2, 3 등과 같이 연속된 숫자일 필요는 없으며, 10, 20, 30 등과 같이 숫자의 순서만 맞으면 됨.

□ 위의 사례를 바르게 수정하면 아래와 같이 됨.

## O (Good)

```
<!--이름 필드: label 지정, 탭 순서 지정-->
<label for="name">이름 :</label> <input name="name"
id="name" type="text" size="16" maxlength="40"
tabindex="10">

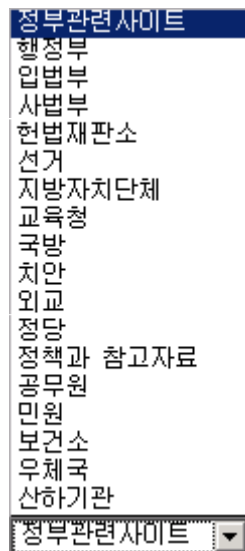
<!--확인 버튼: 양식의 유형을 이미지로 지정-->
<input type="image" name="submit" src="imgs_new/1_btn_ok.gif"
alt="확인" onclick="goLogin(); return false">

<!--주민등록번호: label 지정, 보이지 않는 설명 추가-->
<label for="pid1">주민등록번호 :</label>
주민등록번호 앞자리
<INPUT maxLength=6 size=6 name=resident_code_1
title="주민등록번호 앞자리" tabindex="20">
```

```
-
주민등록번호 뒷자리
<INPUT maxLength=7 size=7 name=resident_code_2
id="pid2" title="주민등록번호 뒷자리" tabIndex="30">
```

## X (Bad)

<그림 112> 이동 버튼이 따로 없는 잘못된 목록 상자(list box)



```
<select name="gov_site" class=exp_select
ONCHANGE="goGovSite(this.options[this.selectedIndex
].value)">
```

※ 출처: 대한민국 전자정부 (<https://www.egov.go.kr/default.html>), 2004. 11. 01

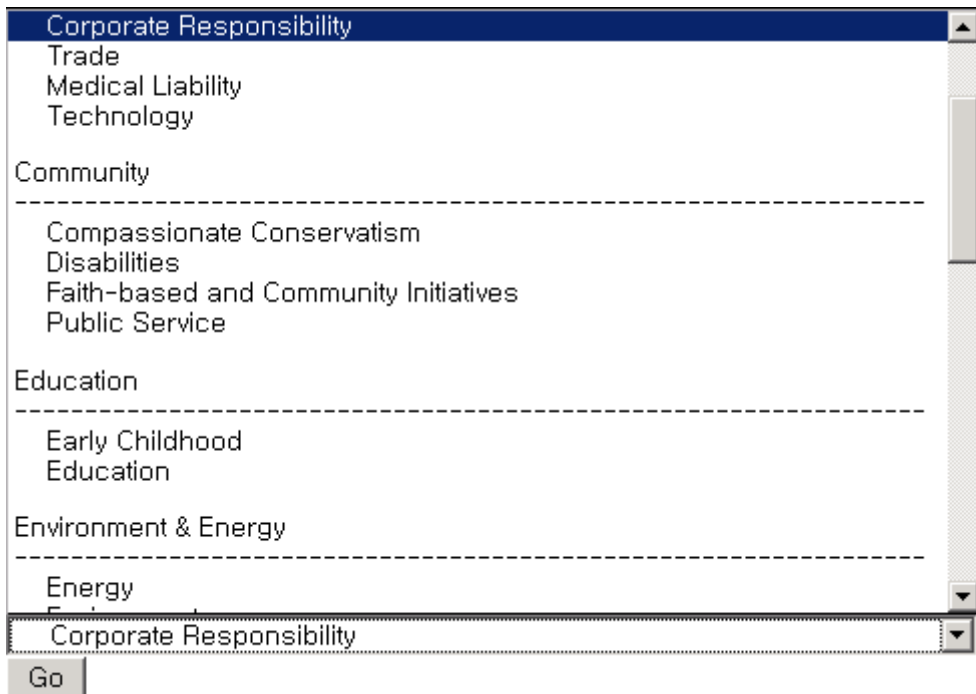
- 위의 전자정부 사이트에서는 목록 상자를 생성하는 <select> 요소의 값이 변하는 이벤트 onchange에,

- 사이트를 바로 이동하는 JavaScript 함수를 연결해놓아, 선택하자마자 사이트가 바뀌어 버리는 문제가 발생함.
- 위와 같이 제작할 경우 시각 및 지체 장애인 등 마우스를 이용하기 힘든 장애인뿐만 아니라,
  - 비장애인들의 경우에도 마우스에 능숙하지 않으면, 실수로 선택을 잘못된 것에 대해 되돌릴 방법이 없는 문제발생
- 화면 낭독 프로그램을 사용하는 시각 장애인들의 경우, 행정부, 입법부, 사법부를 차례대로 읽기도 전에,
  - 첫 번째 항목인 행정부에서 바로 행정부 사이트로 이동해 버리기 때문에, 두 번째 이후 항목에는 전혀 접근할 수 있는 방법이 없음.

## **O (Good)**

- 아래 미국 백악관 사이트에서처럼 항목을 선택한 후, 그것을 최종적으로 실행하는 [가기], [확인], [Submit], 또는 [Go] 등의 버튼을 별도로 두는 것이 바람직함.

<그림 113> 이동 버튼이 따로 있는 바른 목록 상자



※ 출처: 미국 백악관 (<http://www.whitehouse.gov/news/>), 2004. 11. 01

## X (Bad)

<그림 114> 키보드(탭)를 이용한 이동이 불가능한 경우

* 연락처	전화번호	<input type="text"/> - <input type="text"/> - <input type="text"/>	<input checked="" type="radio"/> 집 <input type="radio"/> 직장
	휴대폰	<input type="text"/> - <input type="text"/> - <input type="text"/>	<input type="radio"/> 이동통신사 <input type="radio"/> SK
전화번호나 휴대폰번호 2가지 중 하나는 반드시 입력하세요			
* 거주지 주소	국내	<input type="text"/>	국외

※ 출처: Daum 회원 가입 (<https://register.daum.net/DaumRegister/Register2.jsp>), 2004. 10. 20

```
<input type="text" name="curaddress" value=""
maxlength="30" style="width:316px" class="inputbox">
```

```
onfocus="blur(this)">
```

- 국내에서 매우 자주 발견되는 기법 중 하나가 양식에 마우스 초점(focus)이 이동했을 때 그것을 보이지 않게 만드는 `onfocus="blur(this)"`와 같은 방법임.
- 위의 사례에서는 거주지 주소 입력란에 초점이 이동하지 못하도록 설정해놓아,
- 해당 양식에 키보드 접근이 불가능하게 되며, 장애가 없는 일반인들에게도 현재의 초점을 보이지 않도록 해 사용에 불편을 줌.
- 따라서, `onfocus="blur(this)"`와 같은 기법은 매우 잘못된 관행으로 아주 극단적인 디자인 요구가 있지 않는 한은 절대 쓰지 않는 것이 바람직함.

## O (Good)

- 아래와 같이 `onfocus="blur(this)"`와 같은 이벤트 처리 함수를 없애면 키보드로 접근이 가능해짐.

```
<input type="text" name="curaddress" value=""
maxlength="30" style="width:316px"
class="inputbox">
```

## XIV. 스크립트, 애플릿

지침 13. 스크립트, 애플릿 또는 플러그 인(plug-in) 등과 같은 프로그래밍 요소들은 현재의 보조기술의 수준에서 이들 프로그래밍 요소들의 내용을 사용자에게 전달해줄 수 있을 경우에만 사용하여야 한다.

① 콘텐츠를 디스플레이하기 위해 혹은 인터페이스 요소를 만들기 위해 스크립트 언어를 이용할 경우에는 스크립트에 의해 제공되는 중요한 정보는 최신의 보조기술을 이용해 읽을 수 있어야 한다. 만일 이것이 불가능할 경우에는 스크립트의 동작을 정지시켜도 웹 콘텐츠의 내용을 읽을 수 있어야 한다.

② 애플릿, 플러그인(plug-in) 혹은 다른 응용 프로그램을 이용하여 웹 콘텐츠를 구성하였을 때에는 이들 프로그램 요소에 의해 제공되는 중요한 정보는 최신의 보조기술을 이용해 읽을 수 있어야 한다. 만일 이것이 불가능할 경우에는 이들 프로그램 요소의 동작을 정지시켜도 웹 콘텐츠의 내용을 읽을 수 있어야 한다.

③ 스크립트, 애플릿과 플러그인은 키보드 또는 대체 키보드 인터페이스를 이용하여 사용할 수 있어야 한다.

### 1. 스크립트, 애플릿 삽입

#### (1) 대체적인 접근 방법 제공

☐ 앞서 페이지에서 언급되었듯이 웹 브라우저가 기본적으로

지원하지 않는 객체를 삽입하려면,

- <object> 요소를 사용하며, <object> 요소에 삽입된 내용을 브라우저가 제대로 표시하지 못할 경우에 대비하여 대안적인 내용을 반드시 삽입해 주어야 함
- 흔히 사용되는 스크립트인 ECMA-262 Script (보통 자바스크립트라고 불리우는 것)의 경우도 꼭 필요한 경우에만 사용하는 것이 바람직하며,
  - 스크립트가 작동되지 않더라도 중요한 내용에 접근하는데 문제가 없도록 웹 콘텐츠를 제작해야 함.

## **X (Bad)**

```
[2005년
경제운용 주요 정책 과제...
```

※ 출처: 함께 풀어가는 종합 투자 계획 (<http://jumpkorea.mofe.go.kr/>), 2005. 01. 02

- 위의 예에서 href 속성값으로 JavaScript 함수인 viewArticle을 사용함으로써 JavaScript가 동작하지 않는 브라우저의 접근성을 떨어뜨리고 있는 문제가 발생함
  - viewArticle 함수의 경우 단순히 특정한 페이지를 호출하는 함수이므로 아래와 같이 href의 속성값을 서버측 주소



를 호출하도록 바꾸어주면 접근성이 높아질 것임.

### △ (Bad)

```
<a href="#"
onclick="viewArticle('43','news')">[2005년 경제운용
주요 정책 과제...
```

### O (Good)

```
<a
href="http://...../forum/view.php?bbs_id=forum&
article_id=21">[2005년 경제운용 주요 정책 과제...
```

□ <noscript>의 사용 : 자바스크립트가 작동되지 않을 경우를 대비해 <script> 안쪽에 <!-- --> 와 같이 주석으로 처리하고,

- 추가로 <noscript> 요소 안쪽에 대안적인 페이지를 넣어 줌으로써 더 확실한 접근성을 보장할 수 있음.

### O (Good)

```
<script type="text/javascript">
<!--
자바스크립트 내용
-->
</script>
```

```
<noscript>
 <!-- 대안적인 내용-->

 선택1
 선택2

</noscript>
```

## (2) 흔히 발견되는 자바스크립트 오류

- 자바스크립트를 HTML 중간에 삽입할 때에 <script> 요소를 사용하는데, 이 때에 반드시 스크립트의 유형(type)을 반드시 지정해주어야 함.

### **X (Bad)**

```
<script language="JavaScript" src="example.js" />
```

- 스크립트의 언어(language) 속성은 HTML 4.0 이후 표준에서 낡은(deprecated) 속성으로 바뀌었으므로,
  - language 대신에 type을 반드시 지정해야 하며, 굳이 하위 호환성을 생각한다면 language와 type을 모두 지정할 수 있음.

- 외부 자바스크립트 파일(.js)을 삽입할 때 주의할 점은 외부 파일 문서의 인코딩 방식(euc-kr, utf-8 등)을 명확히 지정해야함.

- 인코딩 방식을 지정함으로써 잘못 디코딩되어 문자가 깨지거나 내용 해석이 안 될 가능성을 줄일 수 있음.

### △ (Good)

```
<script type="text/JavaScript"
language="JavaScript" src="example.js" />
```

### O (Good)

```
<script type="text/JavaScript" charset="utf-8"
src="example.js" />
```

- DOM(Document Object Model) 객체 호출

- JavaScript를 사용하는 큰 이유 가운데 하나는 문서의 구조에 동적으로 접근하여 그것을 조작할 수 있다는 장점으로 많이 활용하고 있음
- 이 때에 문서 구조에 접근하기 위해서는 W3C에서 문서 구조를 모형화한 DOM 표준에 따라 원하는 부분에 접근해야 할 것임.

- 가장 흔히 발견되는 오류는 **form** 객체에 대한 잘못된 접근/호출이며,
- 객체를 DOM 표준에 따라 호출하지 않으면 많은 웹 표시장치(Mozilla, Opera, Konqueror, 수많은 화면 낭독 프로그램, 보이스 브라우저 등)에서 자바스크립트 함수의 원래 기능을 수행하지 못하게 됨.

## **X (Bad)**

```
function moveMailItem(opt) {
 mailitem.submit();
 ...중략...
}

function deleteMailItem() {
 document.mailitem.action = "...";
 ...중략...
}
...중략...
<form name="mailitem" method="post">
```

※ 출처: 코리아닷컴 메일 본문 읽기 화면 (<http://mail.korea.com>), 2004. 12. 26

- 위의 예에서 **mailitem**이라는 이름을 갖는 객체를 자바스크립트 함수에서 호출할 때에 모두 잘못된 방법을 사용하고 있어서 작동하지 않은 브라우저가 생김.

- 아래와 같이 DOM 표준에 따라 호출하도록 바꾸어야 함.

## O (Good)

```
function moveMailItem(opt) {
 document.form.mailitem.submit();
 ...중략...
}

function deleteMailItem() {
 document.form.mailitem.action = "...";
 ...중략...
}
...중략...
<form name="mailitem" method="post">
```

## 2. 플래시(Flash)

- 플래시는 사실상 거의 모든 플랫폼과 거의 모든 브라우저에서 작동되므로, 플래시를 무조건 사용하지 않는 것이 접근성을 높이는 것은 아님.
- 또한 플래시는 다음과 같은 특징으로 인해 접근 가능한 콘

텐츠가 될 수 있는 충분한 잠재력을 가지고 있음 (출처:  
<http://www.webaim.org/techniques/flash/>, 2005. 01. 02)

- 다양한 표현 방법 : 플래시는 콘텐츠를 애니메이션, 텍스트, 음성 등 다양한 방법으로 표시할 수 있음.
  - 크기 확장성(scalability) : 플래시 객체는 래스터(raster) 그래픽이 아닌 벡터(vector) 방식으로 되어 있어, 저시력자가 크게 확대해볼 수 있음.
  - 키보드 접근성 : 플래시는 HTML보다 더 정교하게 키보드 조작이 가능하도록 설계가 가능함.
  - 상호 작용성(engaging) : 플래시는 사용자에게 애니메이션, 그래픽, 사운드 등을 활용하여 정적인 HTML 콘텐츠보다 더 쉽게 상호작용하고, 개념을 전달할 수도 있음.
  - 자체 음성 기능: 플래시 무비에 자체적으로 음성을 내장시킬 수 있기 때문에 화면낭독 프로그램을 사용하지 않더라도 자체적으로 화면낭독 프로그램 역할을 대체하도록 설계할 수도 있음.
- 그러나, 아직까지 현존 보조 기술의 수준에서 플래시에 대한 접근성은 매우 낮은 편이며,

- 플래시의 내장된 접근성 기능들도 Microsoft Active Accessibility(MSAA)라는 기술을 통해 화면 낭독 프로그램이 제한적으로 판독할 수 있음.
- - 즉, Windows와 Internet Explorer 플랫폼이 아닌 환경에서는 플래시의 내장된 접근성 기능들의 혜택을 볼 수 없는 어려움이 있음.
- 그러므로, 플래시로 콘텐츠를 제작할 경우에는 접근성을 높이기 위해 일반 HTML 콘텐츠보다 매우 세심한 주의가 필요함.
- 플래시 콘텐츠 접근성 제작 방법 (Macromedia사의 접근성 담당자인 Bob Regan)
  - 오디오 부분에는 캡션을 붙여라.
  - 장치 독립적으로 디자인하라(주로 키보드로 조작 가능하게 하라는 뜻).
  - 화면 낭독 프로그램 등을 사용할 때 논리적으로 읽는 순서가 나오도록 하고, 키보드를 이용했을 때에도 Tab 순서가 바르게 되도록 하라.
  - 구조를 견고하게 하여 구조를 쉽게 파악할 수 있게 하고,

현재 위치를 사용자가 쉽게 알 수 있게 하라.

- 텍스트가 아닌 요소에는 대체 텍스트를 붙여라.
- 색깔을 주의해서 사용하라.

## X (Bad)

<그림 115> 접근성이 떨어지는 플래시 메뉴



※ 출처: 보건복지부(<http://www.mohw.go.kr>) 2004. 11. 30

- 위 사례에서는 플래시로 메뉴를 구성하고 있으며 키보드를 통한 메뉴 접근이 전혀 불가능하며, 또한 각 메뉴의 설명(대체 텍스트)도 전혀 없어 접근성이 매우 떨어짐
- 일반적으로 마우스를 사용하는 사람들에게도 마우스의 조그



마한 움직임에 따라 메뉴가 급격하게 변하므로 사용성이 떨어짐.

- 꼭 필요에 의해 어쩔 수 없는 경우가 아니면 주 메뉴를 플래시로 제작하는 것은 피하는 것이 바람직함.

## O (Good)

<그림 116> 접근성을 고려한 플래시 메뉴



※ 출처: 유타 대학교 (<http://www.usu.edu/index.cfm?flash=on>), 2005. 01. 02

□ 위의 예에서는 주 메뉴를 플래시로 구성했지만,

- 인터넷 익스플로러 환경에서는 키보드를 통한 메뉴 탐색이 가능하게 제작되었으며,
- 메뉴 항목(위의 그림에서는 04. Community)을 선택했을 때 시각적으로 노란색 테두리가 선명하게 생겨, 현재의 위치를 알 수 있으며,
- 이와 더불어 메뉴 접근이 안 되는 경우에 대비하여 페이

지의 맨 처음 부분에 플래시를 없앨 수 있는 옵션(Turn Flash Off)을 제공하고 있음.

## O (Good)

□ 아래 사례에서는 메인 메뉴가 플래시로 제작되었지만 동시에 텍스트 버전으로도 동일한 메뉴를 선택할 수 있게 함으로써

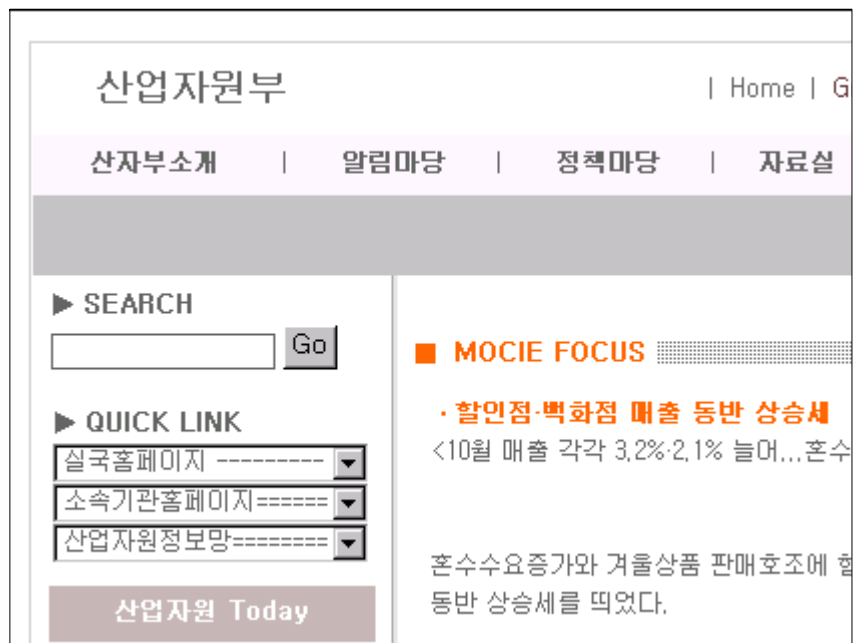
○ 플래시 사용이 어려운 사용자들의 접근성을 보완해주고 있음.

<그림 117> 플래시와 텍스트 메뉴를 함께 제공한 예 (플래시 버전)



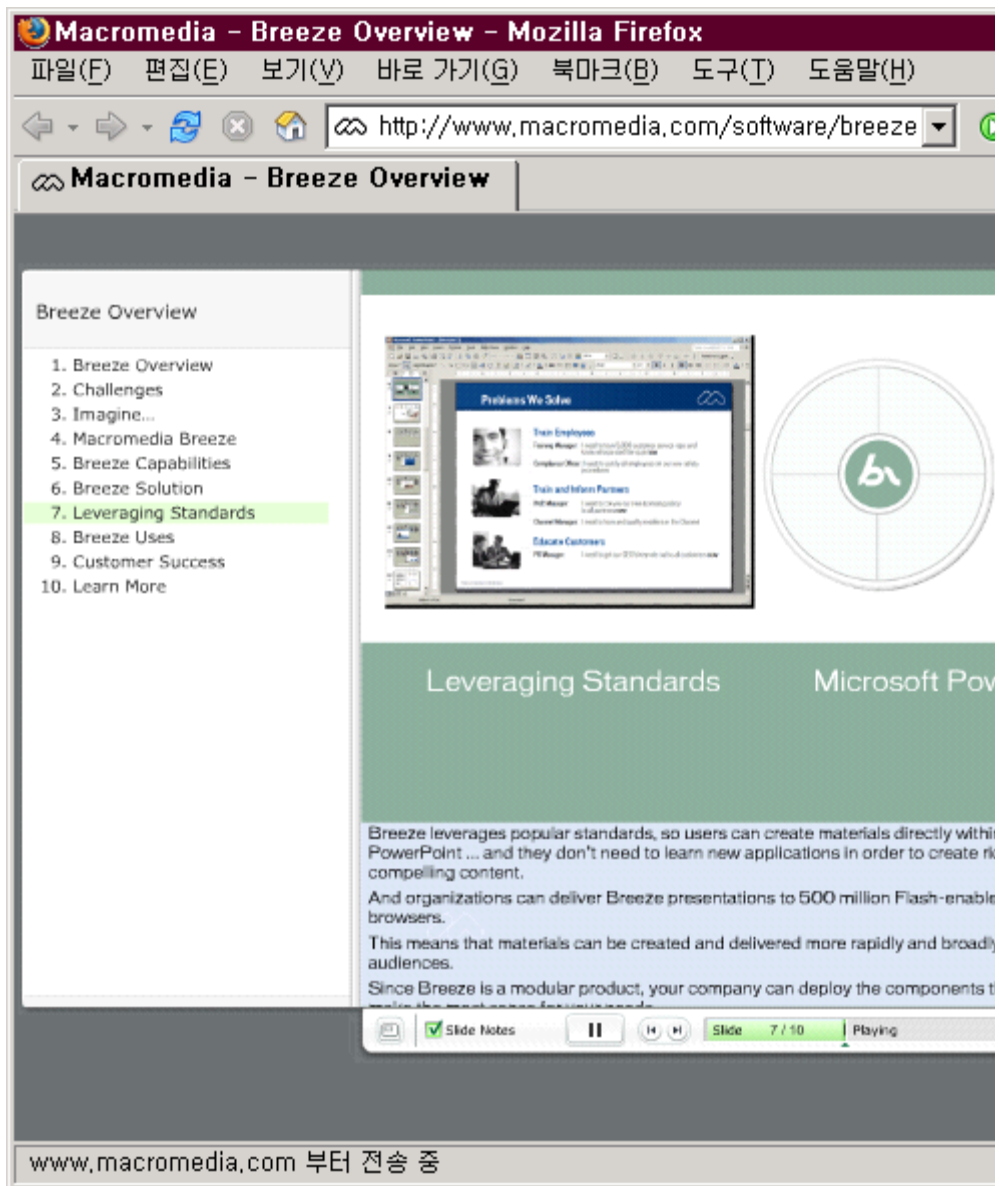
※ 출처: 산업자원부 (<http://www.mocie.go.kr>), 2004. 11. 22

<그림 118> 플래시와 텍스트 메뉴를 함께 제공한 예 (텍스트 버전)



## O (Good)

<그림 119> 캡션을 삽입한 플래시 무비



※ 출처: Macromedia - Breeze Overview

(<http://www.macromedia.com/software/breeze/overview/>), 2005. 01. 02

□ 위의 그림은 플래시 무비에 삽입된 음성과 동일한 캡션을 삽입하여 청각 장애인의 접근성을 높인 예임.

□ 구체적으로 플래시의 접근성을 높이는 제작 방법은 이 문서에서 다루기에는 너무 광범위하므로 아래의 웹 문서를 참조하기 바람

○ Creating Accessible Macromedia Flash Content

(<http://www.webaim.org/techniques/flash/>), 2005. 01. 02

### 3. PDF(Portable Document Format)

□ Adobe사에서 만들어진 PDF 파일은 플랫폼과 장치 독립적으로 문서의 원형을 보존할 수 있는 특징 때문에 웹에서의 사용 빈도가 점점 높아지고 있음

○ Acrobat Reader 6.0에서는 비록 내장 화면낭독 프로그램을 탑재하였지만 여전히 PDF 파일을 화면낭독 프로그램이 접근할 수 있도록 하려면 상당히 주의가 필요함

□ 가장 손쉬운 방법은 PDF 문서와 함께 접근 가능한 HTML 문서를 제공하는 것인데, 이것이 불가능할 경우, 다음과 같은 사항을 준수해야 함

#### (1) 실제 텍스트로 문서를 구성해야 함

- 스캐너로 스캔하여 받은 이미지는 실제 텍스트 데이터가 아니므로 당연히 화면 낭독 프로그램이 읽지 못하며,
- 스캔받은 데이터는 OCR(Optical Character Recognition) 프로그램을 이용하여 실제 텍스트 데이터로 바꾸어야만 화면 낭독 프로그램이 읽을 수 있음

## (2) 적절한 태그를 붙여야 함

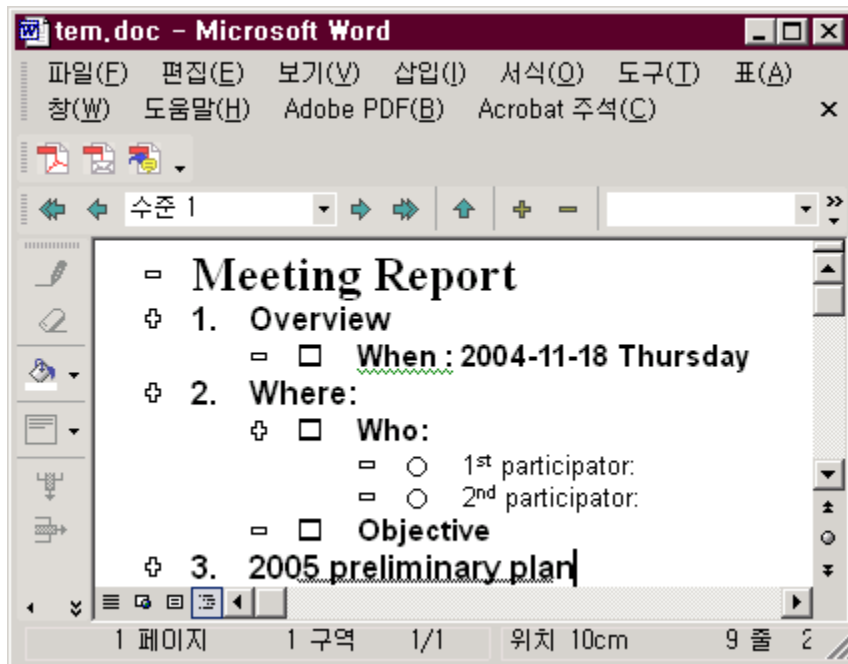
- 현실적으로 PDF 문서에 태그를 수동으로 붙이는 것은 매우 시간과 노력이 많이 들어가는 어려운 작업임으로,
  - PDF 문서 형태가 가장 좋은 문서 형식인가를 생각해 보아야 할 것임
- Adobe사에서 제공하는 Make Accessible Plug-in을 설치한 후 원하는 문서에서 고급(A) > 액세서빌리티(A) > Make Accessible 메뉴를 선택하면 됨.
  - Make Accessible Plug-in 다운로드 사이트:  
<http://www.adobe.com/support/downloads/detail.jsp?hexID=88de>
- 국내에서 많이 사용되는 워드 프로세서인 “한글”의 경우는 아직까지는 구조적인 문서를 작성해도 HTML, PDF로 변환 시 구조 정보가 거의 보전되지 않고 있는 문제가 있음.

### (3) 적절한 마크업을 붙여야 함

- PDF 문서를 생성하기 전에 원 문서를 구조적으로 작성하는 것이 필요함.
- 예를 들어, 마이크로소프트 워드 문서를 작성할 때에는 제목 1, 제목 2와 같은 실제 제목 스타일을 사용해서 문서를 만들어야 하며,
- 단지 글자 크기를 크게 하는 시각적 효과만을 낸다고 해서 구조적인 마크업이 생성되는 것은 아님.

## O (Good)

<그림 120> 워드에서 구조적으로 문서를 작성한 예



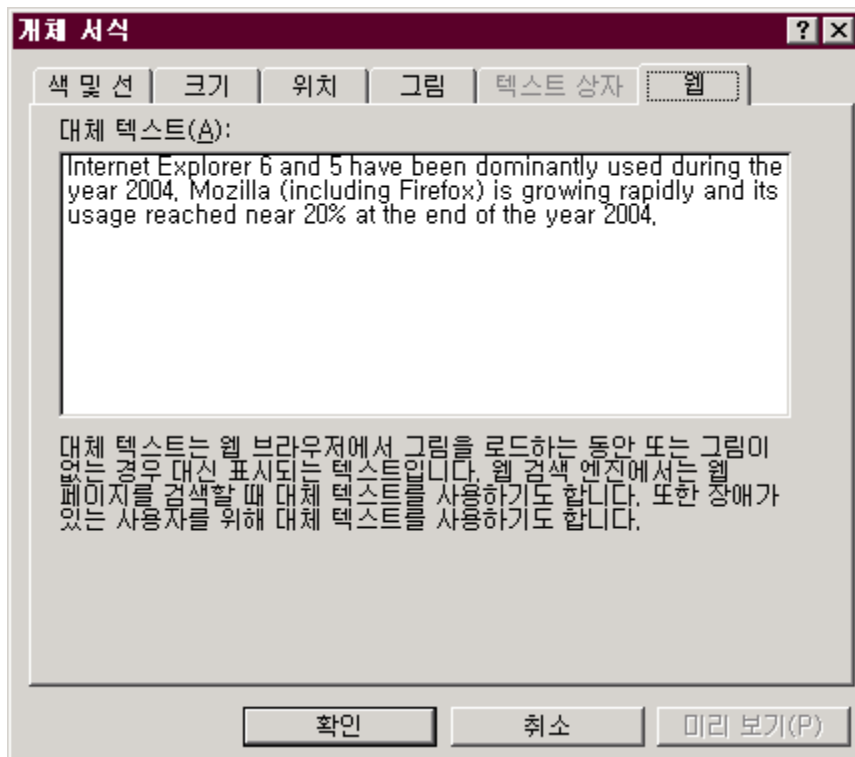
□ 위의 그림은 Microsoft Word에서 스타일과 개요 기능을 이용하여 문서를 구조적으로 작성한 사례로,

- 보기(V) 메뉴에서 개요(O)를 선택하면 문서의 구조를 체계적으로 볼 수 있음.



## O (Good)

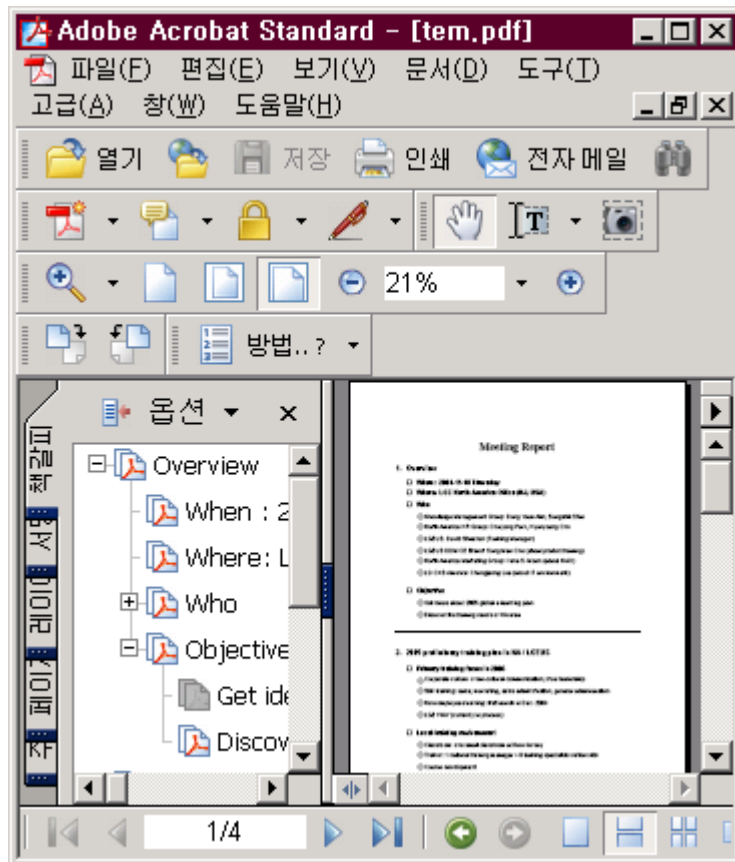
<그림 121> 파워포인트에서 이미지에 대체 텍스트를 삽입하는 대화창



- 위의 그림에서와 같이, 원본 문서를 파워포인트에서 작성할 때에는 삽입된 이미지에 대체 텍스트를 넣어주어야 함.

O(Good)

<그림 122> 구조적인 PDF 문서



- 원본 문서에서 적절한 마크업을 사용해 구조적으로 작성된 문서는 PDF로 변환된 후에도 견고한 구조를 유지할 수 있음.
- 시각 장애인 뿐만 아니라 다양한 유형의 장애인들에게 보편적으로 접근 가능한 PDF 문서를 만드는 자세한 방법에 대해서는 아래의 웹 문서를 참조할 것

- Reading PDF Documents with Adobe Reader 6.0: A Guide for People with Disabilities :  
<http://www.adobe.com/enterprise/accessibility/reader/main.html>

## XV. 별도 페이지

지침 14. 콘텐츠가 13개 지침을 만족하도록 최대한 노력하였으나 해결되지 않는 부분이 남아있다면 텍스트만의 콘텐츠를 제공하는 웹 페이지 (또는 웹 사이트)를 별도로 제공해야 한다.

① 가능한 보조기술 수준이 미흡하여 장애인이 접근 가능한 웹 콘텐츠를 제작할 수 없는 경우에는 텍스트로만 구성된 대체 페이지를 마련하고 기존의 웹 콘텐츠의 첫 페이지에 대체 페이지로 이동하는 링크를 제공하여야 한다.

② 제공하는 대체 페이지는 기존의 웹 콘텐츠가 포함한 정보나 기능을 모두 포함하여야 한다.

③ 제공하는 대체 페이지는 기존의 웹 콘텐츠의 개정 주기에 맞추어 개정되어야 한다.

### 1. 대체 페이지

□ 이상적으로는 단일한 하나의 웹 페이지가 접근 가능하게 제작되어,

○ 장애인과 비장애인, 모든 사용자가 모두 동일한 정보에 접근할 수 있어야 할 것임.

□ 현실적으로 오락성이 매우 강하거나, 시각적인 효과가 매우 중요한 사이트의 경우,

- 현존 기술의 한계나 예산, 시간 부족 등으로 인해 장애인의 접근성을 고려한 보편적인 사이트를 설계하지 못할 경우가 생길 수 있음.
- 또한 저속 인터넷을 사용하여 그래픽 로딩 속도가 느린 사용자나 그래픽 없이 일부러 빠른 속도로 웹의 주요 내용을 보고자 하는 사용자를 위해,
  - 별도로 텍스트 위주로 구성된 사이트를 제공할 수도 있음.
- 이러한 경우에는 별도의 텍스트 위주의 페이지를 만들어 접근성을 보장하는 것이 바람직할 것임.

## O (Good)

<그림 123> 저속 인터넷 사용자를 배려한 Low Graphics version 링크



※ 출처: BBC News Front Page (<http://news.bbc.co.uk/2/hi/default.stm>), 2005. 01. 02

□ 위의 사례에서는 그림과 사진이 많이 들어간 뉴스 페이지에 저속 인터넷 사용자를 배려하여

- 그래픽을 최소화하고 페이지 로딩 속도를 빠르게 한 텍스트 위주의 뉴스 페이지로 갈 수 있는 Low Graphics version이라는 링크를 걸어놓았음.

## O (Good)

<그림 124> 동일한 뉴스를 담고 있는 Low Graphics version



※ 출처: BBC News ([http://news.bbc.co.uk/text\\_only.stm](http://news.bbc.co.uk/text_only.stm)), 2005. 01. 02

## X (Bad)

<그림 125> 전체 플래시로 제작된 사이트



※ 출처 : S-다이어리 (<http://www.s-diary.co.kr>), 2004. 11. 26

- 영화 홍보 사이트와 같이 화려함이 강조되는 사이트는 흔히 전체가 플래시로 제작되기도 함.
- 이 경우, 최소한의 정보성 콘텐츠만이라도 장애인용 대체 페이지를 만들어주어야 할 것임.



## X (Bad)

<그림 126> 전체가 플래시 애니메이션으로 제작된 사이트



※ 출처: 코크플레이닷컴 (<http://www.cokeplay.com>), 2004. 11. 26

- 오락성이 강한 사이트로 보편적 접근성을 고려하지 않고 전체를 플래시 애니메이션으로 제작하였음.
- 이 경우도 최소한의 정보에 대해서는 대체 페이지를 제공해주는 것이 좋을 것임.

## X (Bad)

<그림 127> 별도 장애인 페이지로 가는 링크가 없는 경우



※ 출처: 광진 구청 (<http://www.gwangjin.go.kr>), 2005. 02. 18

- 위의 사례에서는 장애인, 노인용 음성 페이지를 별도로 제작했으나,
  - 해당 페이지로 가는 링크가 일반 페이지 내에 전혀 없음.
- 또, 페이지 제목에 Ctrl + Enter를 누르면 장애인용 페이지로 이동한다고 나와있으나,
  - 이러한 키 조합은 범용적인 것이 아니므로 다른 종류의 브라우저, Mac OS, Linux 사용자, 특수한 키조합을 사용하는 화면 낭독 프로그램 사용자에게는 무용지물이 됨.

- 굳이 키보드 단축키를 제공하려면 HTML 표준에서 제공하는 `accesskey=""`를 사용해야 더 많은 시스템에서 작동을 보장할 수 있음.

## X (Bad)

<그림 128> 전혀 다른 인터페이스를 사용하는 별도의 음성 홈페이지



※ 출처: 광진구청 장애인용 홈페이지 (<http://www.gwangjin.go.kr/friend/>), 2005. 02. 18

- 위의 그림과 같이 별도로 구성된 음성 홈페이지에서도 별도의 인터페이스를 제공하고 있어 접근성을 오히려 떨어뜨리고 있음.
- 일반적이지 않은 특수한 단축키 조합(Ctrl+T, Ctrl+M 등)을 사용하고 있어 기존 화면 낭독 프로그램과 충돌을 일

으킬 수 있음.

- 별도의 음성을 제공함으로써 기존에 이미 화면 낭독 프로그램을 통해 음성을 제공받은 사용자에게 오히려 혼란을 줄 수 있음.
- 음성 홈페이지에 접근하기 위해 인증받지 않은 ActiveX 컨트롤을 설치하는 어려운 절차를 거치게 하고 있음.

□ 플러그 인이나 ActiveX 컨트롤을 사용할 경우는

- 이것이 최대한 보편적인 것이어야 하며, 여러 사용자 환경에서 문제 없이 사용될 수 있어야 함.
- 일반적으로 ActiveX 컨트롤은 특정한 시스템에서만 작동하고, 상호 운용성 측면에서 다른 시스템을 사용하는 사용자를 배제시킬 수 있으므로 사용을 최소화해야 함.
- 플러그 인의 설치 자체가 어려운 사용자들을 위해서 플러그 인이 없을 경우에도 동일한 기능을 할 수 있도록 웹 표준에 맞게 작성된 페이지를 별도로 제공해야 함.

□ 따라서, 아주 예외적인 경우가 아니라면,

- 별도의 장애인용 페이지를 제작하는 것보다,
- 원래의 페이지의 접근성을 높이는 방법이 훨씬 쉽게 구현 가능하고, 실질적으로도 동등한 접근성을 보장해줄 수 있음.

## 2. 대체 페이지의 갱신

- 장애인용 페이지를 별도로 만드는 것은 두 개의 페이지를 관리해야 하는 어려움 때문에 바람직하지 않음.
- 그러나 꼭 두 개의 페이지를 별도로 관리할 경우에는 장애인용 페이지에 원래의 페이지와 동일한 내용을 담고,
  - 그 내용이 동일한 시기에 갱신되어 장애인도 동등한 정보에 차별없이 접근할 수 있도록 운영해야 함.

### X (Bad)

<그림 129> 최신 소식이 올라온 일반 홈페이지

새소식	보훈행사	정보공개	인사정보
<ul style="list-style-type: none"><li>• [2004/11/30] 남구 보훈3단체 고령 보훈가족 위안행사 NEW</li><li>• [2004/11/30] (마산)마산고 전물동문추모비에 오석 안내판 설치 NEW</li><li>• [2004/11/29] 울주군 보훈4단체 고령 보훈가족 위안행사 개최 NEW</li><li>• [2004/11/29] 중구 보훈3단체 고령 보훈가족 위로연 개최 NEW</li><li>• [2004/11/29] 동구 보훈3단체 고령 보훈가족 위로연 개최 NEW</li><li>• [2004/11/29] 새양산병원 국가유공자 및 참전유공자 감면진료 실시 NEW</li><li>• [2004/11/29] 제7회 고현 박상진 의사 추모 학생 글짓기대회 시상식 NEW</li></ul>			

※ 출처: 국가보훈처(<http://www.mpva.go.kr>), 2004. 11. 30

<그림 130> 최신 소식이 올라오지 않은 시각장애인용 페이지

여기는 국가보훈처의 보훈소식을 제공하는 페이지입니다.


전체 게시물 수 /394건

제목	구분	날짜
노 대통령, 독립유공자 초청 오찬	종합	2004-08-26
국가보훈처 나라사랑 홈페이지 전면 개편	종합	2004-08-20
참전유공자등 지원 관련 보훈관계 법령 개정 추진	종합	2004-08-17
독립유공자 예우개선 대책	종합	2004-08-10
독립유공자 후손 초청 한민족공동체의식 고양	종합	2004-08-10

※ 출처: 국가보훈처([http://www.mpva.go.kr/blind/notice/announce\\_list.asp](http://www.mpva.go.kr/blind/notice/announce_list.asp)), 2004. 11. 30

□ 위의 사례에서는 시각 장애인용 페이지를 별도로 두었으나

- 일반 홈페이지와 달리 시각 장애인용 페이지에는 제 때에 내용이 개정되지 않아 11월 말인데도 8월까지의 소식만 올라와 있음.

<p>TTA Standard</p>	<p>정보통신단체표준                      제정일 : 2004년 12월 23일 TTAS.OT-10.0003</p> <p>한국형 웹 콘텐츠 접근성 지침 1.0 (Korean Web Contents Accessibility Guideline 1.0)</p> <p> <b>한국정보통신기술협회</b> Telecommunications Technology Association</p>
---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# 1. 서문 (Preface)

## 1. 표준의 목적

이 문서는 장애를 가진 사람들이 접근할 수 있도록 웹 콘텐츠를 제작하는 방법에 관하여 기술하고 있다. 특별히 이 문서 내에 포함된 지침들은 웹 콘텐츠 저자, 웹 사이트 설계자 및 웹 콘텐츠 개발자들이 웹 콘텐츠를 접근성(Accessibility)을 준수하여 쉽게 만들 수 있도록 도움을 주기 위하여 기획되었다.

## 2. 참조권고 및 표준

### 2.1 국제표준(권고)

- o W3C Recommendation, "Web Contents Accessibility Guideline 1.0", May. 1999.

### 2.2 국내표준 : 없음

### 2.3 기타

- o Section 508 Amendments subpart B -Technical Standards, "Web-based intranet and Internet information and application", Dec. 2000

## 3. 국제표준(권고)과의 비교

### 3.1 국제표준(권고)과의 관련성

- o W3C WAI WCAG 1.0 (1999. 5)의 중요도 1과 미국 재활법 508조 기준을 중심으로 국내 실정에 맞게 구성함



- o 국내 스크린 리더, 화면확대기 등 보조기술(Assistive Technology)과의 호환성 문제를 고려하여 국내 실정에 맞는 지침으로 구성함

### 3.2 상기 국제표준(권고) 등에 대한 추가사항

3.2.1 선택항목 : 없음

3.2.2 National Matter 항목 : 없음

3.2.3 기타항목 : 없음

### 3.3 참조한 국제표준(권고)과 본 표준의 장 구성 비교표

해당사항 없음

### 4. 지적재산권 관련사항

해당사항 없음

### 5. 적합인증 관련사항

해당사항 없음

### 6. 표준의 이력

판수	제.개정일	개정판 내용
제 1판	2004년 12월 23일	제정

# Preface

## 1. Objectives

This document describes the method of developing web contents which all the people include disabled and elderly people can access. Especially, this document helps web contents developers, designers and contents providers make web contents accessible.

## 2. Normative standards and recommendations

### 2.1 International Standards

- o W3C Recommendation, "Web Contents Accessibility Guideline 1.0", May. 1995

### 2.2 Domestic Standards : None

### 2.3 Others

- o Section 508 Amendments subpart B –Technical Standards, "Web-based intranet and Internet information and application", Dec. 2000

## 3. Comparison to international standards

### 3.1 The relationship to International Standards

- o This document is developed on the basis of the W3C WCAG 1.0 (1999. 5) and Section 508 technical standards, and reflects domestic specific circumstances

- o Especially, this document is developed to consider the compatible issues of the Korean assistive technologies(AT) such as screen readers, screen magnifier

3.2 Additional Requirements to the international standards : None

3.3 Differences with International Standards

4. Intellectual property rights

None

5. Requirements for conformance and certification

None

6. The history of standard

Edition	Issued Date	Contents
The 1 <sup>st</sup> edition	2004.12.23	Established

## 목차 (Contents)

1. 개요
2. 표준의 구성 및 범위
  - 2.1 지침 제정 목적
  - 2.2 문서의 구성
3. 접근성 있는 웹 콘텐츠 설계
  - 지침 1. 인식의 용이성
    - 항목 1.1 텍스트 아닌 콘텐츠(Non-text Contents)의 인식
    - 항목 1.2 영상매체의 인식
    - 항목 1.3 색상에 무관한 인식
  - 지침 2. 운용의 용이성
    - 항목 2.1 이미지 맵 기법 사용 제한
    - 항목 2.2 프레임의 사용 제한
    - 항목 2.3 깜빡거리는 객체 사용 제한
    - 항목 2.4. 키보드로만 운용 가능
    - 항목 2.5 반복 네비게이션 링크(repetitive navigation link)
    - 항목 2.6 반응시간의 조절기능
  - 지침 3. 이해의 용이성
    - 항목 3.1 데이터 테이블 구성
    - 항목 3.2 논리적 구성
    - 항목 3.3 온라인 서식 구성
  - 지침 4. 기술적 진보성
    - 항목 4.1 신기술의 사용
    - 항목 4.2 별도 웹사이트 제공

## 한국형 웹 콘텐츠 접근성 지침 1.0

### Korean Web Contents Accessibility Guideline 1.0

#### 1. 개 요

이 문서는 장애를 가진 사람들이 접근할 수 있도록 웹 콘텐츠를 제작하는 방법에 관하여 기술하고 있다. 특별히 이 문서 내에 포함된 지침들은 웹 콘텐츠 저자, 웹사이트 설계자 및 웹 콘텐츠 개발자들이 접근성(Accessibility)을 준수한 웹 콘텐츠를 쉽게 만들 수 있도록 도움을 주기 위하여 기획되었다.

이 지침을 제정하는 가장 큰 목적은 웹 콘텐츠에 접근하려는 모든 사람들이 어떤 컴퓨터나 운영체제, 또는 웹 브라우저(web browser)를 사용하든지, 또는 어떠한 환경에 처해 있는지에 구애받지 않고 접근할 수 있는 웹 콘텐츠를 제작할 수 있도록 도와주는 것이다. 그렇다고 해서 이 지침이 장애인들의 접근권을 위하여 그래픽 이미지나 비디오 자료와 같이 장애인들이 접근하기 어려운 콘텐츠를 사용하지 못하도록 제한하는 것은 아니며, 이들 멀티미디어 콘텐츠들도 보다 광범위한 부류의 사람들이 용이하게 접근할 수 있도록 콘텐츠를 구성하는 방안을 제시하는데 주목적이 있다.

이 지침은 한국 정보통신접근성향상표준화포럼 산하 웹접근성분과위원회가 주축이 되어 제안하는 것으로 국제 표준화 기구인 W3C(World Wide Web Consortium)의 WAI(Web Accessibility Initiative)에서 1999년 5월 제정한 웹 콘텐츠 접근성 가이드라인(WCAG : Web Content Accessibility Guidelines 1.0)과 WCAG 2.0 초안 (2003년 6월 24일판)을 참고하였으며, 부분적으로 미국 재활법 508조를 참고하여 작성한 것이다. 특히 제정시점을 기준으로 국내의 웹 환경 및 관련 보조기술의 실태를 조사하고 그 결과를 토대로 표준(안)을 마련하였다.

## 2. 표준의 구성 및 범위

### 2.1 지침 제정 목적

이 문서는 접근이 가능한 웹 콘텐츠를 만들기 위한 설계 원리를 개괄한다. 만일 이 문서에서 제시하는 원리를 무시한 채로 웹 콘텐츠를 제작한다면 장애를 가진 사람들은 웹 콘텐츠에 접근할 수 없거나 접근하는 데에 어려움이 따르게 된다. 반면에 이 문서에서 제시한 원리에 따라 웹 콘텐츠를 제작한다면 적절한 보조기술을 사용하는 장애인들은 해당 콘텐츠에 충분히 접근할 수 있을 것이다. 웹 문서를 설계함에 있어서 고려해야 하는 접근권의 유형은 다음과 같다.

- 1) 시각을 통하여 정보를 인지할 수 없거나 인지 능력이 떨어지는 시각 장애가 있는 경우
- 2) 청각을 통하여 음향 정보를 인지하지 못하는 청각장애가 있는 경우
- 3) 신경계의 마비, 근골격계의 마비 또는 선천성 기형 등으로 신체의 움직임에 제한이 있는 지체장애가 있는 경우
- 4) 읽기나 문장 이해력이 떨어지는 언어장애가 있는 경우
- 5) 키보드나 마우스를 사용할 수 없는 장애가 있는 경우
- 6) 문서에 적용된 언어에 따라 말하거나 이해하는데 어려움이 있는 장애가 있는 경우 - 예를 들면 외국어로 작성된 콘텐츠를 이해하지 못하는 경우
- 7) 시각, 청각 또는 손을 사용하고 있어서 필요로 하는 일을 할 수 없는 경우 - 예를 들어 운전 중이나 소음이 많은 곳에서 일하는 경우로 장애라기보다는 처한 환경에 따라 제한 받는 경우
- 8) 사용하고 있는 웹 브라우저의 종류나 버전이 다름으로 인하여 어려움이 발생하는 경우 등이다.

콘텐츠 제작자는 모든 사용자들이 충분히 접근할 수 있는 콘텐츠를 설계하거나 제작할 필요가 있다. 이 문서에서는 위에 열거한 경우에도 충분히 그 내용을 인지할 수 있도록 하기 위한 웹 콘텐츠의 설계 방법에 대한 지침을 제시할 것이다. 그러나 이 문서는 기본적인 목적이 HTML 문법을 교육시키거나 HTML을 이용하여 웹 문서를 설계하는데 필요한 기초적인 지식을 제공하려는 것은 아니다.

## 2.2 문서의 구성

이 문서에서는 사람들이 필요한 부분을 빠르게 찾아볼 수 있도록 하기 위해 서로 관계가 있는 항목들을 네 가지 지침으로 분류하였다. 즉, 웹 콘텐츠에 접근하기 위해 갖추어야 할 네 가지 주요 지침은 인식의 용이성, 운용의 용이성, 이해의 용이성, 기술적 진보성의 네 가지이며, 네 가지 지침은 각각 몇 개의 검사항목(또는 항목)으로 구성된다. 본 표준에서 제시하는 검사항목은 총 14개이다. 각 검사항목은 요구사항, 용어정의, 해당 항목을 준수함으로써 얻어지는 혜택 및 적용 예 등으로 구성된다.

## 3. 접근성 있는 웹 콘텐츠 설계

이 문서에서 목표로 하는 것을 개괄하면, 모든 사용자가 그 내용을 인지할 수 있으며, 운용할 수 있고, 이해할 수 있는 웹 콘텐츠를 만들 수 있도록 하는 것이다. 이 경우에 현재까지 개발된 보조기술이나 앞으로 개발될 기술을 사용하는 경우도 염두에 두고 있다. 접근성이 준수된 웹 콘텐츠란 아래의 네 가지 지침을 만족하는 경우에 가능하다. 따라서 이 문서에서는 아래의 네 가지 지침에 포함되는 모든 검사항목을 만족해야 접근성이 있다고 하며, 일부분 또는 전체 검사항목을 만족하지 못하는 경우에는 접근성이 없는 웹 콘텐츠라고 정의한다.

가. **인식의 용이성** : 글로 표현할 수 없는 콘텐츠를 제외하고 장애 유형에 관계없이 모든 사용자가 콘텐츠를 인지할 수 있도록 제공해야 한다.

나. **운용의 용이성** : 콘텐츠에 포함된 모든 구성 요소들은 장애 유형과 관계없이 모든 사용자가 사용할 수 있어야 한다.

다. **이해의 용이성** : 모든 콘텐츠는 가능한 한 그 내용과 사용 방법을 모든 사용자가 이해하기 쉽도록 구성해야 한다.

라. **기술적 진보성** : 현재 개발된 보조기술로는 접근이 어렵거나 불가능한 웹 콘텐츠는 가용한 보조기술을 이용하여 접근할 수 있도록 대체 콘텐츠를 함께 제공해야 한다.

접근성이 있는 웹 콘텐츠는 장애를 가진 사람들만이 아니라 다양한 사람들에게 도움을 줄 수 있다. 예를 들어 엘리베이터는 휠체어를 타고 있는 사람에게만 유용한 것이 아니라 유모차로 이동하는 사람이나 노약자에게도 유용하다. 마찬가지로 접근성이 있는 웹 콘텐츠는 장애를 가지고 있는 사람이나 그렇지 않은 사람 모두에게 도움을 줄 수 있다. 예를 들어, 소음이 심한 지역에서 웹사이트에 접근하거나 웹 문서상의 음성정보를 이용하는 경우에 캡션이 제공된다면 제한적인 환경에도 불구하고 그 내용을 알 수 있을 것이다. 운전중인 경우에는 눈을 다른 데로 돌릴 겨를이 없기 때문에 문서를 읽어 주거나 음성정보를 제공하도록 구성된 콘텐츠가 매우 유용할 것이다. 마찬가지로 영화에 캡션(자막)이 제공된다면 검색 엔진을 활용하여 이 영화로부터 유명한 대사를 쉽게 찾을 수 있는 편리성을 제공하기도 한다.

여기서 고려해야 할 장애의 유형과 장애를 극복하기 위하여 사용될 수 있는 대체 수단에 관한 몇 가지 방법을 살펴보자.

가. 들을 수 없는 사람은 일반적으로 소리를 통해 제공되는 정보를 눈으로 보고 싶어한다.

나. 볼 수 없는 사람은 시각적으로 제공되는 정보를 텍스트 또는 점자 정보로의 변환을 통해 듣거나 읽을 수 있기를 원한다.



다. 민첩하거나 용이하게 움직일 수 없는 사람은 가능한 한 적게 움직여서 원하는 일을 하거나 필요한 만큼 충분한 시간동안 작업 할 수 있기를 원한다.

라. 글씨를 잘 읽지 못하는 사람은 그 내용을 음성으로 읽어주기를 원한다.

위에 예로든 네 가지 경우는 모든 장애 유형이나 경우를 망라한 것이 아니며, 자주 접하게 되는 경우에 이에 대한 대책을 예로 든 것이다. 따라서 보다 복잡한 경우에는 또 다른 접근 방법을 마련해야 한다.

만약 어떤 웹 콘텐츠가 이 문서에서 제시하는 항목에 따라 제작된다면 사용자는 보조기술을 사용하여 웹 콘텐츠에 쉽게 접근할 수 있다. 예를 들어 스크린 리더(screen reader)와 적당한 웹 브라우저(web browser)를 사용하면 웹 페이지의 내용을 음성으로 읽어줄 수 있어서 음성으로 그 내용을 읽어주기 원하는 모든 사람들의 접근권을 보장할 수 있다. 스크린 리더와 같이 장애유형별로 웹 콘텐츠에 접근할 수 있도록 도와주는 많은 보조기술들이 개발되어 있으므로 장애 유형과 필요한 보조기술을 활용하여 웹 콘텐츠의 접근성을 향상시킬 수 있다. 이와 관련한 사항은 관련 장애인 단체에서 제공하는 정보나 웹사이트를 참고하라.

**지침 1. 인식의 용이성** - 웹사이트에서 서비스하고 있는 모든 콘텐츠는 누구나 쉽게 인식할 수 있도록 설계되어야 한다.

**항목 1.1 (텍스트 아닌 콘텐츠(non-text contents)의 인식)** 텍스트 아닌 콘텐츠 중에서 글로 표현될 수 있는 모든 콘텐츠는 해당 콘텐츠가 가지는 의미나 기능을 동일하게 갖추고 있는 텍스트로도 표시되어야 한다.

## 가. 용어 정리

- (1) 텍스트 아닌 콘텐츠(non-text contents)란 그림, 이미지 등으로 제작된 텍스트, 애니메이션, 아스키(ASCII) 그림문자, 목록표시 이미지, 그래픽 버튼, 음성음향자료 또는 비디오 자료 등의 오디오와 영상과 같이 표준 문자(부호) 체계가 아닌, 시각적 또는 청각적 정보가 포함된 콘텐츠를 의미한다. 한글 부호의 경우에는 유니 코드, 조합형 또는 완성형 부호체계를 사용하여 작성된 텍스트 외의 모든 경우를 의미한다.
- (2) 빈 문자(blank text)란 아무런 정보도 가지고 있지 않은 문자열을 의미한다. HTML 등의 문법에서 빈 문자는 “”을 가리킨다. 즉, 빈 문자를 스크린 리더에서 읽으면 아무런 소리도 나지 않는다.

## 나. 요구 조건

- (1) 텍스트 아닌 콘텐츠 중에서 글로 표현될 수 있는 콘텐츠는 모두 해당 콘텐츠와 정확하게 일치하는 텍스트 콘텐츠(text contents)와 함께 제공해야 한다.
- (2) 만약, 텍스트 아닌 콘텐츠 중에서 글로 표현될 수 없는 경우는 해당 콘텐츠에 대한 설명을 텍스트로 반드시 제공해야 한다. 이 때, 제공되는 텍스트 콘텐츠는 설명하고자 하는 원래의 콘텐츠의 내용이나 형태 등의 상세한 정보를 제공해야 한다.
- (3) 텍스트 아닌 콘텐츠를 대체하거나 설명하기 위해 제공되는 텍스트는 콘텐츠 제작자가 원래의 콘텐츠를 사용해 표현하고자 했던 의미나 기능도 포함하여 설명해야 한다.

- (4) 텍스트 아닌 콘텐츠를 대체하거나 설명하기 위해 제공되는 텍스트를 읽어줌으로 인하여 의미의 혼동을 줄 경우에는 빈 문자(blank text)를 대체 텍스트로 사용해야 한다.

#### 다. 적용시 장점

- (1) 시각장애 또는 인지장애 등으로 인해 시각으로 정보를 습득하는 데에 어려움을 겪는 사용자들이 스크린 리더(screen reader)와 같은 보조기술을 사용하여 텍스트 콘텐츠를 음성을 통해 읽음으로써 접근권을 보장받을 수 있게 된다.
- (2) 청각장애나 다른 이유로 음향 정보를 알아들을 수 없는 사용자들을 위해 그 내용을 텍스트로 표시하거나 수화로 해당 내용을 번역함으로써 접근이 가능하게 된다.
- (3) 시각장애인은 물론 시각장애와 청각장애를 함께 갖고 있는 사용자 역시 해당 콘텐츠를 점자로 변환해 읽음으로써 접근권을 보장받을 수 있게 된다.
- (4) 몇 개의 자료를 순서대로 나열하는 목록 등 해당 이미지가 단순히 홈페이지를 아름답게 꾸미기 위해 사용된 경우에는 이미지에 대한 대체 텍스트(alt-text)로 빈 문자를 사용하면 의미의 불필요한 혼동을 줄일 수 있다.

#### 라. 적용 예

- (1) 버튼으로 사용된 이미지 : 간단한 기능 설명부가

슬라이드 쇼에서 다음 슬라이드로 넘어가는 역할을 하도록 만든 오른

쪽 화살표 모양의 이미지 링크(image link)의 경우에 이 버튼 이미지와 함께 "다음 슬라이드"라는 대체 텍스트를 함께 제공한다면 스크린 리더는 이 이미지 버튼을 "링크 - 다음 슬라이드"라고 읽어줄 수 있게 된다.

#### (2) 데이터 차트 : 간단한 레이블과 충분한 설명을 함께 제공

예를 들어 어떤 상품의 6월, 7월 및 8월의 판매실적을 비교해 보여주는 차트에 "그림 1 : 6월, 7월 및 8월의 판매 현황"이라는 간단한 레이블(label)을 붙이고, 차트에 대한 보다 긴 설명문(longdesc)을 붙인다면 해당 차트의 내용과 해당 차트가 표시하고 있는 정보를 사용자에게 전달할 수 있다.

#### (3) 애니메이션 : 간단한 레이블과 충분한 설명을 함께 제공

예를 들어 매듭을 짓는 방법을 보여주는 애니메이션의 경우에 "매듭을 만드는 형태를 보여주는 애니메이션"이라는 간단한 레이블과 함께 매듭짓기 위하여 손놀림하는 동작을 자세하게 설명하는 설명문을 붙인다면 매듭을 만드는 전 과정을 이용자가 자세하게 알 수 있다.

#### (4) 연설 녹음자료 : 간단한 레이블과 긴 설명을 함께 제공

웹 페이지에 음성 데이터가 포함되어 있는 경우에 이 데이터와 함께 간단한 제목을 레이블로 구성하고 음성 데이터의 내용에 대한 텍스트를 함께 제공한다면 청각장애인의 경우에도 모든 정보를 얻을 수 있다.

#### (5) 음악 자료 : 간단한 레이블 제공

예를 들어 베토벤의 교향곡 제 5번의 연주실황을 제공하는 음악 파일을 웹 페이지에서 제공하고 있는 경우에 이 음악 파일과 함께 "베토벤 교

향곡 제 5번, 빈 필하모닉 오케스트라의 연주"라는 레이블을 함께 제공한다면 청각장애인의 경우에 음악 파일이 무슨 내용을 담고 있는지를 쉽게 알 수 있다.

## 항목 1.2 (영상매체의 인식) 시간에 따라 변화하는 영상매체는 해당 콘텐츠와 동기 되는 대체 매체를 제공해야 한다.

### 가. 용어정리

- (1) 시간에 따라 변화하는 영상매체(time-dependent presentation)란 영화와 같이 소리와 영상정보가 동시에 제공되는 경우 혹은 콘텐츠 재생 과정의 특정 시점에서 사용자와의 상호작용 또는 대화가 필요한 매체를 말한다.
- (2) 대체 매체(equivalent media)란 영상매체의 음향정보를 캡션과 같이 시각적으로 보여주거나 영상의 기본적인 내용을 화면해설 형태로 제공하는 매체를 의미한다.
- (3) 화면해설(audio descriptions)이란 영상의 광경, 동작, 비언어적 표현, 그래픽 등의 상태나 변화를 음향정보로 나타내거나 텍스트로 제공하여 필요한 설명을 제공하는 것을 의미한다. 화면해설은 해당 이벤트와 동기 되어 진행되어야 한다.
- (4) 캡션(Captions)이란 영상매체에 포함된 말, 음향 및 주변소리 등을 텍스트로 표현한 매체를 의미한다. 따라서 캡션은 영상매체의 진행에 따라 해당 이벤트와 동기 되어야 한다.

## 나. 요구조건

- (1) 음향 정보로부터 사용자가 충분한 정보를 얻을 수 없는 모든 영상매체는 캡션을 함께 제공해야 한다.
- (2) 캡션은 영상매체나 음향매체와 동기 되어야 한다.
- (3) 모든 생방송은 캡션을 실시간으로 제공해야 한다. 단 음성이 없는 음악 방송의 경우에는 예외로 한다.
- (4) 콘텐츠는 사용자가 캡션과 영상정보를 동시에 접해야 그 내용을 이해할 수 있도록 구성되어서는 안 된다. 즉, 캡션과 영상정보는 서로 보완관계가 되지 않아야 한다.

## 다. 적용시 장점

- (1) 청각장애인은 음향매체와 함께 제공되는 캡션을 통해 음성이나 음향 정보에 접근이 가능하게 된다. 또한, 캡션을 활용하면 해당 콘텐츠에 대한 인덱스를 작성하거나 내용을 검색할 때에도 유용하게 사용될 수 있다.
- (2) 장애인이 아닌 경우에도 영상매체와 함께 동기 되는 대체 매체가 제공되는 경우에 보다 편리하게 콘텐츠를 활용할 수 있다. 예를 들어 캡션은 소란한 환경이나 오디오 재생기능이 갖추어져 있지 않은 경우 유용하며, 외국어 습득과 같이 언어능력이나 읽기 능력을 높이는 데 활용될 수 있다.
- (3) 화면해설의 경우에는 화면을 보지 않더라도 중요한 정보를 음성으로 알 수 있게 해주므로 시각을 통해 사물을 인식하기 어려운 장애인의

경우에는 화면해설을 통해 화상정보를 인지할 수 있다. 그러나 화면해설은 필수사항으로 분류하지 않는다.

## 라. 적용 예

### (1) 동영상 : 화면해설과 캡션 활용

예를 들어, 한 아이가 강아지를 자신의 침대로 오도록 과자를 늘어놓는 강아지에게 웅얼거리며 무언가를 말하고 있는 장면을 가정한다면, 화면을 볼 수 없는 사람은 이 장면을 이해할 수 없다. 따라서 이 경우에 "이 아이가 계단으로부터 자기 침실에 이르는 계단마다 과자를 하나씩 늘어놓았다"라는 화면해설과 "아이가 웅얼거린다"라는 캡션을 제공한다면 청각장애인이거나 시각장애인 모두 그 내용을 정확히 이해할 수 있다.

### (2) 뉴스 동영상 : 영상과 동기되는 캡션 제공

예를 들어, 홍수가 난 대도시의 광경을 기자가 설명하고 있다고 가정할 경우에 기자가 하는 말을 캡션으로 표시하면 청각장애인이 기자의 이야기를 인식할 수 있게 된다.

### (3) 무언극 : 장면별로 장면소개 텍스트 제공

예를 들어, 사다리를 올라가는 무언극(팬터마임 : pantomime)의 한 장면을 애니메이션으로 구성하였을 경우에 “사다리를 올라가는 장면”이라는 대체 텍스트를 애니메이션에 포함시키면 이로써 충분한 정보를 사용자에게 제공할 수 있다.

항목 1.3 (색상에 무관한 인식) 콘텐츠가 제공하는 모든 정보는 색상을 배제하더라도 인지할 수 있도록 구성되어야 한다.

가. 용어정리

- (1) 고 대비(high contrast) 모드란 흑백과 같이 대비차가 매우 크도록 조정하여 화면에 표시하는 방식을 의미한다.

나. 요구사항

- (1) 콘텐츠가 제공하는 텍스트나 그래픽 정보는 색상을 제거하더라도 그 내용을 인지할 수 있어야 한다.
- (2) 웹 페이지에서 보여주는 정보와 배경색 간에는 충분히 대비가 되어야 한다. 특히 웹 페이지의 내용을 고 대비(high contrast) 모드로 표시했을 때에도 충분히 명암대비가 되도록 콘텐츠를 설계하여야 한다.

다. 적용시 장점

- (1) 색상의 차이가 정보의 다름을 나타내지 않으므로 색각(색맹 또는 색약) 이상자의 경우에도 혼동을 일으킬 염려가 없게 된다.
- (2) 도표 등의 경우에 사용하는 색상의 명암 대비가 충분하므로 약시자의 경우에도 도표를 정확히 인지할 수 있다.
- (3) 고대비 모드를 사용할 수밖에 없는 사용자들도 콘텐츠의 내용이나 구조를 손쉽게 이해할 수 있다.



## 라. 적용 예

### (1) 그래프 및 차트 : 색깔보다는 무늬를 이용한 표현 권장

파이 차트, 막대그래프나 꺾은 선 그래프 등을 이용하여 정보를 표시하는 경우에 각각의 영역을 색깔로 구분하기보다는 여러 가지 무늬를 이용하여 표시하면 색상을 인식할 수 없는 사용자들, 예를 들면 약시자나 색각이상자가 쉽게 이해할 수 있다.

### (2) 중요한 항목 표시 : 색깔보다는 특수기호 사용 권장

예를 들어, 가입자의 신상 정보를 입력하는 웹 페이지에서 필수항목을 색깔(예를 들어 빨간색)을 사용하여 표시하는 경우에 이 색깔을 인지하지 못하는 색각 이상자의 경우에는 이 항목의 중요도를 인지할 수 없다. 따라서 색깔 대신 특수 기호(예를 들어 별표)를 이 항목에 추가하면 대부분의 사용자가 이 항목이 중요한 항목인지를 쉽게 인지할 수 있다.

**지침 2 운용의 용이성 - 웹 콘텐츠에 포함된 모든 요소들의 기능은 누구나 쉽게 사용할 수 있어야 한다.**

**항목 2.1 (이미지 맵 기법 사용 제한) 이미지 맵 기법이 필요할 경우에는 클라이언트측 이미지 맵을 사용하며 서버측 이미지 맵을 사용할 경우에는 동일한 기능을 하는 텍스트로 구성된 대체 콘텐츠를 제공해야 한다.**

## 가. 용어 정리

(1) 이미지 맵(image map)이란 이미지의 일부 영역에 링크를 두어 사용자에게 해당 정보를 제공할 수 있는 페이지로 이동하도록 만드는 프로그램 기법이다.

(2) 서버측 이미지 맵(server-side image map)이란 이미지 맵을 구현하는 프로그램이 서버측에 존재하여 사용자의 웹 브라우저가 이미지 맵 상의 어떤 그림 위에 마우스를 위치시키고 클릭하면 클라이언트가 마우스의 위치정보를 서버로 제공하고, 서버는 전달되어온 마우스의 위치에 해당하는 이미지 맵의 영역이 지시하는 링크로 이동한 결과를 클라이언트에 표시하는 방식의 프로그램 기법이다.

(3) 클라이언트측 이미지 맵(client-side image map)이란 HTML 태그를 이용하여 사용자의 브라우저에서 이미지 맵의 영역을 식별하여 직접 해당 영역이 지시하는 링크로 이동하도록 하는 프로그램 기법이다.

#### 나. 요구조건

(1) 구성하려는 이미지 맵의 형태를 클라이언트측에서 기하학적으로 표현하기 어려운 경우가 아니면 서버측 이미지 맵을 사용하지 않는다.

(2) 서버측 이미지 맵 기법을 사용할 경우에는 이미지 맵의 영역에 대응하는 텍스트 링크를 별도로 제공해야 한다(항목 1.1 참조).

(3) 클라이언트측 이미지 맵을 사용할 경우에는 이미지 맵의 영역에 대한 충분한 대체 텍스트를 제공하여야 한다(항목 1.1 참조).

#### 다. 적용시 장점

(1) 서버측 이미지 맵 방식에서 서버는 해당 영역에 대한 대체 텍스트를 사용자에게 제공하지 못하며, 단지 컴퓨터 화면상에 표시할 뿐이다. 따라서 이 항목을 적용하면 대체 텍스트 링크가 없는 서버측 이미지 맵의 사용을 제한하므로 시각장애인의 접근성을 높일 수 있게 된다.

- (2) 클라이언트측 이미지 맵을 사용하는 경우에도 영역을 표시하는 부분에 대체 텍스트를 함께 표시한다면 이미지 맵의 영역을 읽어주므로 시각장애인들이 이미지 맵의 어느 영역을 가리키고 있는지를 확인할 수 있다(항목 1.1 참조).

#### 라. 적용 예

- (1) 클라이언트측 이미지맵 : 대체 텍스트 지원

<AREA> 태그를 사용하여 클라이언트측 이미지 맵을 작성할 경우 대체 텍스트를 해당 영역에서 제공하여야 한다.

- (2) 서버측 이미지맵 : 링크목록 제공

서버측 이미지 맵을 사용할 경우에는 대체 텍스트로 구성된 링크목록을 같은 페이지에 제공하면 시각장애인들이 텍스트 링크를 사용하여 필요한 작업을 할 수 있다(항목 1.1 참조).

**항목 2.2 (프레임의 사용 제한) 콘텐츠를 구성하는 프레임의 수는 최소한으로 하며, 프레임을 사용할 경우에는 프레임별로 제목을 붙여야 한다.**

#### 가. 용어 정리 : 없음

#### 나. 요구조건

- (1) 웹 콘텐츠에는 가급적 프레임을 사용하지 않아야 한다. 만일 프레임을 사용하는 경우에도 사용하는 프레임의 수를 최소한으로 줄여야 한다.
- (2) 프레임을 사용할 경우에는 프레임 별로 서로 독특한(중복되지 않는) 제목을 부여하여 프레임을 식별할 수 있어야 한다.

#### 다. 적용시 장점

- (1) 화면을 볼 수 있는 사용자는 한 페이지를 여러 개의 영역으로 구분한 프레임을 볼 수 있다. 그러나 화면을 볼 수 없는 시각장애인에게 여러 개의 프레임으로 구성된 페이지는 모든 프레임의 내용을 인지하기 어렵다. 따라서 여러 개의 프레임으로 구성된 웹 페이지에서 각각의 프레임에 제목을 부여하면 활성화된 프레임의 제목을 읽어주기 때문에 프레임간의 이동이 매우 편리하다.

#### 라. 적용 예

- (1) 프레임 : 프레임별 제목 부여

여러 개의 프레임으로 구성된 웹 콘텐츠에서 각각의 프레임에 제목을 부여할 수 있다. 예를 들어, 두개의 프레임으로 구성된 웹 페이지의 경우에 각각의 프레임 제목을 “순이”와 “철수”라고 명명한다면 현재 활성화된 프레임의 이름을 읽어 볼 수 있으므로 엉뚱한 프레임에서 작업하는 잘못을 방지할 수 있다.

**항목 2.3 (깜박거리는 객체 사용 제한) 콘텐츠는 스크린의 깜빡거림을 피할 수 있도록 구성되어야 한다.**

#### 가. 용어 정리

- (1) **스크립트(script)**란 컴퓨터에 자동으로 처리를 실행하기 위한 명령을 나열한 파일로 일반적으로 텍스트 파일 형식으로 제공되는 것이다.

#### 나. 요구조건

- (1) 웹 콘텐츠에는 애니메이션 등과 같이 깜빡거리는 주파수의 범위가 3 Hz에서 49 Hz 사이인 콘텐츠 요소들을 포함하지 않아야 한다.

- (2) 만일 위의 (1)에서 명시한 요구조건을 만족할 수 없는 웹 콘텐츠는 깜빡거림이 있는 웹 페이지로 이동하기 전에 이 페이지에 깜빡거림이 있음을 사전에 사용자에게 경고해주어야 한다. 또한, 스크린의 깜빡거림이 배제된 대체 페이지를 별도로 제공하여야 한다(항목 1.1 참조).

#### 다. 적용시 장점

- (1) 광과민성 발작 증세를 지닌 사람들은 빛이 깜빡거리는 것에 반응하여 발작을 일으킨다. 특히 3Hz에서 49 Hz의 사이의 깜빡거림은 발작을 일으키는 원인이 되며, 20Hz 부근이 발작을 가장 잘 일으키는 주파수이다. 따라서 요구조건 (1)을 만족하는 콘텐츠의 경우에는 광과민성 발작증세가 있는 사용자도 접근이 가능하다.
- (2) 정신이 산만한 사람의 경우에는 지속적인 깜빡거림이 있는 콘텐츠를 집중하여 응시할 수 없으므로 이 요구조건을 만족하는 콘텐츠는 정신이 산만한 사람도 접근이 가능하다.

#### 라. 적용 예

- (1) 첫 페이지 구성 : 깜박거리는 개체 배제

웹사이트에 처음 접속했을 때에 스크린에 나타나는 첫 페이지에는 최소한 깜빡거림이 없도록 콘텐츠를 구성하며, 만일 다음 페이지에 깜빡거림이 있는 콘텐츠가 포함되어 있을 경우에는 경고문을 첫 페이지에 표시하고, 깜빡거리는 콘텐츠를 제거한 대체 페이지와 이 페이지로 이동하는 링크를 제공하도록 구성한다.

- (2) 스크립트 사용 : 스크립트 동작하지 않도록 설정

스크립트를 이용하여 깜빡이는 텍스트(blinking text)를 구성하면 스크립트가 동작하지 않도록 함으로써 깜빡임을 제거할 수 있다.

- (3) 프로그래밍 기법 : 객체를 깜박거리게 하는 태그 사용 배제  
<blink> 또는 <marquee> 태그는 사용하지 않는다.

#### 항목 2.4. (키보드로만 운용 가능) 키보드 (또는 키보드 인터페이스)만으로도 웹 콘텐츠가 제공하는 모든 기능을 수행할 수 있어야 한다.

##### 가. 용어정리

- (1) 키보드 인터페이스란 키보드 또는 특수한 입력장치로부터 응용프로그램이 필요한 모든 정보를 입력받도록 하는 장치를 의미한다.
- (2) 포인팅 디바이스(pointing device)란 마우스나 터치패드와 같이 컴퓨터 화면의 특정한 부위를 직접 지정할 수 있는 장치를 의미한다.
- (3) 음성 입력 장치란 음성으로 컴퓨터를 제어할 수 있도록 구성된 시스템 또는 시스템을 구성하도록 하는 프로그램을 의미한다.

##### 나. 요구조건

- (1) 웹 콘텐츠는 키보드 또는 장애를 극복하도록 도와주는 여러 가지 입력 장치를 사용하는 경우에도 콘텐츠가 제공하는 모든 기능을 사용할 수 있어야 한다.

##### 다. 적용시 장점

- (1) 포인팅 디바이스(pointing device)를 사용할 수 없는 시각장애인들도 웹 콘텐츠나 웹사이트 기능을 키보드만으로 사용할 수 있다.
- (2) 키보드 또는 키보드 인터페이스를 사용할 수 없는 지체장애인의 경우에는 키보드 대신 음성 입력 장치를 이용하여 웹 콘텐츠에 접근할 수 있다.

## 라. 적용 예

### (1) 이벤트 발생 방법 : 다양한 입력장치 사용 가능

웹 콘텐츠에서는 포커스인(focus-in), 포커스 아웃(focus-out), 그리고 활성화(activation)의 세 가지 이벤트가 주로 사용된다. 또한, 이들 이벤트는 포인팅 디바이스(예를 들어 마우스), 키보드, 음성 입력 시스템과 같은 다양한 입력 디바이스로 제어가 가능하다.

### (2) 콘텐츠의 제어 : 키보드 또는 키보드 인터페이스 사용 가능

콘텐츠를 구성하는 모든 링크와 데이터 개체의 실행과 선택은 키보드 또는 키보드 인터페이스를 사용하여 수행할 수 있도록 구성한다. 예를 들어, 웹 페이지를 구성하는 콘텐츠 요소들 간의 이동은 탭(tab)키를 사용하여 가능하도록 구성한다.

**항목 2.5 (반복 네비게이션 링크(repetitive navigation link))** 웹 콘텐츠는 반복적인 네비게이션 링크를 뛰어넘어 페이지의 핵심부분으로 직접 이동할 수 있도록 구성하여야 한다.

## 가. 용어 정리

- ① **반복적 네비게이션 링크(repetitive navigation link)**란 여러 개의 네비게이션 링크들이 연속적으로 나열되어 있는 링크 개체를 의미한다. 스크린 리더는 반복적 네비게이션 링크 개체를 구성하는 모든 링크들을 순서대로 읽어주므로 콘텐츠의 핵심 부분으로 이동하기 위해서는 모든 링크를 읽어주는 시간이 필요하게 된다.

## 나. 요구조건

- (1) 웹 콘텐츠 상에 반복적 네비게이션 링크 개체가 포함되어 있으며 이 링크 객체가 콘텐츠의 핵심부분보다 먼저 읽어주도록 구성된 경우에

이들 링크들의 읽기를 생략하고 직접 콘텐츠의 메인 부분으로 직접 이동할 수 있는 링크를 제공하여야 한다.

- (2) 하나의 긴 문장으로 구성된 웹 페이지는 이 페이지의 문장을 여러 개의 논리적인 절로 구분하고 각 절의 색인을 콘텐츠의 첫 부분에 포함시켜 원하는 절의 시작 부분으로 직접 이동할 수 있도록 웹 페이지를 구성한다.
- (3) 모든 웹사이트는 텍스트 또는 대체 텍스트가 포함된 텍스트 아닌 콘텐츠로 구성된 사이트 맵을 제공하여야 한다.

#### 다. 적용시 장점

- (1) 이 항목을 만족하는 경우 페이지의 상단이나 좌측 프레임에 반복적 네비게이션 링크개체가 포함되어 있다면 스크린 리더 사용자는 이들 링크 목록을 순서대로 읽어준 후에야 필요한 부분을 읽어주므로 매우 지루한 과정이 요구된다. 그러나 이 페이지의 처음에 핵심 부분으로 이동할 수 있는 링크를 제공하면 빠르게 필요한 위치로 이동할 수 있다.
- (2) 하나의 긴 문장으로 구성된 콘텐츠 중에서 사용자가 원하는 부분을 찾기 위해서는 처음부터 모두 읽어야 함으로 필요한 부분을 찾는 것은 매우 지루한 일이다. 따라서 문장의 시작 부분에 색인을 제공하여 필요한 부분으로 직접 이동하도록 하는 것은 매우 편리한 기능이다.
- (3) 사이트 맵을 제공하면 사용자는 이를 이용하여 필요한 정보가 위치한 페이지로 직접 이동할 수 있다.



## 라. 적용 예

### (1) 메인 콘텐츠로의 직접 이동 : 프레임과 링크 사용 예

만일 웹 콘텐츠를 두 개의 프레임으로 구분하고 왼쪽 프레임을 반복적 네비게이션 링크용 프레임으로, 오른쪽 프레임을 메인 콘텐츠를 제공하도록 구성할 경우를 생각해 보자. 스크린 리더는 왼쪽의 반복적 네비게이션 링크 프레임의 내용을 모두 읽은 후에야 비로소 중앙의 메인 콘텐츠를 읽어주게 된다. 따라서 왼쪽 프레임에 있는 모든 반복적 네비게이션 링크들을 우회하여 직접 메인 콘텐츠로 이동하는 버튼을 두고, 이 버튼에 “메인 콘텐츠로 이동”이라고 레이블을 붙인다면 반복적 네비게이션 링크 개체 모두를 읽지 않고도 메인 콘텐츠 부분으로 직접 이동이 가능하게 된다.

### (2) 문서 목차 : 긴 문서의 탐색용이

하나의 긴 문서로 구성된 웹 콘텐츠의 시작부분에 이 문서의 목차와 해당 위치로 직접 이동할 수 있는 링크를 제공하고, 문서가 나누어지는 부분에 문서의 처음, 다음 절, 목차 등으로 이동할 수 있는 링크를 두면 문서를 탐색하는데 매우 편리하다.

### (3) 웹사이트 : 사이트 맵의 운영

웹 콘텐츠의 첫 페이지에 사이트 맵으로 이동하는 링크를 제공하면 모든 페이지를 열람하지 않고도 필요한 웹 페이지로 직접 이동이 가능하다.

**항목 2.6 (반응시간의 조절기능) 실시간 이벤트나 제한된 시간에 수행하여야 하는 활동 등은 사용자가 시간에 구매받지 않고 읽거나, 상호작용을 하거나 응답할 수 있어야 한다.**

## 가. 용어정리

(1) **실시간 이벤트(real-time event)**란 콘텐츠의 저자가 시간을 통제할 수 없는 실시간 상황에 벌어지는 이벤트를 의미한다.

(2) 정해진 시간 내에 응답이 필요한 콘텐츠들은 다음과 같다.

- (가) 자동적으로 갱신되도록 구성된 콘텐츠
- (나) 몇 초 후에 다른 페이지로 이동하도록 구성된 콘텐츠
- (다) 깜빡이는 텍스트나 스스로 스크롤 하도록 구성된 텍스트
- (라) 짧은 기간 동안 나타났다 일정시간 후에 사라지는 대화창
- (마) 일정시간 동안 사용하지 않으면 페이지에 대한 접근이 강제 차단되거나 사용할 수 없게 되는 콘텐츠

(3) 팝업(Pop-Up) 창이란 활성화되어 있는 웹 브라우저에 담긴 콘텐츠가 어떤 동작에 의하여 또는 자동으로 새로 열린 웹 브라우저 창을 의미한다. 그러나 운영체제가 표시하는 메시지 창은 팝업 창으로 분류하지 않는다.

#### 나. 요구조건

(1) 웹 콘텐츠의 시간 제약 조건이 최소한 아래의 항목 중 하나는 만족하여야 한다.

- (가) 사용자가 시간제한 기능을 동작하지 않도록 할 수 있거나,
- (나) 일반적으로 사람들이 선호하는 시간의 10배 이상으로 제한 시간을 늘릴 수 있거나,
- (다) 주어진 시간이 종료하기 전에 경고를 발하며, 제한 시간을 늘리기 위하여 최소한 10초 이상 시간이 주어질 수 있어야 한다.

(2) 그러나 아래의 항목은 예외로 하되, 시간제약이 있음을 이 페이지에서 분명하게 알려 주어야 한다.

- (가) 경매 등과 같이 시간 제약이 있으나 다른 대안이 없는 경우거나,

(나) 실시간 게임이나 시간이 정해진 범위에서 작업해야 하는 테스트 등과 같이 시간 제약이 필수적인 경우

(3) 팝업(Pop-Up) 창은 가급적 사용을 피해야 한다. 그러나 꼭 사용해야 하는 경우에는 팝업 창이 열리기 전에 사용자에게 경고를 할 수 있어야 한다. 또한 팝업 창이 열리더라도 포커스가 새로 열린 팝업 창으로 이동해서는 안 된다.

#### 다. 적용시 장점

(1) 독서 장애, 인지 장애, 학습 장애를 지닌 사람들은 기록된 문서를 읽고 이해하는 데에 대부분의 사람들보다 더 많은 시간을 필요로 하는 경우가 종종 발생한다.

(2) 지체장애인들은 이동 객체에 대해 반응할 만큼 빠르게 동작하거나 정확하게 움직일 수 없다.

(3) 자주 갱신되는 콘텐츠에서 보조 기술이나 음성 브라우저 등이 정해진 시간 내에 이 콘텐츠를 읽거나 처리하는 것이 곤란한 경우에 시간 제약에 구애받지 않도록 할 필요가 있다.

(4) 팝업 창을 사용하면 시각장애인의 경우에 새로운 창의 열고 닫힘을 잘 알 수 없으므로 매우 불편하다. 그러나 부득이해서 팝업 창을 사용해야만 하는 경우에는 창이 열리기 전에 사용자에게 이를 알려주어 시각장애인들이 팝업 창이 열린다는 것을 사전에 인지하도록 하는 것이 좋다.

#### 라. 적용 예

(1) 시간제약이 없는 콘텐츠 구성

웹 콘텐츠를 구성할 경우에 시간제한을 두지 않도록 한다.

(2) 시간제한이 있는 콘텐츠 구성

시간제한을 둘 경우에는 이러한 사항을 사용자가 충분히 알 수 있도록 공지하도록 한다. 이를 위하여 경과시간과 남은 시간을 화면에 표시하거나 그 값을 스크린 리더로 사용자에게 알려줄 수 있다.

**지침 3. 이해의 용이성 - 사용자가 가능한 한 쉽게 이해할 수 있도록 콘텐츠나 제어 방식을 구성해야 한다.**

**항목 3.1 (데이터 테이블 구성) 데이터 테이블은 테이블을 구성하는 데이터 셀의 내용에 대한 정보가 충분히 전달될 수 있어야 한다.**

가. 용어 정리

(1) 데이터 테이블(data table)이란 테이블을 구성하는 데이터간의 논리적 관계를 표현하기 위하여 사용하는 형태이다.

나. 요구 조건

(1) 데이터 테이블은 테이블의 제목이나 테이블의 내용을 요약하여 제공해야 한다(항목 1.1 참조).

(2) 데이터 테이블은 데이터 셀별로 대응되는 모든 헤더를 확인할 수 있도록 구성하여야 한다.

다. 적용시 장점

(1) 데이터 테이블을 요구 조건에 맞추어 제작한 경우에는 임의의 데이터 셀에 대응되는 모든 헤더 값을 알 수 있다.

## 라. 적용 예

### (1) 간단한 데이터 테이블 구성

HTML을 이용하여 간단한 데이터 테이블을 구성하는 경우에 데이터 셀과 헤더는 각각 <TD>와 <TH> 태그를 사용하여 테이블을 구성한다.

### (2) 복잡한 데이터 테이블 구성

HTML에서 행을 그룹화 하는데 THEAD, TFOOT, TBODY 등의 속성을 사용하고, 열을 그룹화 하는 데는 COL, COLGROUP 등의 속성을 사용한다. 보다 복잡한 테이블의 구성을 위해서는 "axis", "scope", "headers" 등의 속성을 이용한다.

### (3) 데이터 테이블 설명

HTML 4.01에 따라 데이터 테이블을 구성하는 경우에 테이블에 대한 설명은 <TABLE ... summary = ""> 태그를 사용하여 표시할 수 있다. 또한 <CAPTION> 태그를 사용하여 테이블의 간단한 제목을 표시할 수도 있다. <TABLE title=""> 태그를 사용해도 동일한 효과를 얻을 수 있다. 즉, 데이터 테이블의 제목(title)은 도표의 첫 부분(상단 처음)에 오도록 하여 이어지는 자료가 데이터 테이블임을 알도록 하면 편리하다.

## 항목 3.2 (논리적 구성) 콘텐츠의 모양이나 배치는 논리적으로 이해하기 쉽게 구성하여야 한다.

## 가. 용어정리

(1) 배치용 테이블(layout table)이란 데이터 테이블과는 달리 문서의 모양을 만들기 위한 목적으로 사용하는 테이블이다.

(2) 스타일 시트(style sheet)란 문서의 표현 형태를 규정하는 일련의 명령문을 의미한다. 스타일 시트는 콘텐츠 제공자가 마련한 것, 사용자가

마련한 것, 웹 브라우저에 내장된 형태 등의 세 가지 경우가 있다. CSS(Cascading Style Sheets) 레벨 2가 스타일 시트의 대표적인 규정이다.

#### 나. 요구사항

- (1) 문서의 모양이나 콘텐츠의 배치를 위해서는 스타일 시트(style sheet)를 사용하여야 한다.
- (2) 배치용 테이블을 사용하여 콘텐츠의 모양이나 배치를 할 경우에는 테이블을 구성하는 모든 셀들을 왼쪽상단에서 오른쪽 하단에 이르는 순서대로 늘어놓았을 때에도 그 내용을 충분히 이해할 수 있도록 구성해야 한다.

#### 다. 적용시 장점

- (1) 논리적으로 구성된 웹 콘텐츠는 스타일 시트(style sheet)를 바꾸거나 기능을 제거하더라도 그 내용을 순서대로 읽을 수 있으므로 문서의 의미를 이해하는데 어려움이 없다.
- (2) 배치용 테이블은 종종 일정한 형태를 표현하기 위하여 사용된다. 이 경우에도 스크린 리더로 읽어주는 내용의 순서가 화면에 나타난 콘텐츠의 읽는 순서와 같아지게 된다.

#### 라. 적용 예

- (1) 배치용 테이블 : 배치용 테이블은 캡션(CAPTION)이나 타이틀(TITLE) 태그를 사용하지 않는다.

- (2) 스타일 시트 : 스타일 시트를 사용하여 콘텐츠의 모양이나 배치를 구성할 경우에는 스타일 시트 기능을 제거하거나 다른 스타일 시트를 사용하더라도 그 내용을 이해하기 쉽게 콘텐츠를 구성하여야 한다.

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

#### 가. 용어 정리

- (1) 온라인 서식(on-line form)이란 웹 브라우저 상에서 작성하여 제출하도록 제공되는 양식을 의미한다.
- (2) 애플릿(applet)이란 웹 페이지에 삽입되는 프로그램을 의미한다.
- (3) 플러그인(plugin)이란 웹 브라우저를 사용하는데 함께 사용되는 프로그램을 의미한다. 예를 들어 플래시(Flash) 등이 플러그인의 하나이다.

#### 나. 요구 조건

- (1) 온라인 서식을 구성하는 모든 서식 제어 요소, 예를 들면, 편집 상자(edit box), 라디오 버튼(radio button), 체크 박스(check box) 등은 레이블과 해당 서식 제어 요소간의 표시 순서가 일정하여야 한다.
- (2) 탭(tab)키를 이용하여 서식 제어 요소 간을 이동할 경우에 그 순서가 왼쪽 위에서 오른쪽 아래 부분으로 순차적인 이동이 가능하여야 한다.
- (3) 서식을 구성할 때에 보조기술을 사용하면 접근이 가능한 경우에만 스크립트, 애플릿, 플러그인 또는 다른 프로그램 요소를 사용한다.

만일 이것이 불가능할 경우에는 스크립트, 애플릿(applet), 플러그인(plug-in) 등 프로그램 요소의 기능을 정지시켜도 온라인 서식을 작성할 수 있어야 한다(항목 4.1 참조).

#### 다. 적용시 장점

- (1) 요구조건을 만족하는 경우에는 온라인 양식을 작성함에 있어서 모든 서식 제어 요소의 레이블과 서식 제어 요소간의 관계가 일관성을 가지게 되므로 사용자는 혼란을 피할 수 있게 된다.

#### 라. 적용 예

- (1) 편집 박스 프로그래밍 예 : 예를 들어, 레이블 다음에 편집 박스(edit box)가 위치하는 경우에는 다음과 같이 프로그래밍을 하면 된다.

```
<LABEL for="first_name">이름:
 <INPUT type="text" id="first_name" tabindex="1">
</LABEL>
```

이 경우에 커서가 편집 박스에 위치하면, 스크린 리더는 ‘이름’이라고 읽어준다.

- (2) 편집 박스의 기본값 제공 : 편집 박스는 기본값을 가능한 한 제공한다.
- (3) 편집 박스 테이블 구성 : 편집 박스를 데이터 테이블 형태로 구성할 경우에는 편집 박스별로 테이블 헤더를 읽어주도록 구성한다.

**지침 4 기술적 진보성** - 구성된 콘텐츠는 웹 브라우저의 종류, 버전 등에 관계없이 사용될 수 있어야 한다.



항목 4.1 (신기술의 사용) 스크립트, 애플릿 또는 플러그 인(plugin) 등과 같은 프로그래밍 요소들은 현재의 보조기술의 수준에서 이들 프로그래밍 요소들의 내용을 사용자에게 전달해줄 수 있을 경우에만 사용하여야 한다.

#### 가. 용어 정리

- (1) 애플릿(Applet)이란 웹 페이지에 삽입되는 프로그램을 의미한다.
- (2) 플러그인(plugin)이란 웹 브라우저를 사용하는데 함께 사용되는 프로그램을 의미한다. 예를 들어 플래시(Flash) 등이 플러그인의 하나이다.

#### 나. 요구 조건

- (1) 콘텐츠를 나타내기(display) 위해 혹은 인터페이스 요소를 만들기 위해 스크립트 언어를 이용할 경우에는 스크립트에 의해 제공되는 중요한 정보는 최신의 보조기술을 이용해 읽을 수 있어야 한다. 만일 이것이 불가능할 경우에는 스크립트의 동작을 정지시켜도 웹 콘텐츠의 내용을 읽을 수 있어야 한다.
- (2) 애플릿, 플러그인(plugin) 혹은 다른 응용 프로그램을 이용하여 웹 콘텐츠를 구성하였을 때에는 이들 프로그램 요소에 의해 제공되는 중요한 정보는 보조기술을 이용해 읽을 수 있어야 한다. 만일 이것이 불가능할 경우에는 이들 프로그램 요소의 동작을 정지시켜도 웹 콘텐츠의 내용을 읽을 수 있어야 한다.
- (3) 스크립트, 애플릿과 플러그인은 키보드 또는 대체 키보드 인터페이스를 이용하여 사용할 수 있어야 한다.

#### 다. 적용시 장점

- (1) 보조기술이 지원되는 경우가 아니면 애플릿, 플러그인, 스크립트 등의 프로그램 요소를 사용하지 않도록 제한하므로 장애인들이 이들 프로그램 요소에 의하여 접근이 차단되는 일을 방지할 수 있다. 만일 Javascript 기능을 지원하지 않는 웹사이트에서 Javascript 기능을 정지시켰을 때에도 사용자에게는 필요한 정보가 차단되지 않는다.

#### 라. 적용 예

- (1) 플래시(Flash)에 대한 대체 텍스트를 현재까지 개발된 스크린 리더가 읽을 수 없다면 플래시를 사용하지 않도록 한다.

항목 4.2 (별도 웹사이트 제공) 콘텐츠가 항목 1.1에서 4.1에 이르는 13개 검사 항목을 만족하도록 최대한 노력하였으나 해결되지 않는 부분이 남아있다면 텍스트만의 콘텐츠를 제공하는 웹 페이지(또는 웹사이트)를 별도로 제공해야 한다.

#### 가. 용어 정리

- (1) 텍스트만의 콘텐츠(text-only contents)란 텍스트 아닌 콘텐츠가 포함되지 않고 텍스트로만 구성된 콘텐츠를 의미한다.

#### 나. 요구 조건

- (1) 가능한 보조기술 수준이 미흡하여 장애인이 접근 가능한 웹 콘텐츠를 제작할 수 없는 경우에는 텍스트로만 구성된 대체 페이지를 마련하고 기존의 웹 콘텐츠의 첫 페이지에 대체 페이지로 이동하는 링크를 제공하여야 한다.

(2) 제공하는 대체 페이지는 기존의 웹 콘텐츠가 포함한 정보나 기능을 모두 포함하여야 한다.

(3) 제공하는 대체 페이지는 기존의 웹 콘텐츠의 개정 주기에 맞추어 개정되어야 한다.

#### 다. 적용시 장점

(1) 새로운 웹 콘텐츠 제작기술의 개발로 인하여 아무리 노력하여도 원래의 웹 콘텐츠에 포함된 일부 또는 모든 콘텐츠 요소들을 장애인의 접근이 가능한 콘텐츠로 수정할 수 없을 경우가 있다. 이 경우 대체 텍스트로 구성한 페이지를 별도로 구성하여 운영하면 장애인의 접근성이 충분히 지원된다.

#### 라. 적용 예

(1) 한글97 문서 : 별도의 웹 페이지로 구성

한글97로 작성된 문서를 콘텐츠로 제공하는 사이트의 경우, 스크린 리더를 통하여 한글97 문서를 읽을 수 없다. 이 경우에 동일한 내용을 별도의 HTML 문서로 제작하여 링크한다면 이러한 문제점이 해소된다.

## 웹 접근성을 고려한 콘텐츠 제작 기법

(157-715) 서울특별시 강서구 등촌1동 645-11

발행일 : 2004. 12.

발행처 : 한국정보문화진흥원 정보격차연구센터 접근기술연구팀

전 화 : (02) 3660 - 2656 FAX : (02) 3660 - 2659

E-mail : jhyun22@kado.or.kr URL : <http://www.kado.or.kr>

---