

웹 접근성을 고려한 콘텐츠 제작 기법 2.0

- 한국형 웹 콘텐츠 접근성 지침 2.0(KICS.OT-10.0003/R1,

2010년 12월 31일 제정)을 기준으로

2011. 11

연구책임자 : 최두진 단장 (한국정보화진흥원 정보사회통합지원단)

공동연구책임자 : 김석일 교수 (충북대학교 전기전자 및 컴퓨터공학부)

참여연구원 : 홍경순 부장 (한국정보화진흥원 정보접근지원부)

노석준 교수 (성신여자대학교 교육학과)

윤좌진 과장 (NHN)

신승식 차장 (현대자동차)

신현석 과장 (KTH)

현준호 책임 (한국정보화진흥원 정보접근지원부)

구경모 주임 (한국정보화진흥원 정보접근지원부)

목 차

I. 웹 접근성의 이해	11
II. 웹 접근성을 고려한 콘텐츠 제작기법	29
원칙 1. 인식의 용이성(Perceivable): 모든 콘텐츠는 사용자가 인식할 수 있어야 한다.	29
검사항목 1.1.1 (적절한 대체 텍스트 제공) 텍스트 아닌 콘텐츠는 그 의미나 용도를 이해할 수 있도록 대체 텍스트를 제공해야 한다.	31
검사항목 1.2.1 (자막 제공) 멀티미디어 콘텐츠에는 자막, 원고 또는 수화를 제공해야 한다.	55
검사항목 1.3.1 (색에 무관한 콘텐츠 인식) 콘텐츠는 색에 관계없이 인식될 수 있어야 한다.	62
검사항목 1.3.2 (명확한 지시사항 제공) 지시사항은 모양, 크기, 위치, 방향, 색, 소리 등에 관계없이 인식될 수 있어야 한다.	70
검사항목 1.3.3 (텍스트 콘텐츠의 명도 대비) 텍스트 콘텐츠와 배경 간의 명도 대비는 4.5대 1 이상이어야 한다.	77
검사항목 1.3.4 (배경음 사용 금지) 자동으로 재생되는 배경음을 사용하지 않아야 한다.	83
원칙 2. 운용의 용이성(Operable): 사용자 인터페이스 구성요소는 조작 가능하고 내비게이션 할 수 있어야 한다.	93
검사항목 2.1.1 (키보드 사용 보장) 모든 기능은 키보드만으로도 사용할 수 있어야 한다.	106
검사항목 2.1.2 (초점 이동) 키보드에 의한 초점은 논리적으로 이동해야 하며 시각적으로 구별할 수 있어야 한다.	115
검사항목 2.2.1 (응답시간 조절) 시간제한이 있는 콘텐츠는 응답시간을 조절할 수 있어야 한다.	115

검사항목 2.2.2 (정지 기능 제공) 자동으로 변경되는 콘텐츠는 움직임을 제어할 수 있어야 한다.	119
검사항목 2.3.1 (깜빡임과 번쩍임 사용 제한) 초당 3~50회 주기로 깜빡이거나 번쩍이는 콘텐츠를 제공하지 않아야 한다.	123
검사항목 2.4.1 (반복 영역 건너뛰기) 콘텐츠의 반복되는 영역은 건너뛴 수 있어야 한다.	128
검사항목 2.4.2 (제목 제공) 페이지, 프레임, 콘텐츠 블록에는 적절한 제목을 제공해야 한다.	133
검사항목 2.4.3 (적절한 링크 텍스트) 링크 텍스트는 용도나 목적을 이해할 수 있도록 제공해야 한다.	137
원칙 3. 이해의 용이성(Understandable): 콘텐츠는 이해할 수 있어야 한다.	140
검사항목 3.1.1 (기본 언어 표시) 주로 사용하는 언어를 명시해야 한다.	142
검사항목 3.2.1 (사용자 요구에 따른 실행) 사용자가 의도하지 않은 기능(새 창, 초점 변화 등)은 실행되지 않아야 한다.	145
검사항목 3.3.1 (콘텐츠의 선형화) 콘텐츠는 논리적인 순서로 제공해야 한다. ·	154
검사항목 3.3.2 (표의 구성) 표는 이해하기 쉽게 구성해야 한다.	165
검사항목 3.4.1 (레이블 제공) 입력 서식에는 대응하는 레이블을 제공해야 한다. ·	174
검사항목 3.4.2 (오류 정정) 입력 오류를 정정할 수 있는 방법을 제공해야 한다. ·	183
원칙 4. 견고성(Robust): 웹 콘텐츠는 미래의 기술로도 접근할 수 있도록 견고하게 만들어야 한다.	201
검사항목 4.1.1 (마크업 오류 방지) 마크업 언어의 요소는 열고 닫음, 중첩 관계 및 속성 선언에 오류가 없어야 한다.	202
검사항목 4.2.1 (웹 애플리케이션 접근성 준수) 콘텐츠에 포함된 웹 애플리케이션은 접근성이 있어야 한다.	209

적용 기술 목차

적용 기술 1) 이미지 등에 대한 대체 텍스트 제공	31
적용 기술 2) 버튼 이미지에 대한 대체 텍스트 제공	32
적용 기술 3) Java 애플릿 등에 대한 대체 텍스트 제공	32
적용 기술 4) 이미지 맵에 대한 대체 텍스트 제공	35
적용 기술 5) 텍스트 이미지, 그래픽 문자 등에 대한 대체 텍스트 제공	35
적용 기술 6) 플래시 콘텐츠의 대체 텍스트 제공	37
적용 기술 7) 실버라이트 콘텐츠의 대체 텍스트 제공	38
적용 기술 8) 그래프 등에 대한 설명문 제공	39
적용 기술 9) 생방송 콘텐츠에 대한 대체 텍스트 제공	40
적용 기술 10) CAPTCHA에 대한 대체 텍스트 제공	41
적용 기술 11) 온라인 시험 등에 대한 대체 텍스트 제공	42
적용 기술 12) 대체 텍스트를 빈 공간(alt="")이나 생략해서 제공해야 하는 경우 ·	42
적용 기술 13) 열린 자막 제공	55
적용 기술 14) 닫힌 자막 제공	56
적용 기술 15) 대본 또는 원고 제공	56
적용 기술 16) 수화 제공	57
적용 기술 17) 음성이 없는 동영상의 대체 수단 제공	57

적용 기술 18) 색을 보완하는 텍스트 제공	62
적용 기술 19) 색을 보완하기 위하여 글자모양을 이용하는 방법	63
적용 기술 20) 색을 보완하기 위하여 무늬 또는 모양을 이용하는 방법	64
적용 기술 21) 시각 정보를 이용한 지시문의 보완	70
적용 기술 22) 청각 정보를 이용한 지시문의 보완	71
적용 기술 23) 배경과 텍스트 콘텐츠의 색 지정 방법	77
적용 기술 24) 배경과 텍스트 콘텐츠 색을 기본 값으로 지정하는 방법	78
적용 기술 25) 배경과 텍스트 콘텐츠 색을 변경하는 컨트롤의 제공	78
적용 기술 26) 재생 시간이 3초 미만인 배경음의 사용	83
적용 기술 27) 배경음을 정지 상태로 제공하는 방법	84
적용 기술 28) 플래시 콘텐츠의 배경음 자동 실행 방지	84
적용 기술 29) 실버라이트 콘텐츠의 배경음 자동 실행 방지	89
적용 기술 30) 키보드와 마우스 이벤트 핸들러 제공	95
적용 기술 31) 마우스 드래그와 드롭 기능에서의 키보드 이용 보장	97
적용 기술 32) 웹 애플리케이션 제공 시 키보드 이용 보장	98
적용 기술 33) 실버라이트 플러그인의 키보드 접근 제공	101
적용 기술 34) 콘텐츠 화면 순서와 동일한 키보드 내비게이션 순서 제공	106
적용 기술 35) 시각적으로 표시 가능한 초점 표시 방법	107
적용 기술 36) 플래시에서 키보드 내비게이션 순서 제공 방법	110
적용 기술 37) 실버라이트에서 키보드 내비게이션 순서 제공 방법	111

적용 기술 38) 제한 시간 연장 방법 제공	115
적용 기술 39) 메타 태그를 이용한 페이지 재 이동 방법 제공	116
적용 기술 40) 변화하는 콘텐츠를 일시정지 시키고, 일시정지 된 곳으로부터 다시 시작할 수 있도록 제공	119
적용 기술 41) 전체 배너를 보여줄 수 있는 방법 제공	120
적용 기술 42) 전체 배너의 리스트 제공	120
적용 기술 43) 사용자 요구에 의한 업데이트 방법 제공	120
적용 기술 44) 깜빡이는 시간이 3초 이내인 콘텐츠만 제공	124
적용 기술 45) 경고 페이지 제공	125
적용 기술 46) 반복적인 콘텐츠 영역의 시작 부분에 건너뛰기 링크 제공	128
적용 기술 47) 건너뛰기 링크의 화면 표시	129
적용 기술 48) 웹 페이지의 제목 제공	133
적용 기술 49) 콘텐츠 블록의 제목 제공	134
적용 기술 50) 프레임 제목 제공	134
적용 기술 51) 링크의 목적을 설명하는 링크 텍스트 제공	137
적용 기술 52) <html> 태그에 주 사용 언어 지정	142
적용 기술 53) 초점 변화에 의해 맥락을 변화시키지 말고, 해당 인터페이스가 활성화(activate)되었을 때 기능을 실행	145
적용 기술 54) 명확한 서식 제출(submit) 버튼 제공	146
적용 기술 55) 새 창 열림을 사전에 알림	146
적용 기술 56) 콘텐츠를 의미 있는 순서로 배열	154

적용 기술 57) 스타일을 이용한 글자 간격 조절	157
적용 기술 58) 동적으로 생성된 요소는 그것을 유발시킨 요소 바로 뒤에 위치 ..	158
적용 기술 59) 표 제목을 <caption>으로 제공	165
적용 기술 60) 표의 구조 또는 내용에 대한 요약을 summary로 제공	166
적용 기술 61) 표의 헤더 셀과 데이터 셀의 관계 정의	167
적용 기술 62) 입력 서식과 <label>의 명시적 관계 제공	174
적용 기술 63) <label>을 제공할 수 없는 입력 서식에 title 제공	176
적용 기술 64) 여러 개의 유사한 입력 서식 묶기	178
적용 기술 65) 입력 값 또는 형식에 대한 예시 제공	183
적용 기술 66) 서버측 오류 검사	184
적용 기술 67) 클라이언트측 적합성 검사와 본문에 오류 메시지 표시	185
적용 기술 68) 오류 메시지를 스크립트 경고창으로 제공	194
적용 기술 69) 중요한 서식 제출 시 재확인 절차 제공	197
적용 기술 70) 여는 태그와 닫는 태그의 정확한 사용	202
적용 기술 71) 속성 이름과 속성 값의 정확한 사용	203
적용 기술 72) 태그의 정확한 중첩관계	204
적용 기술 73) 표준에 부합하는 태그와 속성 이름, 값 사용	205
적용 기술 74) 플러그인 플랫폼이 제공하는 접근성 API 활용	209
적용 기술 75) 웹 애플리케이션의 대체 콘텐츠 제공	210
적용 기술 76) 용도, 목적지, 종류, 사용법에 맞는 사용자 인터페이스 요소 사용 ·	212

잘못된 사례 목차

잘못된 사례 1) 의미나 기능이 있는 텍스트 아닌 콘텐츠에 빈 문자열을 대체 텍스트로 제공	45
잘못된 사례 2) 복잡한 정보를 하나의 이미지로 표현하고 간단한 대체 텍스트로만 제공	47
잘못된 사례 3) 주기적으로 변하는 콘텐츠의 대체 텍스트를 변경하지 않는 경우 ..	48
잘못된 사례 4) 의미 있는 정보를 배경 이미지로 제공	48
잘못된 사례 5) 빈칸 이미지의 대체 텍스트가 콘텐츠의 인식을 방해하는 경우	50
잘못된 사례 6) 도와 같은 텍스트 아닌 콘텐츠에 대한 설명이 불충분한 경우	51
잘못된 사례 7) 중요한 정보를 색으로만 제공하는 경우	51
잘못된 사례 8) alt 속성이 아닌 title 속성을 이용하여 대체 텍스트 제공	52
잘못된 사례 9) QR 코드에서 URL을 대체 텍스트로 알려주지 않는 경우	53
잘못된 사례 10) 자막, 원고, 수화 중 어느 한 가지도 제공하지 않는 경우	58
잘못된 사례 11) 동영상의 내용을 요약하여 제공하는 자막 또는 원고의 경우	59
잘못된 사례 12) 대체 수단에 대한 또 다른 대체 수단임을 알려주지 않는 경우 ...	60
잘못된 사례 13) 열린 자막이 수화를 가리는 동영상	61
잘못된 사례 14) 동영상의 해상도가 낮아 수화를 볼 수 없는 경우	61
잘못된 사례 15) CSS로 인하여 링크임이 구분되지 않는 경우	65
잘못된 사례 16) 색으로만 필수 입력 항목을 표시한 경우	66

잘못된 사례 17) 오류 메시지를 색으로만 구분하도록 한 경우	67
잘못된 사례 18) 그래프를 색으로만 인식하도록 하는 경우	68
잘못된 사례 19) 일정 구분을 색으로만 구분하도록 한 경우	69
잘못된 사례 20) 방향, 위치 정보만을 이용하여 사용법을 알려주도록 구현한 경우	72
잘못된 사례 21) 화면 위치만을 이용하여 객체를 지정한 경우	74
잘못된 사례 22) 버튼 모양만을 이용하여 사용법을 알려준 경우	74
잘못된 사례 23) 특수 기호만을 이용하여 사용법을 알려준 경우	76
잘못된 사례 24) 전경색은 기본 값으로 두고 배경색만을 지정한 경우	79
잘못된 사례 25) 배경색은 기본 값으로 두고 전경색만을 지정하는 경우	80
잘못된 사례 26) 배경음이 3초 이상 지속되도록 구현한 경우	91
잘못된 사례 27) 마우스 오버 또는 키보드 초점을 받으면 자동적으로 배경음이 실행되는 경우	92
잘못된 사례 28) 마우스로만 접근 가능한 이미지 버튼을 제공하는 경우	101
잘못된 사례 29) 키보드로 접근할 수 없는 플래시 메뉴	102
잘못된 사례 30) 마우스로만 접근할 수 있는 링크를 제공한 경우	103
잘못된 사례 31) 키보드만으로 이전 입력 서식 이동이 불가능한 경우	105
잘못된 사례 32) 초점을 일부러 보이지 않도록 한 경우	112
잘못된 사례 33) 키보드 이동순서가 비논리적인 경우	113
잘못된 사례 34) 페이지 재 이동 시 회피 수단을 제공하지 않는 경우	116
잘못된 사례 35) 마우스로만 정지할 수 있는 콘텐츠를 제공한 경우	121

잘못된 사례 36) 자동으로 변하는 배너	122
잘못된 사례 37) 깜빡이는 콘텐츠가 많은 경우	125
잘못된 사례 38) 발작 가능성이 있는 동영상을 제공한 경우	126
잘못된 사례 39) 번쩍임이 지속되는 영상을 제공한 경우	127
잘못된 사례 40) 건너뛰기 링크를 제공하지 않은 경우	130
잘못된 사례 41) 건너뛰기 링크를 과도하게 제공한 경우	131
잘못된 사례 42) 콘텐츠가 다른 페이지에 동일한 제목을 제공한 경우	135
잘못된 사례 43) 프레임 제목이 누락된 경우	135
잘못된 사례 44) 특수문자(☒☒☒☒)기호를 제공한 경우	136
잘못된 사례 45) 목적이나 용도를 알기 어려운 링크 텍스트	138
잘못된 사례 46) 목록 선택 상자에서 초점 변경만으로 새 창이 열리는 경우	148
잘못된 사례 47) 체크 상자의 선택만으로 페이지가 다시 로드되는 경우	150
잘못된 사례 48) 텍스트 입력 서식에 값을 넣으면 자동으로 제출되는 경우	151
잘못된 사례 49) 페이지가 로드될 때 자동으로 팝업 창이 열리도록 한 경우	152
잘못된 사례 50) 상위 메뉴와 하위 메뉴의 읽는 순서가 잘못된 경우	160
잘못된 사례 51) 배치용 표를 잘못 사용한 경우	162
잘못된 사례 52) 표를 이미지로 표현하고, 표 제목을 제공하지 않은 경우	170
잘못된 사례 53) 이메일 입력 서식의 레이블을 잘못 제공한 경우	180
잘못된 사례 54) 오류 정보를 확인한 후 오류 발생 페이지로 돌아가지 못하는 경우 ·	199
잘못된 사례 55) 아이디를 중복 선언한 경우	206

잘못된 사례 56) 태그의 중첩관계에 오류가 있는 경우	207
잘못된 사례 57) 이미지 링크를 자바스크립트로 잘못 구현한 경우	214
잘못된 사례 58) 잘못된 이미지 버튼 구현 사례	214

I. 웹 접근성의 이해

□ 웹 접근성의 개념

웹 접근성이란 어떠한 사용자(장애인, 고령자 포함), 어떠한 기술 환경에서도 전문적인 능력 없이 웹 사이트에서 제공하는 모든 정보에 비장애인과 동등하게 접근·이용할 수 있도록 보장해 주는 것이라 정의할 수 있다.

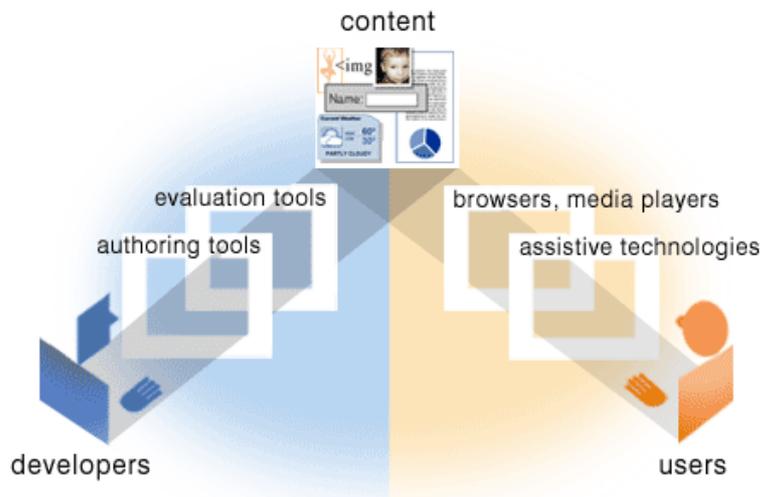
월드 와이드 웹 컨소시엄(W3C: World Wide Web Consortium)의 웹 접근성 이니셔티브(WAI: Web Accessibility Initiatives)에서는 웹 접근성을 “장애인도 웹을 인식하고, 이해하며, 내비게이션하고, 상호작용할 수 있으며, 그들이 웹에 기여할 수 있는 것을 의미한다. 웹 접근성은 또한 나이가 들어감에 따라 능력이 변화하는 고령자를 포함한 다른 사람들도 혜택이 있다.” (<http://www.w3.org/WAI/intro/accessibility.php>) 라고 정의하고 있다. WAI에서는 접근성의 4가지 핵심 원리로 인식의 용이성(perceivable), 운용의 용이성(operable), 이해의 용이성(understandable), 그리고 견고성(robust)을 제시하고 있으며, 장애인 뿐만 아니라 고령자, 비장애인도 접근성의 대상이 됨을 함축하고 있다.

□ 웹 접근성의 필수 요소들

웹 접근성을 준수하려면, 콘텐츠 자체의 접근성 준수만으로는 불가능하다. 즉, 콘텐츠뿐만 아니라 해당 콘텐츠의 개발자와 사용자 간에 상호작용 등을 가능하도록 하는 여러 가지 필수 요소들이 함께 작동하지 않으면 불가능하다. 이러한 필수 요소들을 제시하면 다음과 같다(<http://www.w3.org/WAI/intro/components.php>).

- 콘텐츠(contents): 웹 페이지 또는 웹 애플리케이션에 있는 정보
- 사용자 도구: 웹브라우저(Web browsers), 미디어 플레이어 등
- 보조공학(assistive technology): 스크린리더, 대체 키보드, 스위치, 스캔 소프트웨어 등
- 사용자(users): 웹을 사용하는 사람들(장애인 등 특별한 기술을 필요로 하는 사람들 포함)

- 개발자(developers): 웹 콘텐츠를 개발하는 사람들(장애를 지닌 개발자와 콘텐츠를 제공하는 사용자를 포함한, 설계자, 코딩하는 사람, 저자 등 포함)
- 저작도구(authoring tools): 웹 사이트를 제작하는 소프트웨어
- 평가도구(evaluation tools): 웹 접근성 평가도구, HTML 유효성 검사기(validators), CSS 유효성 검사기 등



출처: <http://www.w3.org/WAI/intro/components.php>

[그림 1] 웹 접근성 필수요소들 간의 관계

□ 웹 접근성 준수의 필요성

그렇다면 우리는 왜 웹 접근성을 준수해야 하는가? 우리가 웹 접근성을 준수해야 하는 대표적인 이유를 크게 네 가지로 나누어 보면, 다음과 같다.

첫째, 장애인, 고령자 등을 포함한 웹 이용자의 확대를 들 수 있다. 장애인의 경우 2000년에는 958천명으로 전체 인구의 2.01%, 2005년에는 1,777천명으로 전체 인구의 3.64%를 차지하였으나, 2010년에는 2,517천명으로 전체 인구의 5.18%를 차지하고 있는 것으로 나타났고, 고령자의 경우 2000년에는 3,395천명으로 전체 인구의 7.11%, 2005년에는 4,366천명으로 전체 인구의 8.95%를 차지하였으나, 2010년에는 5,357천명으로 전체 인구의 11.0%를 차지함을 알 수 있다. 이러한 통계치들을 보면, 장애인뿐만 아니라 고령자 수가 매년 지속적으로 증가하고 있음을 알 수 있다.

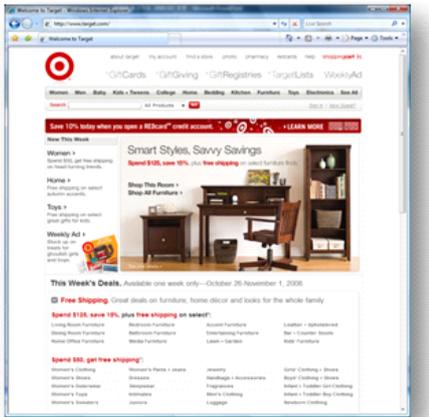
한편, 보건복지가족부와 한국보건사회연구원의 2008년 장애인 실태조사 결과에 따르면, 지체장애, 시각장애, 청각장애 등 대표적인 장애유형별 휴대폰, 컴퓨터, 인터넷 보유율과 사용률이 우리가 생각하고 있는 것보다는 훨씬 높음을 알 수 있다(<표 1> 참조). 바로 이러한 사람들이 웹을 보다 많이 이용하면 할수록 웹 접근성을 제대로 준수하지 않으면 그들이 겪는 접근성의 문제 또한 증가할 수밖에 없다.

<표 1> 장애유형별 휴대폰, 컴퓨터, 인터넷 보유율 및 사용률

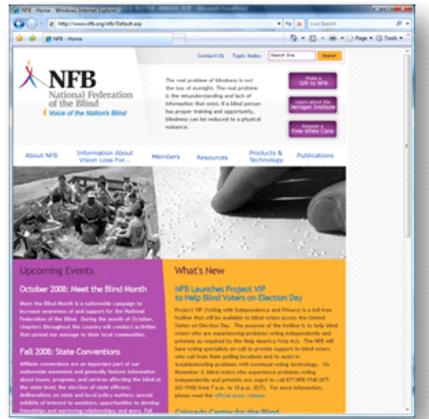
구분		지체 장애	뇌병변 장애	시각 장애	청각 장애	언어 장애	지적 장애	전체
휴대폰	보유율	77.4	50.0	74.4	59.5	58.0	27.1	67.1
	사용률	77.0	49.2	74.4	59.3	55.8	27.2	66.7
컴퓨터	보유율	55.3	52.2	48.8	48.5	57.0	55.7	53.0
	사용률	30.2	13.8	21.4	20.1	29.9	35.4	26.7
인터넷	보유율	53.8	50.6	48.1	45.6	55.2	53.7	51.4
	사용률	29.7	13.7	21.2	19.3	27.4	28.9	25.7

출처: 보건복지가족부, 한국보건사회연구원. 2008년 장애인 실태조사, p. 230.

둘째, 웹 접근성을 준수하는 것이 법률로서 의무화되었다. 이는 웹 접근성을 준수하지 않을 시 법률적인 처벌을 받게 됨을 의미한다. 아직까지 우리나라에서는 웹 접근성 미준수로 인해 법률적인 처벌을 받은 경우는 없다. 그러나 Target을 상대로 NFB(National Federation of the Blind)의 회원인 Bruce Sexton(시각장애인 학생)이 대체 텍스트를 제공하지 않고, 마우스가 아닌 키보드만으로 구매가 불가능하는 등 미국장애인법(ADA)을 위반했다는 이유로 2006년에 제소해 2008년 8월 27일에 Target이 6백만 달러(한화 60억원)를 지불(소송 참가 개인당 최대 \$7,000 지급)하도록 한 판결, Ramada.com과 Priceline.com 등에 대한 판결 등 외국의 경우를 보면 이는 언제든지 그리고 특정 분야가 아닌 모든 분야에 걸쳐 일어날 수 있음을 알 수 있다.



Target 홈페이지
(<http://www.target.com/>)



NFB 홈페이지
(<http://www.nfb.org/nfb/default.asp>)



Ramada.com 홈페이지
(<http://www.ramada.com/>)



Priceline.com 홈페이지
(<http://www.priceline.com/>)

셋째, 웹 접근성을 준수하게 되면 디자인 및 설계 등에 대한 효율성을 제고함으로써 비용절감 효과도 있다. Henry(2006)에 의하면, 일반적으로 “한 웹 사이트에서 웹 접근성 해결책을 실행하면 종종 기술적인 향상을 가져올 수 있다”고 한다. 접근성을 준수하지 않은 불필요한 이미지나 동영상 자료 등을 올리면 접근성을 준수할 수 없기 때문에 꼭 필요한 이미지나 자막, 수화 등이 포함된 동영상만을 올리게 된다. 그 결과, 서버에 대한 부하량이 감소하게 되며, 이는 다운로드 속도의 증가를 가져올 수 있다. 또한 단 한가지의 장치나 브라우저가 아닌 여러 장치 또는 다양한 브라우저에서도 접근할 수 있기 때문에 상호작용성과 장치 독립성을 증진할 수 있으며, CSS(Cascading Style Sheets)를 사용함으로써 사이트 개발 시간을 단축하고 유지관리 시간이나 비용도 절감할 수 있다.

넷째, 공공기관 및 기업의 이미지 제고와 이윤을 창출하기 위해 접근성을 준수할 필요가 있다. 그것이 공공기관이든 민간 기업이든 어떤 조직이 웹 사이트를 접

근 가능하게 만들기 위해 노력을 하면 할수록 해당 기관은 긍정적인 투자 대비 회수율과 비용 효율성을 가져올 수 있다.

웹의 중요한 혜택들 중 한 가지는 융통성과 웹이 제공하는 서비스에 대한 접근의 용이성에 기인한다. 실제로, 이것은 전화, 면대면 또는 우편과 같은 다른 수단을 통해 공공서비스나 다른 서비스들과 상호작용하는 데 필요한 시간, 노력, 비용의 절감을 의미한다. 접근성의 결여는 많은 사람들이 대안적인 채널들을 사용함으로써 보다 높은 처리비용을 지속적으로 지출해야 함을 의미한다.

특히, 공공기관이 웹 접근성을 준수함으로써 얻을 수 있는 투자 대비 회수율 또는 비용 효율성은 최근 EU의 연구결과를 보면 보다 명확하게 알 수 있다(<표 2> 참조).

<표 2> 접근 가능한 전자정부를 위한 비용-효과성 모형 -연간 순수 비용절감액

(단위: 유로; 대상: EU 25개국)

대상집단 도달율	경제적 비용효과성 범주	웹 사이트 접근성을 달성하기 위한 추가적인 비용			
		2%	5%	15%	30%
5%	전자접근성을 위한 정부 투자 비용	-24,256,800	-60,641,999	-181,925,997	-363,851,994
	정부의 처리비용 절감액	158,223,882	158,223,882	158,223,882	158,223,882
	순수 정부 비용 대비 이익	133,967,082	97,581,883	-23,702,115	-205,628,112
	국민의 이익	153,112,707	153,112,707	153,112,707	153,112,707
	전체 비용 대비 이익	287,079,790	250,694,590	129,410,592	-52,515,405
10%	전자접근성을 위한 정부 투자 비용	-24,256,800	-60,641,999	-181,925,997	-363,851,994
	정부의 처리비용 절감액	316,447,764	316,447,764	316,447,764	316,447,764
	순수 정부 비용 대비 이익	292,190,964	255,805,765	134,521,767	-47,404,230
	국민의 이익	306,225,415	306,225,415	306,225,415	306,225,415
	전체 비용 대비 이익	598,416,379	562,031,180	440,747,181	258,821,184
20%	전자접근성을 위한 정부 투자 비용	-24,256,800	-60,641,999	-181,925,997	-363,851,994
	정부의 처리비용 절감액	632,895,528	632,895,528	632,895,528	632,895,528
	순수 정부 비용 대비 이익	608,638,729	572,253,529	450,969,531	269,043,534
	국민의 이익	612,450,829	612,450,829	612,450,829	612,450,829
	전체 비용 대비 이익	1,221,089,558	1,184,704,358	1,063,420,360	881,494,363

출처: European Commission, 2008, p. 10.

<표 2>를 보면, 웹 사이트를 이용하는 사람(장애인, 노인)들의 비율을 높일수록 접근성을 위한 비용 추정치가 내려가 전반적인 비용 대비 회수율은 더 높아짐을 알 수 있다. 즉, 이 웹 접근성에 관한 투자 대비 비용효과성 모형은 전자정부 서

비스 등에 대한 접근성 준수율이 높아질수록 또는 국민들이 웹을 통해 정보에 접근할 수 있는 비율이 높아질수록 정부와 국민들은 다른 수단들을 통해 정보에 접근하는 데 드는 처리비용을 현격하게 절감할 수 있음을 잘 보여주고 있다.

마지막으로, googlebot과 같은 검색엔진에서 찾는 정보들을 명시적으로 제시함으로써, 즉 검색엔진에 최적화함으로써 자신의 홈페이지를 홍보하기 위해 별도의 비용을 지출하지 않고도 사이트를 홍보할 수 있어 궁극적으로 웹 사이트의 활용률을 증가시킬 수 있다. 또한 이미 살펴본 바와 같이, 접근성을 준수하지 않은 것으로 인해 소송을 당하게 되면 높은 법률비용을 지출해야 하지만 접근성을 준수하면 그러한 비용을 지출할 필요가 없으며, 하나의 접근 가능한 콘텐츠를 통해 다양한 특성을 지닌 사용자들의 요구를 수용할 수 있기 때문에 장애유형별로 별도의 자료를 제작·배포할 필요가 없게 된다.

< 참고 > 웹 접근성 vs 웹 사용성

웹 사용성 전문가인 Nielsen(1993)은 웹 사용성의 다섯 가지 구성 요소로 학습가능성(learnability), 사용의 효율성(efficiency of use), 기억가능성(memorability), 오류 허용한계(error tolerance) 또는 효과성(effectiveness), 그리고 주관적 만족(subjective satisfaction)을 제시하였다.

학습 가능성이란 버튼의 레이블이나 용어 등이 간단명료해 쉽게 학습할 수 있는가의 여부를 말하며, 사용의 효율성이란 복잡하고 여러 단계를 거쳐 수행해야 하는 작업을 한두 번의 조작만으로, 커다란 노력이나 비용을 들이지 않고도, 신속하게 처리할 수 있어야 함을 의미한다. 또한 기억가능성이란 한두 번의 학습만으로도 특정 기능이나 내용을 쉽게 기억할 수 있어야 함을 의미하며, 오류허용한계 또는 효과성이란 사용자가 사소한 실수나 부주의, 착오 등으로 인해 실수를 범할지라도 해당 실수를 복구할 수 있도록 해 주며, 그러한 수정 과정을 통해 당초 계획했던 행동이나 목표를 보다 성공적으로 달성할 수 있도록 해주는 것을 말한다. 또한 주관적 만족이란 사용자가 특정 웹 사이트를 이용하는 데 있어 색상 등이 자신의 취향이나 선호도에 맞게 배열되어 보다 편안한 분위기에서 웹 서핑을 함으로써 궁극적으로 만족감을 얻는 것을 의미한다.

일반적으로, 웹 접근성은 Nielsen(1993)이 말한 다섯 가지 구성요소 중 처음 네 가지 요소들, 즉 학습가능성, 사용의 효율성, 기억가능성, 오류 허용한계 또는 효과성과 밀접한 관련이 있는 반면, 주관적 만족은 포함되지 않는다. 예를 들어, 접근성에서는 전경색과 배경색이 충분한 정도의 가독성에만 초점을 맞춘다. 사용자가 검정색 바탕에 노란색 글씨를 선호하지만 검정색에 흰색 글씨를 사용하였다고 제시되는 문제는 사용성 관련 문제이다. 이처럼, 사용성은 사용자들이 기능성(functionality)을 얼마나 잘 사용할 수 있는가에 관한 문제, 즉 웹 사용성은 사용자가 애플리케이션의 작동, 목적, 콘텐츠를 얼마나 잘 이해할 수 있도록 할 것인가를 나타내는 척도(Mueller, 2003)인 반면, 접근성은 사용성을 위해 필요하지만 모든 사용성 설계 원칙들이 접근성에 적용되지는 않는다. 한마디로, 접근성은 사용자가 애플리케이션과 얼마나 잘 상호작용할 수 있는가를 나타내는 척도(Mueller, 2003)라 할 수 있다. 따라서 접근성은 사용성을 위한 필요조건이지 충분조건은 아니며, 바로 이 점에서 접근성은 사용성과 차이가 있다.

□ 장애인의 웹 사용 시 애로점

우리나라의 장애인복지법 시행령 [별표 1]에는 동 법 시행령 제2조(장애인의 종류 및 기준)에 근거해 장애인의 종류와 기준을 구체적으로 명시해 놓고 있다. 이 시행령에 명시된 장애인은 지체장애, 뇌병변장애, 시각장애, 청각장애, 언어장애, 지적장애, 자폐성장애, 정신장애, 신장장애, 심장장애, 호흡기장애, 간장애, 안면장애, 장루·요루장애, 간질장애의 총 15가지 장애 중 하나 이상의 장애를 지닌

사람을 말한다.

한편, 이들 장애인의 종류와 기준은 해당 장애인의 특성에 따라 보다 세분화될 수 있다. 예를 들어, 시각장애인의 경우 전맹, 약시, 색각장애 등으로 나눌 수 있다. 웹 접근성을 고려할 때는 일반적으로 이러한 세분화된 특성들을 고려하는 것이 일반적이다.

이미 언급한 바와 같이, 웹 접근성을 준수하면 다양한 특성을 지닌 장애인들뿐만 아니라 고령자, 비장애인 모두에게 혜택을 준다. W3C는 웹에서 접근성 장애를 겪을 수 있는 대상별로 시나리오를 작성한 후 각 대상별 접근성 준수방안을 제시하고 있다. W3C에서 제시한 장애유형별 대표적인 웹 접근성 장애를 요약·제시하면 다음과 같다(<http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web/>).

1) 시각장애

- 대체 텍스트를 가지고 있지 않은 이미지
- 적절하게 기술되지 않은 복잡한 이미지(예: 그래프나 차트)
- 텍스트나 오디오로 기술되지 않은 비디오
- 순차적으로(셀 단위로 또는 “선형화된” 모드로) 읽을 때 의미를 알 수 없는 표
- “NOFRAME” 대체수단을 가지고 있지 않거나 의미 있는 이름들을 가지고 있지 않는 프레임
- 논리적인 순서로 탭을 할 수 없거나 부적절하게 레이블된 서식
- 모든 명령어들에 대한 키보드 지원이 되지 않는 브라우저와 저작도구들
- 기반이 되는 운영체제의 표준 애플리케이션 프로그래밍 인터페이스를 사용하지 않은 브라우저와 저작도구들
- 화면낭독 프로그램(screen reader)이 해석하기 어려운 비표준 문서 서식
- 쉽게 변경되지 않는(크게 하거나 작게 할 수 없는) 절대 글자 크기가 사용된 웹 페이지
- 일관성 없는 배열로 인해 확대했을 경우 주변의 콘텐츠를 잃어 버려 내비게이션하기가 어려운 웹 페이지

- 부적절한 명도대비로 인해 명도대비를 사용자가 자신의 스타일시트로 대체하거나(override) 쉽게 변경할 수 없는 웹 페이지나 웹 페이지상의 이미지
- 확대했을 경우 다음 줄로 자동 변경되는 것을 차단한 이미지로 된 텍스트
- 웹 사이트에 있는 텍스트를 강조하기 위해 독특한 표시물(marker)로 사용된 색상
- 배경색이나 패턴과 부적절하게 대비된 텍스트

2) 청각장애 및 언어장애

- 웹 방송(webcasts 등을 포함), 웹상의 오디오에 관한 자막이나 원고가 없는 경우
- 제1언어가 문어나 구어가 아닌 수화인 사람들이 이해하기 어려운 텍스트로 꽉 찬 페이지에서 콘텐츠 관련 이미지가 없는 경우
- 간단명료한 언어가 제공되지 않는 경우
- 웹 사이트에서 음성 입력을 요구하는 경우
- 음성기반 상호작용을 요하거나 어떠한 대체 입력 모드도 가지고 있지 않은 웹 사이트

3) 지체장애

- 웹 페이지상의 시간제한 반응 옵션
- 마우스 명령어에 대한 키보드 대체물을 지원하지 않는 브라우저와 저작도구들
- 논리적인 순서로 탭(tab)할 수 없는 서식

4) 지적 및 신경장애

- 시각을 보완하기 위해 오디오로 변환할 수 있는 대체 텍스트를 제공하지 않거나 오디오에 대한 자막을 제공하지 않는 경우와 같이, 웹 사이트에 있는 정보에 대한 대체 유형/포맷을 제공하지 않는 경우
- 쉽게 끌 수 없도록 된, 주의를 산만하게 하는 시각 또는 비디오 요소

- 웹 사이트가 명료하고 일관성 있게 조직되지 않은 경우
- 불필요하게 복잡한 언어를 사용한 경우
- 절대 글자 크기를 사용한 웹 페이지
- 발작을 촉발할 수 있는 시각적 또는 오디오 주기 사용

한편, W3C는 몇 가지 대표적인 장애유형별 웹 접근성 문제를 준수했을 때 장애 등으로 인한 콘텐츠의 접근 문제는 더 이상 문제가 되지 않는다고 주장한다. W3C가 제시한 대표적인 장애유형별 접근성 준수 사례들을 제시하면 <표 3>과 같다(<http://www.w3.org/WAI/bcase/soc>).

<표 3> 대표적인 장애유형별 웹 접근성 준수 사례

장애 유형	준수 사례
시력이 감퇴하고 있는 고령자	<ul style="list-style-type: none"> • 전경색과 배경색 간의 충분한 명도대비 • 약간의 시각장애를 지닌 사람들이 화면확대기와 같은 보조공학을 사용할 필요 없이 직접 읽을 수 있도록, 읽기 쉬운 글꼴과 줄 간격을 늘릴 수 있는 것과 글자 크기를 크게 할 수 있는 텍스트 • 정보를 전달하기 위해 텍스트의 비트맵 이미지를 사용하는 것 대신에 더 나은 브라우저 기반 확대를 할 수 있는 스타일이 맞추어진 텍스트 • 사용자들을 산만하게 하거나 발작을 유발하지 않는 방식으로 깜빡거리거나 번쩍이거나 움직이지 않는 텍스트와 다른 요소
민첩성이 떨어지거나 미세운동 조절을 하기 어려운 고령자	<ul style="list-style-type: none"> • 객체의 클릭 가능한 영역을 늘릴 수 있는 기능 • 모든 웹 사이트의 상호작용(장치 독립성)을 위해 마우스를 사용하는 것 대신에 키보드를 사용할 수 있도록 한 것
청각이 손상된 고령자	<ul style="list-style-type: none"> • 오디오 콘텐츠에 대한 원고와 자막 • 오디오의 전경 “정보”와 배경 “소음” 간의 대비
문해능력이 낮은 사람과 언어가 유창하지 않은 사람	<ul style="list-style-type: none"> • 간단명료한 언어 • 보충적인 설명 • 명료하고 일관성 있는 디자인, 내비게이션, 링크 • 그룹으로 나누어진 정보 묶음 • 너무 많이 깜빡거리거나 번쩍이거나 움직이지 않는 텍스트 • 사용자들에게 콘텐츠를 읽고 사용할 수 있는 충분한 시간 제공
저대역 인터넷 접속이나 오래된 테크놀로지를 사용하는 사람	<ul style="list-style-type: none"> • 이미지, 멀티미디어, 그리고 다른 비텍스트 객체들에 대한 대체텍스트 • 색상으로 전달되는 정보를 위한 풍부한(redundant) 코딩과 전경색과 배경색 간의 충분한 명도대비 • 상대값으로 정의된 텍스트 크기

	<ul style="list-style-type: none"> • 정보를 전달하기 위해 텍스트의 비트맵 이미지 대신에 다운로드 속도를 높여줄 수 있는 스타일에 맞춘 텍스트 • 콘텐츠를 프레젠테이션으로부터 분리하기 위해 효과적으로 사용된, 그렇게 함으로써 파일 크기와 파일 다운로드 요구조건들을 줄여 다운로드 속도를 증가시킬 수 있는 스타일시트 • 몇몇 오래된 테크놀로지는 스타일시트를 처리할 수 없기 때문에, 스타일시트 없이도 읽을 수 있고 이해할 수 있도록 조직화된 사이트 • 타당한 W3C 테크놀로지를 사용하며 오래된 테크놀로지에서도 작동할 가능성이 높은 사이트 • 사용자들이 원하는 페이지들을 열 수 있도록 도와주고 잘못된 경로를 거쳐 가는 것을 막아줌으로써 낭비되는 페이지 로딩 시간을 절약할 수 있도록 도와주는, 명료하고 일관성 있는 디자인, 내비게이션, 그리고 링크
<p>신규 사용자와 웹을 빈번하게 사용하지 않는 사용자</p>	<ul style="list-style-type: none"> • 명료하고 일관성 있는 디자인, 내비게이션, 그리고 링크 • 이미지맵에 대한 풍부한 텍스트 링크 제공 • 새로운 브라우저 창이 열리기 전에 사용자에게 공지

□ 웹 접근성 관련 국내의 법률과 표준

웹 접근성 관련 국내의 법률이나 표준 등에는 어떠한 것들이 있는지 살펴보고, 특히 웹 접근성에 관해 보다 구체적으로 명시해 놓은 「장애인차별금지 및 권리구제 등에 관한 법률」(이하 ‘장애인차별금지법’)과 「장애인차별금지법 시행령」에 대해 좀 더 구체적으로 살펴보면 다음과 같다.

먼저, 웹 접근성 관련 국내 법률과 정책, 지침으로서는 「장애인·노인·임산부 등의 편의증진에 관한 법률」 제4조(접근권), 「장애인복지법」 제20조(정보에의 접근), 2005년 12월에 국가표준으로 승인된 “인터넷 웹 콘텐츠 접근성 지침(IWCAG 1.0)”, 2008년 4월 11일부터 시행되고 있는 ‘장애인차별금지법’ 제21조(정보통신·의사소통에서의 정당한 편의제공의무), 그리고 2009년 5월 22일 공포된, 기존 「정보화촉진기본법」의 전부개정법인 「국가정보화기본법」 제32조(장애인·노인 등의 정보 접근 및 이용 보장) 등을 들 수 있다.

제32조(장애인·고령자 등의 정보 접근 및 이용 보장) ① 국가기관 등은 인터넷을 통하여 정보나 서비스를 제공할 때 장애인·고령자 등이 쉽게 웹 사이트를 이용할 수 있도록 접근성을 보장하여야 한다.

② 「정보통신망 이용촉진 및 정보보호 등에 관한 법률」 제2조제3호에 따른 정보통신서비스 제공자(이하 "정보통신서비스 제공자"라 한다)는 그 서비스를 제공할 때 장애인·고령자 등의 접근과 이용의 편의를 증진하기 위하여 노력하여야 한다.

③ 정보통신 관련 제조업자는 정보통신기기 및 소프트웨어(이하 "정보통신제품"이라 한다)를 설계, 제작, 가공할 때 장애인·고령자 등이 쉽게 접근하고 이용할 수 있도록 노력하여야 한다.

④ 국가기관등은 정보통신제품을 구매할 때 장애인·고령자 등의 정보 접근과 이용 편의를 보장한 정보통신제품을 우선하여 구매하도록 노력하여야 한다.

⑤ 행정안전부장관은 장애인·고령자 등의 정보 접근 및 이용 편의 증진을 위한 정보통신서비스 및 정보통신제품 등의 종류·지침 등을 정하여 고시하여야 한다.

특히, ‘장애인차별금지법’ 제21조와 ‘장애인차별금지법시행령’ 제14조(정보통신·의사소통에서의 정당한 편의제공의 단계적 범위 및 편의의 내용)는 공공 및 민간 웹 사이트에 대한 웹 접근성 준수를 단계적으로 의무화하고 있다. 이를 좀 더 구체적으로 살펴보면, 다음과 같다.

먼저, ‘장애인차별금지법’ 제21조는 장애인이 비장애인과 동등하게 전자정보에 접근, 이용하는 데 필요한 수단, 즉 웹 접근성을 보장하는 웹 사이트를 보장받아야 한다고 명시하고 있다.

제21조(정보통신·의사소통 등에서의 정당한 편의제공의무) ① 제3조제4호·제6호·제7호·제8호가목 후단 및 나목·제11호·제18호·제19호에 규정된 행위자, 제12호·제14호부터 제16호까지의 규정에 관련된 행위자, 제10조제1항의 사용자 및 같은 조 제2항의 노동조합 관계자(행위자가 속한 기관을 포함한다. 이하 이 조에서 "행위자 등"이라 한다)는 당해 행위자 등이 생산·배포하는 전자정보 및 비전자정보에 대하여 장애인이 장애인 아닌 사람과 동등하게 접근·이용할 수 있도록 수화, 문자 등 필요한 수단을 제공하여야 한다. 이 경우 제3조제8호가목 후단 및 나목에서 말하는 자연인은 행위자 등에 포함되지 아니한다.

② 공공기관 등은 자신이 주최 또는 주관하는 행사에서 장애인의 참여 및 의사소통을 위하여 필요한 수화통역사·문자통역사·음성통역사·보청기기 등 필요한 지원을 하여야 한다.

③ 「방송법」 제2조제3호에 따른 방송사업자와 「인터넷 멀티미디어 방송사업법」 제2조제5호에 따른 인터넷 멀티미디어 방송사업자는 장애인이 장애인 아닌 사람과 동등하게 제작물 또는 서비스를 접근·이용할 수 있도록 폐쇄자막, 수화통역, 화면해설 등 장애인 시청 편의 서비스를 제공하여야 한다.

④ 「전기통신사업법」에 따른 기간통신사업자(전화서비스를 제공하는 사업자만 해당한다)는 장애인이 장애인 아닌 사람과 동등하게 서비스를 접근·이용할 수 있도록 통신설비를 이용한 중계서비스(영상통화서비스, 문자서비스 또는 그 밖에 방송통신위원회가 정하여 고시하는 중계서비스를 포함한다)를 확보하여 제공하여야 한다.

⑤ 다음 각 호의 사업자는 장애인이 장애인 아닌 사람과 동등하게 접근·이용할 수 있도록 출판물(전자출판물을 포함한다. 이하 이 항에서 같다) 또는 영상물을 제공하기 위하여 노력하여야 한다. 다만, 「도서관법」 제18조에 따른 국립중앙도서관은 새로이 생산·배포하는 도서자료를 점자, 음성 또는 확대문자 등으로 제공하여야 한다.

1. 출판물을 정기적으로 발행하는 사업자
2. 영화, 비디오물 등 영상물의 제작업자 및 배급업자

⑥ 제1항에 따른 필요한 수단을 제공하여야 하는 행위자 등의 단계적 범위 및 필요한 수단의 구체적인 내용과 제2항에 따른 필요한 지원의 구체적인 내용 및 범위와 그 이행 등에 필요한 사항, 제3항과 제4항에 따른 사업자의 단계적 범위와 제공하여야 하는 편의의 구체적 내용 및 그 이행 등에 필요한 사항은 대통령령으로 정한다.

이를 보다 구체적으로 실현하기 위해, ‘장애인차별금지법 시행령’에는 각 행위주체별로 접근성 준수 의무화 단계를 설정해 놓고 있다. <표 4>에서 보는 바와 같이, 국가 및 지방자치단체, 대통령령으로 정한 공공기관의 경우 1년 이내, 즉 2009년 4월 11일까지, 그 외 대상은 1년에서 5년 내, 즉 2013년 4월 11일까지 단계적으로 웹 접근성을 준수하도록 하고 있으며, 장애인에게 정당한 편의 제공 규정을 위반할 경우 최고 3,000만원의 과태료가 부과되고, 행위가 악의적인 경우 3년 이하 징역이나 3,000만원 이하의 벌금을 부과할 수 있도록 규정하고 있다.

<표 4> 장애인차별금지법 시행령에 명시된 웹 접근성 준수 의무화 단계별 대상기관

년도	공공기관	교육기관 (책임자)	의료기관	복지시설	문화예술체육	법인
'09.4.11	<ul style="list-style-type: none"> ▪ 공공기관 	<ul style="list-style-type: none"> ▪ 국공사립특수 학교 ▪ 특수 학급 이 설치된 국공립 각급학교 ▪ 장애전담 보 육시설 	<ul style="list-style-type: none"> ▪ 종합병원 	<ul style="list-style-type: none"> ▪ 사회복지시설 (사회복지관 등) ▪ 장애복지시설 (요양 및 재활 시설 등) 		<ul style="list-style-type: none"> ▪ 상시 300명 이상의 근로자 사용 사업장과 국가 및 지방자치단체 *
'10.4.11					<ul style="list-style-type: none"> ▪ 국립문화예술 단체, 박물관, 미술과 ▪ 국립중앙도서관, 공공도서관 	
'11.4.11	<ul style="list-style-type: none"> ▪ 국공립 유치원 ▪ 『초·중등교육법』, 『고등교육법』에 따른 국공사립 각급 	<ul style="list-style-type: none"> ▪ 일반병원, 치과, 한방병원 (임원 30인 이상) 				<ul style="list-style-type: none"> ▪ 상 시 100-300명의 근로자 사용 사업장*

	<ul style="list-style-type: none"> ▪ 학교 ▪ 영재학교, 영재교육원 ▪ 보육시설(100인 이상) 					
'12.4.11					<ul style="list-style-type: none"> ▪ 민간종합공영장 	
'13.4.11	<ul style="list-style-type: none"> ▪ 사립유치원 ▪ 평생교육시설, 연수기관 ▪ 직업훈련기관(1,000㎡이상) ▪ 국공립 보육시설 및 법인 설립 보육시설 	<ul style="list-style-type: none"> ▪ 그 외 병원(입원 30인 이하) 			<ul style="list-style-type: none"> ▪ 체육관련 행위자 	<ul style="list-style-type: none"> ▪ 모든 법인
'15.4.11					<ul style="list-style-type: none"> ▪ 민간 종합공영장 및 소공영장(300석 미만) ▪ 영화관(300석 미만) ▪ 사립박물관미술관 	

※ 법인인 사용자가 사용자의 지위에서 제공하는 정보(예: 직원의 모집·채용, 임금 및 복리후생 등의 정보)가 아닌 경영공시 및 제품판매를 위한 정보의 경우 예외

특히, 여기에서 주목할 사항은 현재 단계적 웹 접근성 준수 대상이 아니더라도 직원의 모집·채용, 임금 및 복리후생 등에 관한 정보와 같이 법인인 사용자가 사용자의 지위에서 제공하는 정보는 접근성을 준수해 제공해야 한다는 점이다.

다음으로, 웹 접근성 관련 대표적인 국외 법률이나 표준을 살펴보면 다음과 같다. 먼저, 웹 접근성에 대해서는 호주의 장애차별금지법(Disability Discrimination Act: DDA), 브라질의 접근성에 관한 법(Law on Accessibility, L. 10.098), 벨기에의 차별금지법(Anti-Discrimination Law), 핀란드의 JHS 129 공공행정에서 웹 서비스를 설계하기 위한 지침(JHS 129 Guidelines for Designing Web Services in the Public Administration), 독일의 무장애정보기술에 관한 법령(Ordinance on Barrier-Free Information Technology), 영국의 장애인차별금지법 1995(Disability Discrimination Act 1995), 일본의 기본 IT법(Basic IT Law), 미국의 재활법 제508조(Section 508 of the Rehabilitation Act) 등을 포함해 전 세계의 상당히 많은 나라들이 웹 설계에 관한 법률이나 정책, 지침 등을 가지고 있다.

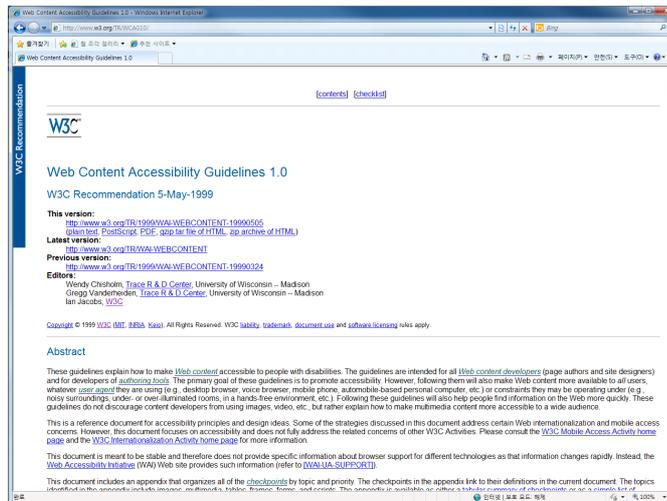
특히, 웹 접근성에 관한 법률과 지침들을 마련하고 이를 체계적으로 잘 수행하

고 있는 국가들 중 대표적인 두 국가를 예를 들어 살펴보면 다음과 같다.

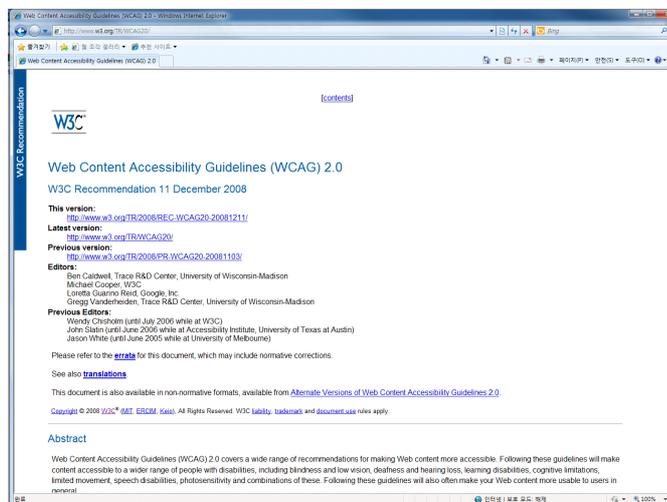
먼저, 미국은 2000년 12월에 W3C의 지침 중 우선순위(Priority) 1과 2를 기초로 16개 항목에 대한 웹 접근성 표준(Section 508 Standards) 제정하였으며, 2001년 6월부터 「재활법 제508조(Section 508 of the Rehabilitation Act)」에 의거 연방정부 및 소속기관에서 제공하는 모든 웹 사이트에 웹 접근성 준수를 의무화하였다. 또한 2006년 7월부터 W3C의 WCAG 2.0 개정에 발맞추어 해당 표준의 개정을 진행 중에 있다.

영국은 2004년 10월에 「장애인차별금지법(Disability Discrimination Act, 1995년 제정)에 의거해 모든 웹 사이트 이용에 웹 접근성을 준수할 것을 의무화하여 시행하고 있다. 영국 정부는 이를 보다 체계적으로 지원하기 위해 2006년부터 W3C의 WCAG 1.0을 활용해 「PAS(Publicly Available Specification) 78」이란 웹 접근지침 해설, 평가방법 등을 포함한 가이드라인을 마련·제공하고 있다.

또한 웹 접근성에 관한 국제표준화기구인 W3C 역시 웹 접근성에 관한 여러 가지 가이드라인을 마련·제공하고 있는데, 그 대표적인 것이 바로 1999년 5월 5일에 제정·공표된 웹 콘텐츠 접근성 지침(Web Content Accessibility Guidelines: WCAG) 1.0이다. 이 WCAG 1.0은 총14개의 가이드라인과 각 가이드라인에 따라 한 개에서 7개까지의 체크포인트들(checkpoints)로 구성되어 있다. 또한 이들 체크포인트 각각은 1부터 3까지의 우선순위(Priority)을 가지고 있는데, 1은 반드시 준수해야 하며(must), 2는 가급적 준수해야 하고(should), 3은 준수하면 좋음(may)을 의미한다. 이 WCAG 1.0은 많은 나라들이 그대로 채택 활용하거나 수정해 사용하고 있다. 한편, W3C는 2008년 12월에 WCAG 2.0을 발표하였다.



<그림 2> W3C의 WCAG 1.0 홈페이지(<http://www.w3.org/TR/WCAG10/>)



<그림 3> W3C의 WCAG 2.0 홈페이지(<http://www.w3.org/TR/WCAG20/>)

우리나라도 W3C의 WCAG 1.0 중 우선순위 1과 2 중 일부를 토대로 “인터넷 웹 콘텐츠 접근성 지침(IWCAG 1.0)” (2005. 12. 국가표준으로 승인)을 마련해 시행하고 있다. 또한 다른 나라들과 마찬가지로, 우리나라도 W3C의 WCAG 2.0의 개정에 발맞추어 WCAG 2.0 및 미국의 Section 508 개정 내용 등을 수용해 한국형 웹 콘텐츠 접근성 지침(KWCAG) 2.0을 개정하였으며, 2010년 12월 31일에 이를 국가표준으로 지정하였다.

[참고문헌]

- 보건복지가족부, 한국보건사회연구원 (2009). 2008년 장애인 실태조사.
 - ◆ European Commission (2008). Accessibility of ICT products and services to Disabled and Older People: Evidence-based analysis for a possible co-ordinated European approach to web accessibility. 2010. 12. 20일 접속. http://ec.europa.eu/information_society/newsroom/cf/itemdetail.cfm?item_id=4722.
 - ◆ Henry, S. L. (2006). Understanding Web accessibility. In Thatcher, J. et al. (2006). Web accessibility: Web standards and regulatory compliance.
 - ◆ Mace, R. L. (1985). Universal design: Barrier free environments for everyone. Designers West, 33(1), 147-152.
 - ◆ Mueller, J. P. (2003). Accessibility for everybody: Understanding the Section 508 accessibility requirements. New York: Apress.
 - ◆ Nielsen, J. (1994), Usability Engineering. Boston, MA: AP Professional.
 - ◆ Vanderheiden, G., & Tobias, J. (2000). Universal design of consumer production: Current industry practice and perceptions. 2002. 7. 16일 접속. http://trace.wisc.edu/docs/ud_consumer_products_hfes2000/

[참고할만한 자료]

- 웹 접근성 관련 법률, 표준, 지침 정보
 - ◆ 국가정보화 기본법 및 시행령(국가법령정보센터(<http://www.law.go.kr/main.html>))에서 검색 가능)
 - ◆ 장애인차별금지 및 권리구제 등에 관한 법률 및 시행령(국가법령정보센터(<http://www.law.go.kr/main.html>))에서 검색 가능)
 - ◆ 장애인·노인·임산부 등의 편의증진보장에 관한 법률 및 시행령(국가법령정보센터(<http://www.law.go.kr/main.html>))에서 검색 가능)
 - ◆ 장애인복지법 및 시행령(국가법령정보센터(<http://www.law.go.kr/main.html>))에서 검색 가능)
 - ◆ 장애인 등에 대한 특수교육법 및 시행령(국가법령정보센터(<http://www.law.go.kr/main.html>))에서 검색 가능)
 - ◆ 한국형 웹 콘텐츠 접근성 지침 2.0 : <http://www.wah.or.kr/Example2.0/kwcag2.0.pdf>
 - ◆ Section 508 of the Rehabilitation Act: <http://www.section508.gov/>
 - ◆ Guide to good practice in commissioning accessible websites(PAS 78: 2006): http://www.equalityhumanrights.com/uploaded_files/pas78.pdf
 - ◆ Common Look and Feel Standards 2.0 Canada: <http://www.tbs-sct.gc.ca/clf2-nsi2/>
 - ◆ Disability and the Equality Act 2010 UK: http://www.direct.gov.uk/en/DisabledPeople/RightsAndObligations/DisabilityRights/DG_4001068
 - ◆ W3C의 웹 접근성 지침 1.0(WCAG 1.0): <http://www.w3.org/TR/WCAG10/>
 - ◆ W3C의 웹 접근성 지침 2.0(WCAG 2.0): <http://www.w3.org/TR/WCAG20/>

○ 웹 사용성(Web Usability) 관련 정보

- ◆ The Usability Professional's Association: <http://www.usabilityprofessionals.org/>
- ◆ Usability.gov: <http://www.usability.gov/>
- ◆ Jakob Nielsen's Alertbox: <http://www.useit.com/alertbox/>
- ◆ Webusability: <http://webusability.com/>
- ◆ Web Usability: <http://www.usability.com.au/>

○ 보편적 설계(Universal Design) 또는 보편적 학습 설계(Universal Design for Learning: UDL) 관련 정보

- ◆ Center for Applied Special Technology(CAST): <http://www.cast.org/index.html>
- ◆ The Center for Universal Design: <http://www.ncsu.edu/www/ncsu/design/sod5/cud/index.htm>

II. 웹 접근성을 고려한 콘텐츠 제작기법

원칙 1. 인식의 용이성(Perceivable): 모든 콘텐츠는 사용자가 인식할 수 있어야 한다.

□ 인식의 용이성

웹 사이트에서 제공하는 콘텐츠를 이용하기 위해서는 해당 콘텐츠를 정확히 인식할 수 있어야 한다. 시각장애인은 시각을 이용하여 인식하도록 만들어진 이미지를 인식할 수 없다. 여기에서 텍스트는 화면 낭독 프로그램을 이용하면 읽어줄 수 있으므로 시각장애인이 인식할 수 있는 콘텐츠로 간주한다.

약시자와 고령자는 통상적인 크기의 폰트로 표시된 텍스트 콘텐츠를 인식하기 어렵다. 색각장애인은 특정한 색으로 표시된 콘텐츠의 구별이 어렵다. 청각장애인은 동영상의 음성, 오디오 등과 같은 청각 콘텐츠를 인식할 수 없다.

콘텐츠를 인식하기 어려운 경우는 비단 장애인에게만 국한된 것이 아니다. 다음과 같은 특별한 환경에서의 콘텐츠는 비장애인이 인식하는 경우도 어려움을 준다.

- 직사광선이 내려 쪼이는 장소에 설치된 모니터에 표시되는 콘텐츠
- 매우 작은 폰트를 사용한 콘텐츠
- 흑백 모니터에 표시되어 색의 차이가 드러나지 않는 콘텐츠
- 소음이 매우 심한 환경에서 제공되는 청각을 이용하는 콘텐츠

이와 같이 장애 또는 주변 여건 등으로 인하여 인식이 불가능하거나 어려움을 겪는 콘텐츠는 사용자가 그 내용을 동일하게 인식할 수 있도록 추가적인 정보를 함께 제공해야 한다. 따라서 이 원리에 따라 제작된 콘텐츠는 특정 감각을 이용할 수 없거나 특별한 환경에서 콘텐츠에 접근하는 사용자들도 그 정보를 다양한 방법(예: 보조공학 이용)으로 인식할 수 있게 된다.

또한 동영상의 오디오 트랙, 음성이나 사운드와 같은 멀티미디어 콘텐츠는 청각

장애인이 그 내용을 인식하기 어렵다. 따라서 멀티미디어 콘텐츠는 그 내용을 인식할 수 있도록 자막, 원고, 화면 해설과 같은 대체 콘텐츠를 제공해야 한다.

시각을 이용하여 콘텐츠를 인지하는 사용자의 경우도 비슷한 채도의 사물이나 명도 대비가 낮은 사물을 구분하기 어렵다. 그러므로 콘텐츠를 명료하게 전달하기 위해서는 명도 대비와 채도 대비가 뚜렷해야 한다.

□ 관련 지침

인식의 용이성은 사용자가 장애유무 등에 관계없이 웹 사이트에서 제공하는 모든 콘텐츠를 동등하게 인식할 수 있도록 제공하는 것을 의미한다. 인식의 용이성은 대체 텍스트, 멀티미디어 대체 수단, 명료성의 3가지 지침으로 구성되어 있다.

<표 5> 인식의 용이성 관련 지침 및 검사항목

지침(3개)	검사항목(6개)
1.1(대체 텍스트) 텍스트 아닌 콘텐츠에는 대체 텍스트를 제공해야 한다.	1.1.1(적절한 대체 텍스트 제공) 텍스트 아닌 콘텐츠는 그 의미나 용도를 이해할 수 있도록 대체 텍스트를 제공해야 한다.
1.2(멀티미디어 대체 수단) 동영상, 음성 등 멀티미디어 콘텐츠를 이해할 수 있도록 대체 수단을 제공해야 한다.	1.2.1(자막 제공) 멀티미디어 콘텐츠에는 자막, 원고 또는 수화를 제공해야 한다.
1.3(명료성) 콘텐츠는 명확하게 전달되어야 한다.	1.3.1(색에 무관한 콘텐츠 인식) 콘텐츠는 색에 관계없이 인식될 수 있어야 한다.
	1.3.2(명확한 지시사항 제공) 지시사항은 모양, 크기, 위치, 방향, 색, 소리 등에 관계없이 인식될 수 있어야 한다.
	1.3.3(텍스트 콘텐츠의 명도 대비) 텍스트 콘텐츠와 배경 간의 명도 대비는 4.5대 1 이상이어야 한다.
	1.3.4(배경을 사용 금지) 자동으로 재생되는 배경음을 사용하지 않아야 한다.

검사항목 1.1.1 (적절한 대체 텍스트 제공) 텍스트 아닌 콘텐츠는 그 의미나 용도를 이해할 수 있도록 대체 텍스트를 제공해야 한다.

□ 개요

이미지 등 텍스트가 아닌 콘텐츠를 이용할 경우, 해당 이미지가 제공하는 의미나 용도를 가급적 동일하게 인식할 있도록 적절한 대체 텍스트를 제공해야 한다. 텍스트로 제공할 수 있는 콘텐츠를 이미지 등 텍스트 아닌 콘텐츠로 제공하는 것은 바람직하지 않은 방법이므로 지양해야 한다. 또한 대체 텍스트는 간단명료하게 제공해야 한다.

□ 적용 기술

적용 기술 1) 이미지 등에 대한 대체 텍스트 제공

웹 페이지에 이미지를 포함시키기 위하여 img 태그를 사용하는 경우, alt 속성을 이용하여 짧은 대체 텍스트를 제공한다.

이미지에 대한 대체 텍스트는 이미지의 시각적인 특징보다는 이미지가 표현하려는 의미나 기능에 중점을 두어야 한다.

예를 들어, 1988 서울올림픽 웹 사이트로 연결되는 이미지 링크를 제공할 경우 이 이미지에 대한 대체 텍스트를 "1988 서울올림픽" 또는 "1988 서울올림픽에 대해 알아보기" 등과 같이 제공한다.

```

```

<참고> 적절한 대체 텍스트 작성 방법

1) 이하에서 ___은 개선 전 소스와 개선 후 소스의 비교 부분 또는 수정된 부분을 나타냄

특정 감각만을 이용해서 인식할 수 있는 콘텐츠는 해당 감각이 없는 사용자들도 인식할 수 있어야 한다. 예를 들어, 오디오 파일은 청각장애인이 이용할 수 없다. 시각적 이미지는 시각장애인이 인식할 수 없다. 따라서 오디오 파일이나 이미지는 각각 청각장애인과 시각장애인이 인식할 수 있는 대체 수단을 제공해야 한다. 대체 텍스트는 대체 수단의 한 가지 방법이다.

텍스트 아닌 콘텐츠에 대한 대체 텍스트의 적절성을 판단하기 위해서는 대체 텍스트가 다음 질문에 대한 답을 제시하는가를 살펴보아야 한다.

- 1) 텍스트 아닌 콘텐츠를 이곳에 제공하는 이유가 무엇인가?
- 2) 텍스트 아닌 콘텐츠로 제공하려는 정보는 무엇인가?
- 3) 텍스트 아닌 콘텐츠가 의미하는 내용이 무엇인가?
- 4) 텍스트가 아닌 콘텐츠가 제공하는 기능이나 정보를 어떻게 설명할 것인가?

예를 들어, 미술품 '모나리자'의 사진에 대한 대체 텍스트로는 "레오나르도 다빈치작 모나리자"가 적절하다. 장한나가 연주하는 '브람스 첼로 소나타 No. 1'의 MP3 파일에 대한 대체 텍스트로는 "장한나 연주, 브람스 첼로 소나타 1번"이 적절하다.

위에 열거한 방법은 HTML 기반의 콘텐츠뿐만 아니라, 자바스크립트, 플래시, 실버라이트 등과 같은 리치 인터넷 애플리케이션(Rich Internet Application: RIA)을 구성하는 모든 텍스트 아닌 콘텐츠에도 적용된다.

적용 기술 2) 버튼 이미지에 대한 대체 텍스트 제공

이미지를 이용한 입력 버튼은 input 태그를 사용하여 구현할 수 있다. 이 경우, 버튼에 사용된 이미지의 대체 텍스트는 alt 속성을 이용하여 제공한다.

```
<form action="http://sample.co.kr/submit/test" method="post">  
  <input type="image" name="submit" src="button.gif" alt="확인" />  
</form>
```

적용 기술 3) Java 애플릿 등에 대한 대체 텍스트 제공

웹 콘텐츠를 이용하여 Java 애플릿 등을 제공하는 방법은 applet 태그를 사용하는 방법, embed 태그를 사용하는 방법, object 태그를 사용하는 방법이 있다.

- applet 태그의 사용

Java 애플릿을 웹 페이지에 포함시켜 제공하기 위하여 applet 태그를 사용하는 경우, alt 속성을 이용하여 애플릿에 대한 대체 텍스트를 제공할 수 있다.

```
<applet code="TicTacToe.class" width=120 height=120 alt="tic tac toe 게임">
  tic tac toe 게임
</applet>
```

<참고>

- 1) applet 태그는 HTML 4.0 이후에는 사용하지 않도록 권고하는 방법임
- 2) 웹 브라우저는 applet 태그 내의 대체 텍스트를 읽어주나 웹 브라우저에 따라서 alt 속성을 읽어주는 경우도 있으므로 위의 예와 같이 두 곳에 동일한 대체 텍스트를 제공하는 것이 바람직함
- 3) 우리나라에서 통용 중인 센스리더와 Jaws for Windows 는 applet 태그 내의 대체 텍스트를 읽어줌

- embed 태그의 사용

Java 애플릿을 웹 페이지에 포함시켜 제공하기 위하여 embed 태그를 사용하는 경우 alt 속성을 이용하여 대체 텍스트를 제공한다.

```
<embed type="application/x-java-applet" code="Tic.class"
  width=120 height=120 alt="tic tac toe 게임">
</embed>
```

참고로, 대체 텍스트가 아니라 Java 애플릿에 대한 대체 수단을 제공하고자 하는 경우 noembed 태그를 사용하여 대체 수단을 제공한다.

```
<embed type="application/x-java-applet" code="Tic.class"
  width=120 height=120 alt="tic tac toe 게임">
  <noembed>
    <a href="tac_tac_toc.htm">Tic Tac Toe 게임</a>
  </noembed>
</embed>
```

noembed 태그는 다음과 같이 embed 태그 밖에 제공해도 된다.

```
<embed type="application/x-java-applet" code="Tic.class"
      width=120 height=120 alt="tic tac toe 게임">
</embed>
<noembed>
  <a href="tic_tac_toc.htm">Tic Tac Toe 게임</a>
</noembed>
```

<참고>

- 1) embed 태그는 광범위하게 사용되나 HTML 4.01 표준에 정의되지 않은 요소임. 최근에 제안된 HTML5에는 표준에 포함되었음
- 2) 우리나라에서 통용 중인 화면 낭독 프로그램은 embed에 대한 대체 텍스트를 지원하지 못하므로 embed 태그를 사용하기 위해서는 화면 낭독 프로그램(screen reader)의 지원여부를 먼저 확인해야 함

• object 태그의 사용

object 태그를 이용하여 웹 페이지에 표시하는 java 애플릿은 title 속성을 이용해서도 대체 텍스트를 제공할 수 있다. title 태그 대신 object 태그 내에 대체 텍스트를 포함시킬 수도 있다.

```
<object type="application/x-java-applet" data="Tic.class"
      width=120 height=120 title="tic tac toe 게임">
  <param name="code" value="Tic.class">
  tic tac toe 게임
</object>
```

대체 수단을 제공할 경우 object 태그를 중첩하여 구현할 수 있다. 예를 들어, java 애플릿을 제공하지 못하는 경우 이미지(tic_tac_toc.gif)를 제공하는 방법은 다음과 같다.

```
<object type="application/x-java-applet" data="Tic.class"
      width=120 height=120 title="tic tac toe 게임">
  <param name="code" value="Tic.class">
  <object data="tic_tac_toc.gif" type="image/gif">
```

```
Tic Tac Toc 게임
</object>
</object>
```

<참고>

- 1) 웹 브라우저에 따라 object 태그의 title 속성을 읽어주지 못하고 object 태그 내의 대체 텍스트를 읽어 주므로 두 곳에 모두 제공하는 것이 좋음
- 2) 우리나라에서 통용 중인 센스리더와 Jaws for Windows는 object 태그 내에 제공한 대체 텍스트만을 읽어줌

적용 기술 4) 이미지 맵에 대한 대체 텍스트 제공

이미지 맵(image map)은 이미지를 여러 개의 선택 가능한 영역으로 구분하고 각 영역을 링크로 연결한 콘텐츠이다. 이 경우, 선택 가능한 영역은 area 태그로 구현한다. 이때 area 태그별로 alt 속성을 이용하여 해당 영역의 용도나 목적을 대체 텍스트로 제공한다.

다음의 예는 이미지를 두 개의 영역으로 나누고, 각 영역에 각각 "ENGLISH"와 "KOREAN"이라는 대체 텍스트를 제공하도록 구현한 경우다. 여기에서 전체 이미지(language.gif)에 대해서도 대체 텍스트("language")를 제공하고 있다.



<language.gif>

```

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

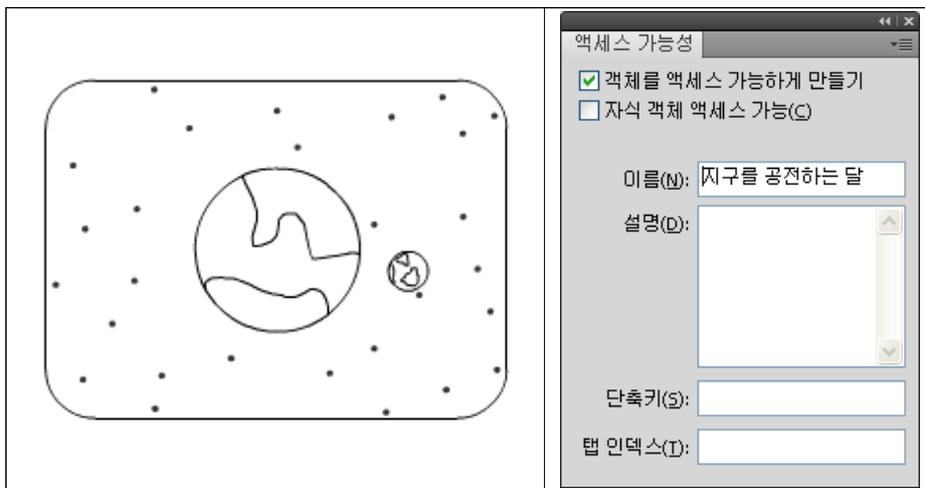
적용 기술 5) 텍스트 이미지, 그래픽 문자 등에 대한 대체 텍스트 제공

한글이나 영문자를 형상화한 로고 등과 같은 텍스트 이미지는 해당 이미지에 포함된 단어 또는 문장을 대체 텍스트로 제공한다. 텍스트 이미지에 대한 대체 텍

적용 기술 6) 플래시 콘텐츠의 대체 텍스트 제공

플래시(Flash)를 이용한 콘텐츠 구현 시 콘텐츠에 대한 대체 텍스트는 액세스 가능성(Accessibility) 패널의 '객체를 액세스 가능하게 만들기(Make object accessible)' 항목을 체크하고, '이름(N)' 항목에 대체 텍스트를 제공한다. 설명(D) 항목은 보다 자세한 설명문을 제공하는 경우에 사용한다.

- 다음의 예는 달이 지구를 공전하는 플래시 애니메이션으로 액세스 가능성 패널의 이름 항목을 이용하여 대체 텍스트를 제공하고 있다.



- 플래시는 스크립트 언어를 이용한 코딩 시 대체 텍스트를 제공할 수 있다. 예를 들어, ActionScript 2.0를 이용하여 프로그램을 개발하는 경우 무비클립(MovieClip)에 대한 `_accProps` 객체의 속성을 설정해야 한다.

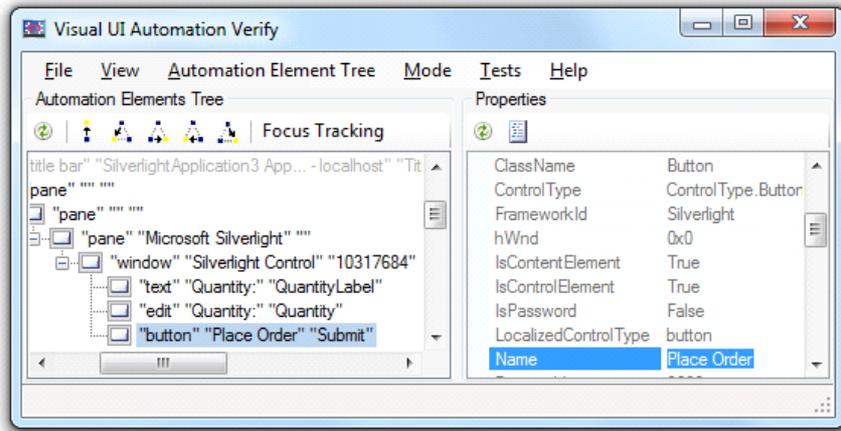
```
// ActionScript 2.0 Example
// Let 'samplemovieclip' is a MovieClip instance placed on the movie's main
timeline
_root.samplemovieclip._accProps = new Object();
_root.samplemovieclip._accProps.name = "지구를 공전하는 달";
```

- 마찬가지로, ActionScript 3.0을 이용하여 프로그램을 개발하는 경우 무비클립에 대한 `accessibilityProperties` 객체의 속성을 설정한다.

```
// ActionScript 3.0 Example
// Let 'samplemovieclip' is a MovieClip instance placed on the movie's main
timeline
samplemovieclip.accessibilityProperties = new AccessibilityProperties();
samplemovieclip.accessibilityProperties.name = "지구를 공전하는 달";
```

<참고> 플래시, 실버라이트 콘텐츠의 대체 텍스트 평가 방법

플래시, 실버라이트 등의 웹 애플리케이션을 구성하고 있는 사용자 인터페이스 자동화 요소들의 대체 텍스트는 UIA Verify와 같은 평가 도구를 이용하여 확인할 수 있다.



이들 평가 도구는 웹 애플리케이션뿐만 아니라 웹 콘텐츠를 구성하는 사용자 인터페이스 요소에 대한 레이블, 단축키 니모닉, 도움말, 해당 요소에 할당된 값 등을 알려준다.

적용 기술 7) 실버라이트 콘텐츠의 대체 텍스트 제공

실버라이트(Silverlight)에서 텍스트 아닌 콘텐츠에 대한 대체 텍스트는 AutomationProperties.Name 속성 또는 AutomationProperties.LabeledBy 속성을 이용하여 제공한다. 콘텐츠 또는 컨트롤에 대한 설명문이나 도움말은 AutomationProperties.HelpText 속성을 이용하여 제공한다.

- AutomationProperties.Name 속성 이용방법

AutomationProperties.Name 속성은 주로 사용자 인터페이스 자동화 클라이언트에서 선택할 수 있는 자동화 속성으로 대체 텍스트를 제공하는 데 사용된다.

다음 예는 상품 이미지(smt_case_12.png)에 대한 대체 텍스트를 제공하는 코드이다.

```
<Image Source="smt_case_12.png"
```

```
AutomationProperties.Name="스마트폰 케이스"/>
```

- AutomationProperties.LabeledBy 속성이용 방법

AutomationProperties.LabeledBy 속성은 다른 사용자 인터페이스 요소에서 정의된 텍스트를 바인딩을 통하여 대체 텍스트로 제공할 경우에 사용하며 그 역할은 AutomationProperties.Name과 동일하다.

다음 예는 텍스트 입력 상자의 대체 텍스트(레이블)를 "실버라이트 설명"이라고 제공하는 코드이다.

```
<TextBlock x:Name="SilverlightLabels">실버라이트 설명</TextBlock>  
<TextBox x:Name="SilverlightOne" width="25"  
AutomationProperties.LabeledBy="{Binding ElementName=SilverlightLabels}"/>
```

적용 기술 8) 그래프 등에 대한 설명문 제공

그래프나 사진과 같이 그 의미하는 바를 간단한 대체 텍스트로 충분한 설명이 불가능할 경우 추가적인 설명을 제공해야 한다. 자세한 설명을 제공하는 텍스트 아닌 콘텐츠의 경우도 대체 텍스트를 제공해야 한다.

설명을 제공하는 방법은 다음과 같은 네 가지 방법이 있다.

- longdesc 속성 제공

텍스트 아닌 콘텐츠(chart.gif)에 대한 자세한 설명을 longdesc 속성을 이용하여 제공할 수 있다. 다음의 예는 longdesc 속성을 이용하여 복잡한 그래프에 대한 설명을 제공하는 웹 페이지(dest_chart.html)로 이동하도록 구현한 경우이다.

```
<p>  
</p>
```

- 설명문 제공

텍스트 아닌 콘텐츠의 내용에 관한 자세한 설명을 해당 콘텐츠의 전후에 제공

함으로써 콘텐츠에 대한 추가적인 설명을 제공하는 것이다. 이 방법은 longdesc 속성의 경우와 같이 설명 페이지로 이동하는 불편함을 해결할 수 있다.

다음의 예는 이미지(chart.gif)에 이어서 이미지에 대한 자세한 설명을 함께 제공하는 구현방법이다.

```
<p></p>
<p>2008년도 중앙기관의 웹 접근성 조사결과에 따르면 ...
...
<p>
```

- 이미지 링크 제공

텍스트 아닌 콘텐츠 또는 그 주변에 설명문으로 이동하는 링크를 제공하는 방법이다. 이 방법은 설명문으로 이동하는 링크의 위치와 모양을 다양하게 제공할 수 있는 장점이 있다.

다음의 예는 이미지에 링크를 걸어 설명문으로 이동하도록 구현한 것이다.

```
<p>
  <a href="chartdesc.html">
    
  </a>
</p>
```

- 링크 버튼 제공

이미지 옆에 링크 버튼을 제공하고, 이 버튼을 클릭하면 같은 페이지의 해당 이미지에 대한 추가적인 설명을 제공하는 영역으로 이동하도록 구현할 수 있다. 이 경우 버튼 레이블은 '그래프 설명문' 또는 '그래프 설명문으로 이동' 등이 가능하다.

여기에 링크버튼 제공 예시 제시 필요!!!

적용 기술 9) 생방송 콘텐츠에 대한 대체 텍스트 제공

실시간 오디오와 실시간 동영상 콘텐츠에 대한 대체 텍스트는 콘텐츠의 내용이 시시각각으로 변화한다. 이 경우, 대체 텍스트를 콘텐츠의 내용 변화에 따라 동적으로 제공하는 것은 매우 어려운 일이다. 따라서 실시간 오디오 또는 동영상 콘텐츠의 대체 텍스트로는 해당 콘텐츠의 용도를 알려주는 것으로도 충분하다. 다음은 그 일례이다.

- 한국도로공사가 제공하는 경부고속도로 서울방향 서초 나들목에 설치된 CCTV 실시간 영상의 링크에 대한 대체 텍스트를 "경부선 서울방향 서초 나들목 실시간 영상"이라고 제공한다.
- 인터넷을 통하여 중계되는 생방송 KBS FM 라디오 콘텐츠로 이동하는 이미지 링크의 대체 텍스트를 "KBS FM 라디오 생방송"이라고 제공한다.

적용 기술 10) CAPTCHA에 대한 대체 텍스트 제공

<참고> CAPTCHA

예를 들어, 다음 그림과 같이 의도적으로 글자 모양을 비틀어 만든 이미지에 대하여 컴퓨터는 그 내용의 정확한 판단이 불가능하나 인간은 그 내용을 파악할 수 있다. 이러한 방법으로 사용자가 컴퓨터인지 인간인지를 구별하기 위해 사용하는 방법을 CAPTCHA(Completely Automated Public Turing test to tell Computers and Humans Apart)라고 한다. CAPTCHA에는 다음 그림과 같은 시각적인 방법이 일반적이거나 오디오로 제공되는 청각적인 방법도 있다.



CAPTCHA는 사용 목적상 그 내용을 대체 텍스트로 제공한다면 CAPTCHA가 의도하는 본래 목적을 달성할 수 없다. 따라서 CAPTCHA에 대한 대체 텍스트는 그 용도를 알려주는 것으로 충분하다. 다음은 CAPTCHA에 대한 대체 텍스트의 예다.

- 텍스트로 제공되는 CAPTCHA에 대한 대체 텍스트로는 "숫자 이미지용 CAPTCHA"가 적절하다.
- 오디오로 제공되는 CAPTCHA에 대한 대체 텍스트로는 "오디오

CAPTCHA" 등이 적절하다.

CAPTCHA를 제공하는 웹 콘텐츠는 반드시 대체 수단을 제공해야 한다. CAPTCHA에 대한 대체 수단을 제공하는 방법은 다음과 같다.

<참고> CAPTCHA의 대체 수단

- 1) 두 가지 이상의 감각, 예를 들어 시각적 CAPTCHA와 청각적 CAPTCHA를 동시에 제공
 - 2) 사용자에게 문제를 텍스트 콘텐츠로 제시하고 문제에 대한 답을 입력하도록 하는 문제 풀이 방식의 CAPTCHA 제공
 - 3) 상담원을 이용하여 CAPTCHA가 제공하는 역할을 대신할 수 있는 서비스 제공
 - 4) 기타 CAPTCHA의 이용을 우회할 수 있는 방법 제공
- ※ 보다 다양한 방법은 W3C WAI에서 제공하는 "Inaccessibility of CAPTCHA (<http://www.w3.org/TR/turingtest/>)"을 참고하기 바람

적용 기술 11) 온라인 시험 등에 대한 대체 텍스트 제공

검사(색맹검사, 청각검사, 시력검사 등) 또는 시험(듣기 평가 예문, 단어 받아쓰기 등)을 위하여 제공되는 텍스트 아닌 콘텐츠는 검사의 결과 또는 시험의 정답을 유추할 수 있는 내용을 대체 텍스트로 제공할 수 없다. 따라서 이들 콘텐츠에 대한 대체 텍스트는 콘텐츠의 간략한 용도를 알려주는 대체 텍스트를 제공해야 한다.

다음은 올바른 사례들이다.

- 색맹 검사용 콘텐츠에 대하여 "색맹 검사용 이미지 1" 등과 같이 대체 텍스트를 제공한다.
- 토플(TOEFL) 듣기 평가용 예제의 오디오 콘텐츠에 대하여 "토플 문항 5번용 오디오" 등과 같이 대체 텍스트를 제공한다.
- 시력검사용 이미지에 대하여 "시력 검사용 이미지" 등과 같이 대체 텍스트를 제공한다.

적용 기술 12) 대체 텍스트를 빈 공간(alt=" ")이나 생략해서 제공해야 하는 경우

이미지에는 반드시 대체 텍스트를 제공해야 한다. 다만, 이미지에 대체 텍스트를 제공함으로써 사용자에게 혼란을 주는 경우 대체 텍스트를 빈 공간(`alt=""`)으로 제공해야 한다. 대체 텍스트를 빈 공간으로 제공하는 경우는 다음과 같다.

- 장식용으로 사용하는 이미지 등

텍스트 설명이 함께 제공되는 이미지 또는 의미가 없는 글머리표 등과 같은 이미지에는 대체 텍스트를 빈 공간으로 제공해야 한다. 이미지에 대한 대체 텍스트의 `alt` 속성을 빈 문자열(`alt=""`)로 설정한다. 여기서 `alt` 속성을 빈 문자열로 제공한다는 것은 `alt` 속성을 제공하되 그 값을 `null`로 제공한다는 의미이다.

다음의 예는 이미지 글머리표의 대체 텍스트를 빈 문자열로 제공하는 경우다.

```
...
<p> 화면 낭독 프로그램</p>
<p> 화면 확대 프로그램</p>
...
```

<참고>

`title` 속성과 `alt` 속성을 모두 제공하면 일부 화면 낭독 프로그램은 두 속성을 모두 읽어주어 혼란을 줄 수 있다. 따라서 대체 텍스트는 반드시 `alt` 속성으로 제공하며, `alt` 속성을 제공하지 않는 태그의 경우에만 `title` 속성으로 제공한다.

- 배경 이미지

배경 이미지는 정보 제공의 목적보다는 시각적인 장식 효과를 내기 위하여 사용되는 이미지이다. 따라서 배경 이미지에 대한 대체 텍스트는 보조기술이 무시하도록 제공되어야 한다.

CSS를 이용하여 배경 이미지를 제공하면 대체 텍스트를 보조기술로 전달하지 않는다. 다음 CSS 코드는 웹 페이지의 배경 이미지를 지정하는 예다.

```
...
<style type="text/css">
  body { background: #ffe url('/images/home-bg.jpg') repeat; }
</style>
```

```
</head>
<body>
...
```

- 플래시 콘텐츠

플래시 콘텐츠를 구성하는 무비클립 또는 버튼 등에 대체 텍스트를 제공함으로써 사용자에게 혼란을 주게 될 경우 대체 텍스트를 제공하지 않아야 한다. 장식용으로 사용되는 무비클립 또는 텍스트 설명이 함께 제공되는 무비클립 등이 여기에 해당된다.

- 어떤 무비클립에 대한 대체 텍스트를 제공하지 않기 위해서는 액세스 가능성(Accessibility) 패널을 열고, 무비클립을 선택한 후에 액세스 가능성 패널의 '객체를 액세스 가능하게 만들기(Make object accessible)' 항목을 체크 해제(unchek)한다. 이 항목을 체크 해제하면 이 무비클립의 이름(N) 항목이나 설명(D) 항목에 대체 텍스트나 설명문을 제공하더라도 보조기술은 이 무비클립에 대한 대체 텍스트를 무시하게 된다.
- 플래시 콘텐츠를 ActionScript 2.0으로 개발하는 경우 무비클립(movieclip)에 대한 _accProps 객체의 silent 속성을 'true' 로 설정해야 한다.

```
// ActionScript 2.0 Example
// 'decorative_mc' is an instance placed on the movie's main timeline
_root.decorative_mc._accProps = new Object();
_root.decorative_mc._accProps.silent = true;
```

- ActionScript 3.0으로 개발하는 경우 무비클립에 대한 accessibilityProperties 객체의 silent 속성을 'true' 로 설정한다.

```
// ActionScript 3.0 Example
// 'decorative_mc' is an instance placed on the movie's main timeline
decorative_mc.accessibilityProperties = new AccessibilityProperties();
decorative_mc.accessibilityProperties.silent = true;
```

- 실버라이트 콘텐츠

실버라이트 콘텐츠를 구성하는 콘텐츠에 대체 텍스트를 제공함으로써 사용자에게 혼란을 주게 될 경우 대체 텍스트를 제공하지 않아야 한다. 장식용으로 사용되

는 배경 이미지, 글머리표 등이 여기에 해당된다.

대체 텍스트를 제공하지 않아야 하는 실버라이트 객체는 Automation Properties 클래스를 이용한 Name 속성 및 LabeledBy 속성 등을 제공하지 않는다.

□ 잘못된 사례 및 개선방법

잘못된 사례 1) 의미나 기능이 있는 텍스트 아닌 콘텐츠에 빈 문자열을 대체 텍스트로 제공

대체 텍스트는 텍스트가 아닌 콘텐츠가 제공하는 기능이나 정보를 동등하게 전달할 수 있도록 제공해야 한다. 예를 들어, 어떤 이미지에 대한 대체 텍스트로 이미지 파일명인 "image.jpg" 등과 같이 제공하거나 "이미지 1", "이미지 2" 등과 같이 이미지에 대한 일련번호를 제공하거나 그냥 "이미지"라고 하는 것은 부적절하다.

다음의 예는 의미를 지닌 세 개의 이미지에 무의미한 대체 텍스트를 제공 (alt="")하여 보조기술이 이미지와 동등한 정보를 인식할 수 없는 경우다.

- Alt= 01. [보다 효율적이고 편리한 맞춤형 One-Stop 서비스 조화](#)
- Alt= 02. [대상별, 생애주기별, 욕구정보를 통한 주민서비스 안내](#)
- Alt= 03. [공공 및 민간시설/단체의 서비스 정보 제공](#)

```
<ol>  
<li></li>  
<li></li>  
<li></li>  
</ol>
```

개선방법

이를 해결하기 위해서는 alt 속성을 이용하여 각각의 이미지에 적절한 대체 텍스트를 제공하며, ol(ordered list) 방식으로 구현해야 한다.

<개선 사례 1>

```
<ol>
<li></li>
<li></li>
<li></li>
</ol>
```

잘못된 사례 2) 복잡한 정보를 하나의 이미지로 표현하고 간단한 대체 텍스트만 제공

다음 예는 기관의 경영이념을 표방하는 정보를 제목 이미지와 내용 이미지로 제작한 경우다. 이 예에서 제목 이미지는 빈 문자열의 대체 텍스트를 제공하고 있으며, 내용 이미지는 "이미지"라는 부적절한 대체 텍스트를 제공하여 그 의미를 제대로 전달하지 못하고 있다.



개선방법

조직도, 경영이념 등과 같이 많은 정보를 하나의 이미지로 구성하여 정보를 제공하는 콘텐츠는 설명문을 제공해야 한다. 또한 이미지 자체에 대한 적절한 대체 텍스트도 제공해야 한다.

```

```

더 바람직한 방법은 내용 이미지를 '단계별 혁신목표 성과' 부분과 '· · · 추진 전략 로드맵'의 두 가지 이미지로 구분하고 각각의 이미지에 대하여 alt 속성과 longdesc 속성을 제공하는 것이다.

잘못된 사례 3) 주기적으로 변하는 콘텐츠의 대체 텍스트를 변경하지 않는 경우

텍스트 아닌 콘텐츠가 갱신되는 경우 대체 텍스트도 동시에 갱신되어야 한다.

- 매출 자료를 토대로 월 매출실적을 제공하도록 웹 애플리케이션을 제공하는 경우 달이 변경되었음에도 불구하고 월 매출실적이 변경되지 않는 대체 텍스트는 잘못된 구현 예다.
- 계절에 따라 소개 페이지(intro page)의 사진이 주기적으로 변경됨에도 불구하고 사진에 대한 대체 텍스트가 동시에 변경되지 않는다면 잘못 구현된 예다.
- 몇 개의 상품 소개 이미지가 2초에 한 번씩 주기적으로 변화하도록 플래시로 구현한 콘텐츠에서 이미지가 바뀔 때마다 대체 텍스트가 갱신되어야 함에도 불구하고 플래시 콘텐츠 전체에 대한 대체 텍스트를 "상품보기"로 제공한 것은 잘못 구현한 경우다.

개선방법

위에 열거한 잘못을 개선하기 위해서는 플래시 콘텐츠 전체에 대한 대체 텍스트를 제공하지 않고, 플래시 콘텐츠를 구성하고 있는 모든 이미지를 무비클립으로 변환하며, 각 무비클립별로 액세스 가능성 패널의 '객체를 액세스 가능하게 만들기(Make object accessible)' 항목을 체크(check)하고, 이름(N) 항목에 각 이미지별로 적절한 대체 텍스트를 제공하면 된다.

잘못된 사례 4) 의미있는 정보를 배경 이미지로 제공

의미있는 이미지를 CSS만으로 표현함으로써 이미지가 표현하려는 중요한 의미나 정보를 사용자에게 제공할 수 없도록 구현하지 않아야 한다.

- '연이율 5.5%!'라는 텍스트 이미지(rate.png)를 배경 이미지로 구현하고 은행이 제공하는 금리를 텍스트로 제공하지 않아 정작 중요한 이자율을 보조 기술 사용자가 인식할 수 없다.

CSS 코드:

```
p#anual_interest {  
  padding-left: 200px;  
  background: transparent url(./rate.png) no-repeat top left;  
}
```

HTML 코드:

```
<p id="anual_interest">  
  당 은행은 가장 높은 금리를 보장합니다!  
</p>
```

개선방법

이 문제를 해결하기 위해서는 이미지를 배경 이미지가 아닌 `img` 태그를 사용하여 구현하고 `alt` 속성을 제공한다.

```
<p>  
  당 은행은 가장 높은 금리를 보장합니다!  
</p>
```

- 다음 사례는 온라인 음반 매장에서 CD 타이틀 별로 '신규입고', '재고 없음', '구입가능' 등으로 재고를 구분하여 목록을 보여주는 콘텐츠로 재고 여부를 배경 이미지로 표현한 경우다. 이 경우 타이틀의 재고 여부를 보조기술 사용자가 인식할 수 없다.

CSS 코드:

```
ul#cdlist li {  
  padding-left: 20px;  
}  
ul#cdlist li.new {  
  background: transparent url(new.png) no-repeat top left;  
}  
ul#cdlist li.instock {  
  background: transparent url(instock.png) no-repeat top left;  
}  
ul#cdlist li.outstock {  
  background: transparent url(outstock.png) no-repeat top left;  
}
```

HTML 코드:

```
<ul id="booklist">
  <li class="new">CD Title #1</li>
  <li class="instock">CD Title #2</li>
  ...
  <li class="outstock">CD Title #n</li>
</ul>
```

개선방법

이 문제를 해결하기 위해서는 '신규입고', '구입가능', '재고없음'을 의미하는 세 가지 이미지(new.png, instock.png, outstock.png)를 img 태그로 표현하고 alt 속성을 이용하여 대체 텍스트를 제공한다.

```
<ul id="booklist">
  <li> CD Title #1</li>
  <li> CD Title #2</li>
  ...
  <li> CD Title #n</li>
</ul>
```

잘못된 사례 5) 빈칸 이미지의 대체 텍스트가 콘텐츠의 인식을 방해하는 경우

보조기술 사용자에게 혼란을 줄 수 있는 대체 텍스트를 제공하는 것은 잘못된 경우다. 다음 사례는 콘텐츠에 빈칸 이미지(spacer.gif)를 삽입하여 간격을 조절하고 있다. 화면 낭독 프로그램은 이 콘텐츠를 "나무 종류: '빈칸' 상수리나무"라고 읽어주므로 빈칸 이미지(spacer.gif)에 대한 대체 텍스트가 도리어 콘텐츠의 인식을 방해하게 된다.

```
<div>나무 종류: 상수리
나무</div>
```

개선방법

이를 방지하기 위하여 빈칸 이미지에 대한 대체 텍스트를 빈 문자열(alt="")로 설정한다.

```
<div>나무 종류: 상수리
나무</div>
```

잘못된 사례 6) 지도와 같은 텍스트 아닌 콘텐츠에 대한 설명이 불충분한 경우

텍스트 아닌 콘텐츠에 대한 대체 텍스트 또는 설명(longdesc)은 이 콘텐츠가 가진 기능이나 용도를 잘 이해할 수 있도록 설명해야 한다.

예를 들어, 다음 그림과 같이 지하철 역 주변의 주요 건물을 소개하는 지도에서 설명문이 건물의 이름 또는 상호만을 제공하는 것은 충분하지 않다.

 <p><map.gif></p>	<pre> <p> 등촌역 근처 주요 건물 </p> 강서생복요리전문, 미래에셋, 00주유소(로드뷰 제공) ... 그린월드호텔, 북두칠성.</pre>
---	---

개선방법

지도의 용도가 지하철 출구와 건물들과의 상관관계를 나타내고 있으므로 이 지도에 대한 설명(longdesc)은 지하철역의 출구 번호별로 각 건물까지의 도달 거리, 접근 방법 등에 대한 내용을 포함해야 한다.

```

<p> 등촌역 주변 주요 건물 </p>
1번 출구: 강서생복요리전문, 우측 30m 전방;
2번 출구: 미래에셋, 좌측 10m 전방; 북두칠성, 좌측 100m 전방; 00주유소(로드뷰 제공), 150m 전방
...
8번 출구: 그린월드호텔, 정면 50m 전방.
```

잘못된 사례 7) 중요한 정보를 색으로만 제공하는 경우

대체 텍스트에는 색, 위치, 모양 등을 묘사하는 단어를 사용하지 않고 그 차이

를 직접 문장으로 표시해야 한다.

다음 예는 영업 직원별 연간 매출액을 표시한 막대 그래프이다. 이 그래프에 대해 다음 대체 텍스트와 같이 문장 중에 색을 표시하는 경우 색을 인식할 수 없는 사용자에게 충분한 정보를 제공하지 못한다. 즉, 다음 그림에 보인 바와 같이 “... 그래프에서 빨간색 막대는 ...”의 문장에서 시각장애인은 그림 중의 빨간색 막대와 관련한 정보를 알 수 없다.



``

`<char.png>`

개선방법

대체 텍스트가 색으로 인한 차이를 구체적으로 설명하도록 수정할 필요가 있다.



``

`<char.png>`

잘못된 사례 8) alt 속성이 아닌 title 속성을 이용하여 대체 텍스트 제공

이미지에 대한 대체 텍스트를 alt 속성으로 반드시 제공해야 한다. 다음 예는 alt 속성 대신 title 속성을 이용하여 대체 텍스트를 제공하고 있다.



```

```

개선방법

이미지에 대한 대체 텍스트는 반드시 alt 속성을 이용하여 제공해야 한다.

```

```

잘못된 사례 9) QR 코드에서 URL을 대체 텍스트로 알려주지 않는 경우

웹 페이지의 일부 콘텐츠에 편리하게 접속할 수 있도록 하기 위하여 제공되는 QR 코드, 바코드 등의 이미지에는 QR코드나 바코드의 용도를 대체 텍스트로 제공해야 한다. 예를 들어, 어떤 웹 사이트로 이동하기 위한 QR코드의 대체 텍스트는 접속할 수 있는 웹 사이트 주소를 대체 텍스트로 제공해야 한다.

다음 예는 QR 코드 이미지에 대한 대체 텍스트를 제공하나 접속할 주소를 알려주지 않은 경우다.



```
<QR_code.jpg>
```

```
<h2>DDOS 공격 우수 대응 사례</h2>
<p>DDOS 공격 우수 대응 사례에 대하여
...
<p>

```

개선방법

QR 코드 이미지에 저장된 주소 정보 등을 대체 텍스트 또는 설명으로 제공하거나 링크를 제공하여 QR 코드가 알려주는 웹 페이지로 직접 이동이 가능하도록 한다.

```
<h2>DDOS 공격 우수 대응 사례</h2>
<p>DDOS 공격 우수 대응 사례에 대하여 ... <p>

```

검사항목 1.2.1 (자막 제공) 멀티미디어 콘텐츠에는 자막, 원고 또는 수화를 제공해야 한다.

□ 개요

멀티미디어 콘텐츠를 동등하게 인식할 수 있도록 제작하기 위해서는 자막, 원고 또는 수화를 제공해야 한다. 대체 수단에서 가장 중요한 요소는 멀티미디어 콘텐츠와 동등한 내용을 제공하는 것이다.

가장 바람직한 방법은 닫힌 자막(Closed caption)을 오디오와 동기화시켜(Synchronized) 제공하는 것이다. 대사 없이 영상(Video)만 제공하는 경우 화면해설(텍스트, 오디오, 원고)을 제공하고, 음성만 제공하는 경우 자막, 원고 또는 수화를 제공해야 한다.

생방송 콘텐츠의 경우 자막 또는 수화의 제공을 이 검사항목의 예외로 인정한다. 그러나 생방송 콘텐츠의 경우에도 자막 또는 수화를 제공하는 것이 바람직하다.

생방송 콘텐츠의 경우에도 검사항목 1.1.1, 적용기술 9)에 의거한 콘텐츠에 대한 대체 텍스트를 반드시 제공해야 한다.

□ 적용 기술

적용 기술 13) 열린 자막 제공

청각장애인은 시청각 콘텐츠에 포함된 대화를 청취할 수 없으므로 이를 시각 정보로 제공해야 한다. 열린 자막(Open caption)이란 동영상의 오디오 트랙에 포함된 대화 및 중요한 음향 정보를 텍스트 자막으로 표시하도록 비디오 트랙에 포함시켜 만드는 것을 말한다. 따라서 열린 자막을 제공하는 동영상은 자막이 항상 화면에 표시되므로 특별한 동영상 플레이어 사용할 필요가 없다.

적용 기술 14) 닫힌 자막 제공

닫힌 자막은 사용자가 필요에 따라 자막을 화면에 표시하지 않거나 표시하도록 제작하는 방법이다. 닫힌 자막의 내용은 동영상의 대화와 중요한 음향 정보를 포함해야 한다.

닫힌 자막은 동영상과 동기화되어야 하며, 이를 위해서는 닫힌 자막을 제공할 수 있는 동영상 플레이어를 사용해야 한다.

대화나 음향 효과가 제공되지 않는 동영상은 '이 동영상은 음성을 제공하지 않음'이라는 메시지를 닫힌 자막 방식을 이용하여 수시로 제공할 필요가 있다.

<참고> 자막 제공 방법

SMIL 1.0:

<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/SM11>

SMIL 2.0:

<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/SM12>

플래시:

<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/FLASH9>

실버라이트: <http://www.silverlight.net/learn/quickstarts/accessibility/#MediaAndCaptioning>

적용 기술 15) 대본 또는 원고 제공

오디오를 제공하는 콘텐츠는 포함하고 있는 대화 또는 음향 정보를 사용자가 접근할 수 있도록 텍스트 대본을 제공한다. 대본은 오디오 내용과 일치해야 한다.

대본의 양이 많아 상하 스크롤이 필요한 경우 동영상 또는 오디오와 동기화되어 스크롤 되는 것이 바람직하다.

대본을 제공하는 방법은 다음 중 한 가지 방법을 적용한다.

<참고> 대본 제공 방법

- 동영상 콘텐츠 바로 옆에 대본을 텍스트로 제공
- 동영상 콘텐츠 또는 동영상 콘텐츠 주위에 대본으로 이동하는 링크 제공

적용 기술 16) 수화 제공

빠르게 변화하는 자막을 인식하기 어려운 청각장애인을 위하여 동영상의 자막을 수화로 통역하는 영상을 함께 제공한다. 함께 제공되는 수화 동영상과 원래의 동영상은 서로 다른 영역에 표시될 수 있다.

<참고> 수화 동영상의 제공 방법

SMIL 1.0:

<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/SM13>

SMIL 2.0:

<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/SM14>

수화 동영상을 원래의 동영상과 같은 비디오 트랙에 포함시켜 제공하는 경우 동영상의 해상도가 낮을 경우 수화를 알아보기 어려운 문제가 있으므로 동영상의 해상도를 적절히 유지해야 한다.

수화와 함께 자막을 제공하는 경우 자막으로 인하여 수화가 가리지 않도록 닫힌 자막으로 구현하여 사용자가 자막을 화면에서 삭제할 수 있어야 한다.

적용 기술 17) 음성이 없는 동영상의 대체 수단 제공

대화나 독백, 설명 등과 같은 음향 정보가 없는 동영상은 애니메이션, 동영상을 통한 텍스트나 그래픽 정보 제공, 사람이나 동물들의 행동을 묘사하는 경우 많이 사용된다. 따라서 이러한 동영상은 콘텐츠가 이야기하고자 하는 바를 시각장애인이 인지하기 위해서는 동영상의 내용을 충실히 묘사할 수 있는 대체 수단을 제공해야 한다.

1) 대체 텍스트 또는 설명으로 제공

동영상의 내용을 텍스트 문서로 작성하고 이를 대체 텍스트 또는 설명문으로 제공한다. 여기에서 중요한 점은 대체 텍스트 또는 설명만으로도 동영상이 의도하는 바를 사용자가 충분히 이해할 수 있도록 정보를 제공해야 한다는 점이다.

대체 텍스트 또는 설명을 제공하는 방법은 다음과 같다.

<참고> 대체 텍스트 또는 설명 제공 방법

- ◆ alt 속성으로 제공
- ◆ longdesc 속성으로 제공
- ◆ 동영상 콘텐츠 주위에 설명문 제공
- ◆ 동영상 콘텐츠 또는 동영상 콘텐츠 주위에 설명문으로 이동하는 링크 제공

2) 오디오 해설 제공

동영상의 내용을 설명해 주는 해설을 오디오 파일로 제공하는 방법이다. 이 방법은 첫 번째 방법에서 작성한 텍스트 문서를 읽어주는 경우로 간주할 수 있다.

예를 들어, 종이접기 동영상에 대한 오디오 해설은 "첫째, 정사각형의 종이를 준비한다. 두 번째, 종이를 반으로 접어 이등변 삼각형을 만든다. 세 번째는 이등변 삼각형의 ... ' 와 같이 단계별로 접는 방법을 설명해야 한다.

오디오 해설은 동영상의 오디오 트랙을 이용하여 제공한다. 만일 동영상 장면에 대한 오디오 해설이 장면의 지속시간보다 길어 오디오 해설이 종료되기 전에 동영상의 장면이 바뀌는 경우 동영상의 해당 장면이 바뀌기 직전에 동영상을 일시 정지시키고 오디오 해설이 완료된 다음 동영상이 계속 이어지도록 하는 것이 바람직하다.

□ 잘못된 사례 및 개선방법

잘못된 사례 10) 자막, 원고, 수화 중 어느 한 가지도 제공하지 않는 경우

- 1) 대화나 음향정보를 제공하는 다음 멀티미디어 콘텐츠는 청각장애인이 접근할 수 없다.



개선방법

이 콘텐츠에는 청각장애인이 인식할 수 있도록 자막, 원고 또는 수화 중 하나를 제공해야 한다.

- 2) 다음 예와 같이 우리나라 자연의 사계절 아름다움을 표현한 동영상에서 잔잔한 배경음악을 함께 제공하는 콘텐츠는 시각장애인이 인식할 수 없다.



개선방법

시각장애인이 접근할 수 있도록 영상의 변화를 화면 해설 방식으로 제공할 필요가 있다. 만일 화면에 텍스트를 제공한다면 화면 해설과 함께 텍스트를 읽어 주어야 한다.

잘못된 사례 11) 동영상의 내용을 요약하여 제공하는 자막 또는 원고의 경우

자막, 대본 등을 제공하는 동영상 콘텐츠의 경우 자막 또는 대본이 동영상의 모든 내용을 포함해야 한다. 다음과 같은 동영상 콘텐츠는 잘못된 사례에 속한다.

- 동영상의 대화를 요약하여 제공
- 내용 이해에 필수적인 음향정보 누락
- 동영상의 일부 대화에 대한 대본 생략

다음은 요약 정보를 제공하는 경우이므로 잘못된 사례이다.



개선방법

멀티미디어 콘텐츠의 자막, 대본 또는 수화는 요약 정보를 제공하지 않고 동영상의 대화, 중요한 음향 정보를 충실히 제공해야 한다.

잘못된 사례 12) 대체 수단에 대한 또 다른 대체 수단임을 알려주지 않는 경우

사용자는 멀티미디어 콘텐츠가 제공되는 경우 항상 대본 등의 대체 수단을 함께 제공할 것으로 예측한다. 그런데 멀티미디어 콘텐츠가 텍스트 콘텐츠 또는 여타의 콘텐츠에 대한 대체 수단으로 제공되는 경우 대본을 중복해서 제공할 필요가 없다. 이 경우, 멀티미디어 콘텐츠가 대체 수단임을 명시해야 한다.

예를 들어, 종이 접기 절차를 텍스트로 제공하는 웹 페이지의 한 부분에 종이 접기 절차를 시연하는 동영상 또는 오디오 콘텐츠를 제공하는 경우 이 콘텐츠에 대본이나 긴 설명문 등을 제공하기보다는 "종이 접기에 대한 대체 수단"이라는 대체 텍스트를 제공하는 것으로 충분하다.

멀티미디어 콘텐츠 주변에 종이 접기 본문으로 이동하는 링크를 함께 제공하는 것도 바람직하다.

잘못된 사례 13) 열린 자막이 수화를 가리는 동영상

수화와 자막을 제공하는 동영상 콘텐츠에서 자막으로 인하여 수화를 가리는 콘텐츠는 수화를 인식할 수 없게 하므로 바람직하지 않다.

개선방법

이를 개선하기 위해서는 수화와 자막을 동시에 제공할 때 다음과 같이 대체 수단을 제공해야 한다.

- 수화와 자막 중에서 최소한 한 가지 대체 수단을 닫힌 방법으로 제공하여 사용자가 대체 수단을 선택할 수 있도록 함
- 수화와 자막을 동시에 제공하는 경우 수화를 전면에 나타나게 함

잘못된 사례 14) 동영상의 해상도가 낮아 수화를 볼 수 없는 경우

웹 페이지를 통하여 제공하는 동영상 콘텐츠에 수화를 제공하는 경우 동영상을 확대하지 않으면 수화를 인지하기 어렵다.

- 1) 동영상의 해상도가 낮으면 화면을 확대하더라도 수화를 인식하기 어렵다.

개선방법

이를 개선하기 위해서는 수화를 제공하는 동영상은 확대를 하더라도 수화를 인식할 수 있도록 충분한 해상도의 동영상을 제공해야 한다.

- 2) 수화통역사와 배경 간의 명도 대비가 작은 콘텐츠는 청각장애인이 인식하기 어렵다.

개선방법

수화통역사의 손과 얼굴과 배경간의 명도 대비는 3:1 이상이어야 한다. 수화 동영상의 명도 대비에 관한 검사는 검사항목 1.3.3(텍스트 콘텐츠의 명도 대비)을 만족해야 한다.

검사항목 1.3.1 (색에 무관한 콘텐츠 인식) 콘텐츠는 색에 관계없이 인식될 수 있어야 한다.

□ 개요

콘텐츠에서 제공하는 모든 정보는 색을 배제하여도 특정한 색을 구별할 수 없는 사용자나 흑백 디스플레이 사용자, 흑백 인쇄물을 보는 사용자가 해당 콘텐츠를 인식할 수 있도록 제공해야 한다.

□ 적용 기술

적용 기술 18) 색을 보완하는 텍스트 제공

색을 이용하여 서식의 용도를 나타내는 경우와 같이 정보를 제공하는 수단으로 색을 사용한다면 보완 방법을 함께 제공하여 색의 의미를 전달할 수 있어야 한다. 가장 쉬운 방법은 색의 의미를 텍스트로 제공하는 것이다.

- 다음 예와 같이 예약이 마감되었음을 '완'이라는 글자를 병기하고 색으로 표시한다.



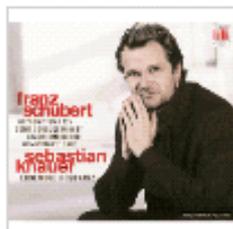
- 입력서식에서 필수 입력 항목임을 알려주기 위하여 해당 항목의 레이블에 '*' 표시를 추가하고 색으로 강조하는 것도 좋은 방법이다.

```
<label for="name" class="required"> * 이름: </label>
<input id="name" type="text" size="25" value=""/>
<style type="text/css">
  .required {
    color:red;
  }
</style>
```

- 입력서식에서 필수 입력 항목임을 알려주기 위하여 해당 항목의 레이블에 '(필수)' 표시를 추가하고 색으로 강조할 수도 있다.

```
<label for="name" class="required">이름(필수): </label>
<input id="name" type="text" size="25" value=""/>
<style type="text/css">
  .required {
    color:red;
  }
</style>
```

- 이미지와 대체 텍스트를 사용하는 것도 가능하다. 예를 들어, 온라인 쇼핑몰에서 새로 입고된 상품임을 알려주기 위하여 상품 주위에 신규 상품임을 알려주는 이미지  를 함께 제공한다. 물론 이 이미지에 대해서는 “new” 라는 대체 텍스트를 제공해야 한다.



슈베르트: 즉흥곡, 잊혀진 소나타 & 아다지오
Knauer / Schubert: Impromptus D.935, S
and Rondo Concertante D487 

COMPACT  Stereo

- 굳건한 표현과 탁월한 테크닉, 섬세한 감수성을 우저의 슈베르트 소나타집은 어떤 탁월한 연주와! 활력으로 가득차 있으며, 외르크 데무스와 롤란트 타 D916b의 세계 초연 녹음을 담고 있다는 점에서

적용 기술 19) 색을 보완하기 위하여 글자모양을 이용하는 방법

색을 구분할 수 없거나 약시 사용자를 위하여 텍스트의 표시 방법을 달리하는 방법을 사용할 수 있다. 글자 모양을 이용한 방법으로는 다른 글꼴의 사용, 밑줄 (underline) 표시, 굵은 글씨체 또는 이탤릭체, 글자 크기의 변경, 명도 대비 차이를 이용한 방법 등이 있다.

대표적인 방법은 다음과 같다.

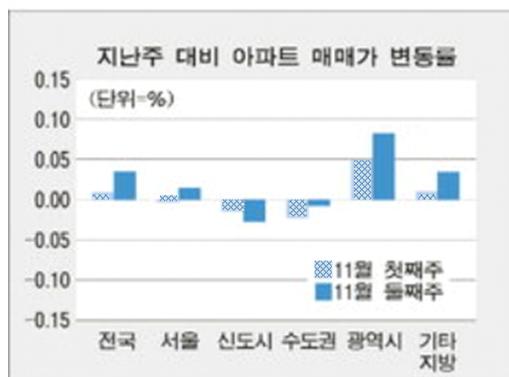
- 링크 텍스트를 색을 이용할 뿐만 아니라 밑줄 표시를 하거나 이탤릭체로 표시
- 온라인 쇼핑몰에서 새로 입고된 상품임을 알려주기 위하여 색과 함께 상품 명을 크고 굵은 글씨체로 표시
- 텍스트 콘텐츠의 일부를 링크 텍스트로 이용하는 경우 링크 텍스트의 색을 밝은 파란색(#3366CC)으로 표시하고 나머지 텍스트를 검정색(#000000)으로 표시하면 색을 구분하지 못하는 경우도 링크 텍스트와 나머지 텍스트 간의 명도 대비를 이용하여 링크임을 알려줄 수 있음

적용 기술 20) 색을 보완하기 위하여 무늬 또는 모양을 이용하는 방법

색을 구분할 수 없거나 약시 사용자를 위하여 무늬 또는 모양을 이용하여 콘텐츠를 표시한다.

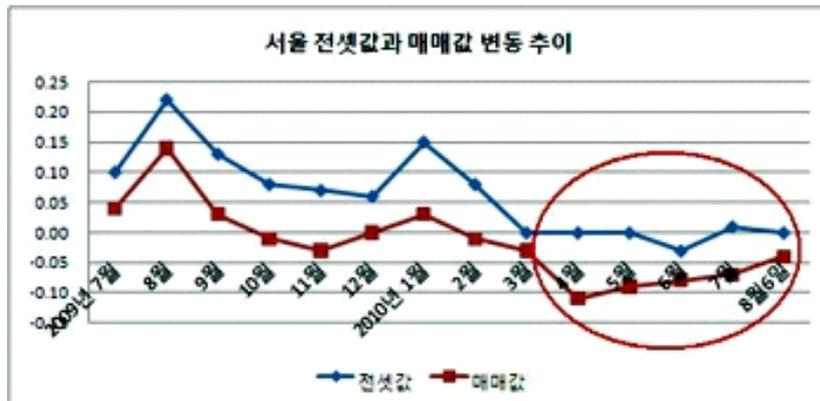
1) 무늬 사용

지역별 아파트 매매가 변동률을 막대그래프로 표시할 때 주별 막대를 무늬를 이용하여 표시할 경우 범례도 무늬를 이용하여 표시한다.



2) 형상 이용

어떤 지역의 전세 값과 매매 값 변동추이를 꺾은선 그래프로 나타낼 때 전세 값과 매매 값의 월별 변동 추이를 각각 다른 모양(다이아몬드, 네모)으로 표시하여 그래프를 구분할 수 있도록 한다.



3) 점선과 실선 사용

어떤 프로세스를 완성하기 위하여 반복적인 단계를 알려주는 플로차트에서 조건에 따른 처리 과정을 빨간 점선의 화살표와 파란 실선의 화살표로 구분하여 표시한다.

점선과 실선을 이용한 사례 예시 제시할 것!!!

□ 잘못된 사례 및 개선방법

잘못된 사례 15) CSS로 인하여 링크임이 구분되지 않는 경우

다음 예와 같이 CSS를 이용하여 링크 텍스트의 밑줄을 사라지게 하는 경우는 잘못된 것이다.

```
<head>
<style type="text/css">
p a:link {text-decoration: none}
p a:visited {text-decoration: none}
p a:active {text-decoration: none}
p a:hover {text-decoration: underline; color: red;}
</style>
</head>
```

```

</style>
</head>
<body>
...
<p>웹 접근성에서 <a href="./criteria.html/#criteria_1_3_1">색에 관한 검사항목</a>에 관한 잘못된 사례 및 개선방법은 다음과 같다. </p>
...
</body>

```

개선방법

이를 해결하려면 위 CSS 코드에서 link의 표시 방법을 다음과 같이 수정해야 한다. 이때 링크 텍스트는 파란색으로 표시되며, 밑줄이 표시된다.

```
p a:link {text-decoration: underline; color: blue;}
```

잘못된 사례 16) 색으로만 필수 입력 항목을 표시한 경우

사용자 서식의 필수항목을 색과 함께 시각적인 추가 수단을 제공해야 한다. 예를 들어, 다음의 회원 가입 서식에서 필수 항목을 꽃분홍색으로 표시하고 필수항목임을 알려주는 것은 잘못된 경우다.

개인회원 가입

회원가입에 동의해 주셔서 감사합니다. **꽃분홍색은 필수 입력 항목입니다.**

회원가입에 동의해 주셔서 감사합니다. **꽃분홍색은 필수 입력 항목입니다.**

> 실명이나 기타 허위 가입된 아이디 또는 주민번호를 사용한 회원은 법적인 보호를 받으실 수 없으며 서비스 이용에 제한을 받을 수 있습니다.

> 한번 부여받은 ID는 변경할 수 없습니다.

회원 ID **중복확인** 숫자/영문으로 조합된 4~10자만 허용

비밀번호 숫자/영문으로 조합된 4~10자만 허용(특수문자불가)

비밀번호확인

회원실명

회원주민등록번호 -

연락가능한 이메일 @

회원가입 메뉴

- 로그인
- 이용약관
- 회원정보수정
- 회원탈퇴
- ID, 비밀번호찾기
- 메일링서비스 변경

개선방법

색을 배제하고서라도 구분할 수 있도록, 다음 그림과 같이 '*' 등을 색과 함께 제공하여 색 이외의 방법으로도 구분할 수 있도록 제공해야 한다.

개인회원 가입
회원가입에 동의해 주셔서 감사합니다. **꽃분홍색은 필수 입력 항목입니다.**

회원가입 메뉴

- 로그인
- 이동약관
- 회원정보수정
- 회원탈퇴
- ID, 비밀번호찾기
- 비밀번호변경

▶ 실명이나 기타 허위 가입된 아이디 또는 주민번호를 사용한 회원은 법적인 보호를 받을 수 없으며 서비스 이용에 제한을 받을 수 있습니다.

▶ 한번 부여받은 ID는 변경할 수 없습니다.

* 회원 ID **중복확인** 숫자/영문으로 조합된 4~10자만 허용

* 비밀번호 숫자/영문으로 조합된 4~10자만 허용(특수문자불가)

* 비밀번호확인

* 회원성명

* 회원주민등록번호 -

* 연락가능한 이메일 @

잘못된 사례 17) 오류 메시지를 색으로만 구분하도록 한 경우

오류가 발생한 서식 부위를 색으로만 표시하면 시각장애인이나 색을 구분할 수 없는 사용자는 오류가 발생한 서식의 위치를 인식할 수 없다.

예를 들어, 온라인 서식을 작성하여 제출할 때 어떤 항목을 채우지 않아 오류가 발생하였을 경우 오류가 발생한 서식의 레이블을 빨간색으로만 표시하는 것은 잘못된 것이다.

개선방법

이를 개선하기 위해서는 오류가 발생한 서식의 레이블을 빨간색 굵은 글꼴로 표현하거나 밑줄을 이용하여 표시하는 등 색 이외의 추가적인 인지 수단을 제공해야 한다. 이외에도 다음 그림과 같이 오류 메시지에 테두리를 제공하는 것도 좋은 방법이다.

성명(필수)을 입력하십시오. 이메일 주소는 선택사항입니다.

성명: * 이 영역은 필수입니다.

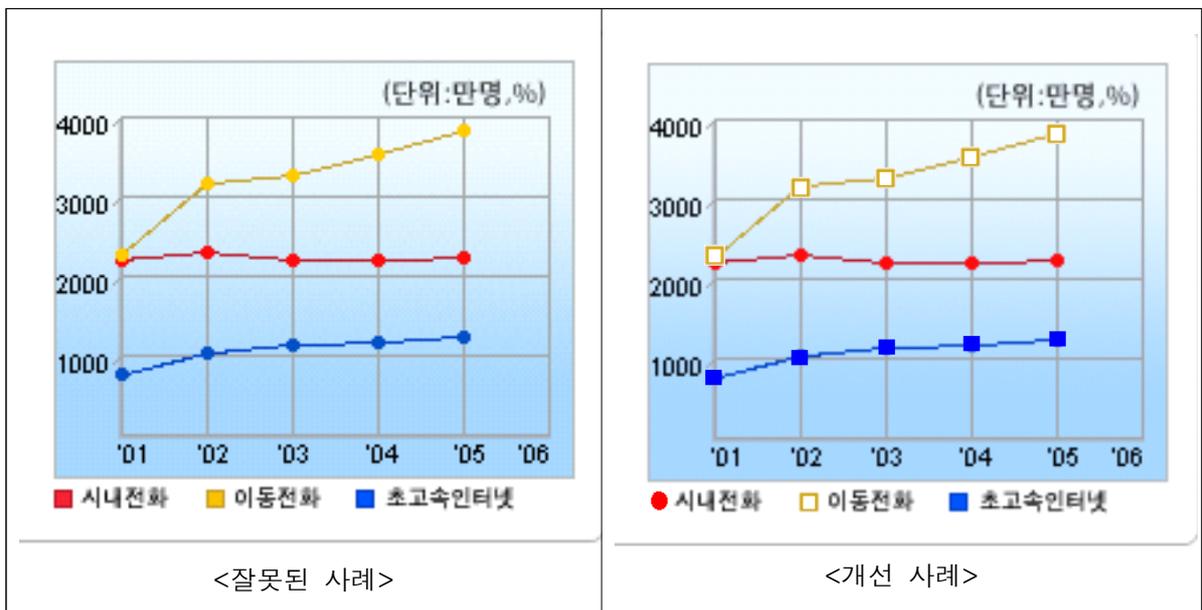
이메일:

잘못된 사례 18) 그래프를 색으로만 인식하도록 하는 경우

년도에 따른 통신서비스별 가입자 수 추이를 나타낸 왼쪽의 잘못된 사례에 해당하는 색을 이용하여 통신 서비스를 구분하고 있어 색을 구분할 수 없는 사용자는 인식할 수 없다.

개선방법

이를 개선하려면 색 이외에 추가적인 인지수단을 제공해야 한다. 다음 그림에서 오른쪽의 개선 사례는 통신서비스별로 가입자 수의 변화와 범례를 다른 형상으로 표시한 것이다.



잘못된 사례 19) 일정 구분을 색으로만 구분하도록 한 경우

다음 그림의 왼쪽에 보인 잘못된 사례는 월별 예약현황을 분홍색과 하늘색으로 표시한 경우다. 색을 구분할 수 없는 사용자는 예약 가능 또는 예약 마감 여부를 알 수 없다.

개선방법

이를 개선하려면 색을 배제하고서라도 내용을 구분할 수 있도록 무늬를 이용하여 구분할 수 있도록 한다. 오른쪽 그림의 개선 사례는 예약 가능과 예약 마감을 무늬를 이용하여 구분한 경우로, 색을 구분하지 못하는 사용자도 예약현황을 구분할 수 있다.



검사항목 1.3.2 (명확한 지시사항 제공) 지시사항은 모양, 크기, 위치, 방향, 색, 소리 등에 관계없이 인식될 수 있어야 한다.

□ 개요

본 검사항목은 특정 요소를 가리키거나 지시사항을 전달하는 콘텐츠에 한정해 적용하는 것으로, 시각이나 청각 등과 같은 특정 감각에만 의존하여 제공해서는 안 된다는 것이다. 즉, 다른 감각을 통해서도 지시사항을 인식하는 데 문제가 없도록 제공해야 한다. 본 검사항목은 화면에 표시되는 텍스트 콘텐츠와 대체 텍스트가 제공된 텍스트 아닌 콘텐츠의 경우 보조기술을 통해 다른 감각으로의 전환이 가능하기 때문에 화면에 표시되는 텍스트를 음성으로 다시 출력해 주는 것을 의미하지는 않는다.

□ 적용 기술

적용 기술 21) 시각 정보를 이용한 지시문의 보완

시각적인 지시나 위치를 사용하면 시각장애인이나 화면의 일부를 확대하여 사용하는 약시자는 지시하는 콘텐츠를 찾아내기 어렵다. 또한 페이지 레이아웃은 폰트나 브라우저의 크기 변경 등으로 달라질 수 있어 콘텐츠의 상대적인 위치도 변경될 수 있으므로 주의해야 한다.

색, 크기, 모양 또는 위치를 이용하여 어떤 컨트롤이나 콘텐츠를 지시하는 경우 시각장애인이 인식할 수 있도록 대체 수단을 제공해야 한다. 다음 경우는 그 일례이다.

- 웹 페이지에 둥근 모양의 빨간색 버튼과 초록색 버튼을 제공하고 각 버튼의 레이블을 각각 '확인' 과 '취소'로 구성한 콘텐츠에서 "작성한 서식을 제출하려면 빨간색 확인 버튼을 누르시오."라는 지시문이 지정하는 버튼을 시각장애인도 사용할 수 있다.
- 웹 페이지에 두 개의 크기가 다른 둥근 모양 버튼을 제공하고 각 버튼의 레이블을 '확인' 과 '취소'로 제공하는 콘텐츠에서 "작성한 서식을 제출하려면

큰 확인 버튼을 누르시오." 라는 지시문이 지정하는 버튼을 시각장애인도 사용할 수 있다.

- 웹 페이지에 둥근 모양, 네모 모양 및 오각형 모양의 버튼을 제공할 경우 버튼별로 텍스트 레이블을 제공하면 모양과 텍스트 레이블을 이용하여 지시하는 버튼을 인식할 수 있다. 예를 들어, "작성한 서식을 제출하려면 둥근 색의 확인 버튼을 누르시오." 라고 하면 시각장애인도 지시하는 버튼을 확인하여 사용할 수 있다.
- 웹 페이지에서 방향을 이용하여 콘텐츠를 지시하는 경우도 방향과 함께 콘텐츠의 레이블을 함께 제시하면 지시하는 콘텐츠를 쉽게 인식할 수 있다. 예를 들어, "오른쪽 상단의 '요약보기'를 클릭하십시오." 라는 지시문이 지시하는 콘텐츠를 시각장애인도 사용할 수 있다. 더 좋은 방법은 그냥 "'요약보기'를 클릭하십시오." 라고 하는 것이다.

적용 기술 22) 청각 정보를 이용한 지시문의 보완

청각 정보를 이용하여 지시문을 제공하는 웹 콘텐츠는 청각 정보를 인식할 수 없는 사용자를 위한 대체 수단을 제공해야 한다.

예를 들어, 온라인 시험을 제공하는 웹 페이지에서 제한시간이 종료되기 1분전임을 알려주기 위한 방법으로 '시험 종료 1분전'이라는 음성을 들려주는 콘텐츠는 '시험 종료 1분 전'이라는 메시지를 화면에 출력하도록 하면 청각장애인도 그 내용을 보고 상황을 인식할 수 있게 된다.

더 좋은 방법은 '시험 종료 1분 전'이라는 메시지를 화면에 출력함과 동시에 화면을 몇 차례 깜빡이도록 하는 '소리 탐지(Sound Alert)' 기능을 활성화 시키면 청각장애인도 경고음을 시각적으로 인지할 수 있다.

<참고> 소리 탐지(Sound Alert)

컴퓨터가 주의 또는 경고를 위한 음향효과를 제공하는 대신 화면을 깜빡임으로 시각적으로 주의 또는 경고할 수 있는 방법을 소리 탐지(Sound Alert)라고 하며, 대부분의 운영체제가 이 기능을 제공함

□ 잘못된 사례 및 개선방법

잘못된 사례 20) 방향, 위치 정보만을 이용하여 사용법을 알려주도록 구현한 경우

시각적인 위치나 방향을 이용하도록 지시하는 내용이 포함된 콘텐츠는 페이지 레이아웃의 변화가 예상되는 경우 또는 화면의 레이아웃을 시각적으로 인식하기 어려운 사용자가 사용하기 어렵다. 따라서 다음 사례는 잘못된 경우들이다.

- 1) 웹 페이지의 내비게이션 방법을 다음 페이지로 이동하려면 오른쪽 버튼, 이전 페이지로 이동하려면 왼쪽 버튼을 누르시오"와 같이 제공하는 것은 시각장애인이 사용할 수 없으므로 잘못된 콘텐츠이다.

개선방법

이를 개선하기 위하여 지시문을 "다음 페이지로 이동하려면 '다음' 버튼, 이전 페이지로 이동하려면 '이전' 버튼을 누르시오"와 같이 수정해야 한다.

<참고>

컨트롤이나 입력 서식 등을 웹 페이지 내의 상대적인 위치를 이용하여 지시하는 경우 실제 웹 콘텐츠의 위치와 웹 페이지 표현 위치가 다르면 혼란을 줄 수 있으므로 주의해야함

- 2) 다음 코드는 '미리보기' 버튼이 화면상의 오른쪽 하단에 표시하고 위치와 버튼 레이블을 이용하여 버튼을 선택하도록 지시하는 콘텐츠이다. 이 경우, 시각장애인은 콘텐츠를 선형화하여 읽을 수 있으므로 '오른쪽 하단'의 의미가 불분명하다. 따라서 위치에 대한 정보를 이용하기보다 그냥 해당 버튼의 레이블을 이용하여 지시하는 것이 좋다.

개선 전:

```
<table>
  <tbody>
    <tr>
      <td colspan="2">오른쪽 하단의 [미리보기] 버튼을 누르시오.</td>
      <td>
        <span style="background: ButtonFace; color: ButtonText; border:
          medium outset ButtonShadow;
```

```

width: 5em; display: block; font-weight: bold; text-align: center;">
인쇄</span>
</td>
</tr>
<tr>
<td>
<span style="background: ButtonFace; color: ButtonText; border:
medium outset ButtonShadow;
width: 5em; display: block; font-weight: bold; text-align: center;">
취소</span>
</td>
<td>
<span style="background: ButtonFace; color: ButtonText; border:
medium outset ButtonShadow;
width: 5em; display: block; font-weight: bold; text-align: center;">
확인</span>
</td>
<td>
<span style="background: ButtonFace; color: ButtonText; border:
medium outset ButtonShadow;
width: 5em; display: block; font-weight: bold; text-align: center;">
미리보기</span>
</td>
</tr>
</tbody>
</table>

```

개선방법

고령자나 약시자의 경우 브라우저의 글자체를 확대시켜 콘텐츠를 표시하면 콘텐츠의 표시 위치가 지시하는 위치와 달라져 혼란을 줄 수 있다. 따라서 가급적 위치 정보를 이용하여 지시하지 않도록 콘텐츠를 구현하는 것이 좋다.

즉, 다음과 같이 위치 정보를 제공하지 않고 버튼 레이블을 이용하여 지시하도록 수정한다.

개선 후:

```

<table>
<tbody>
<tr>
<td colspan="2"> [미리보기] 버튼을 누르시오.</td>

```

```

<td>
  <span style="background: ButtonFace; color: ButtonText; border:
  medium outset ButtonShadow;
  width: 5em; display: block; font-weight: bold; text-align: center;">
  인쇄</span>
</td>
...

```

잘못된 사례 21) 화면 위치만을 이용하여 객체를 지정한 경우

검색포털에서 어떤 단어에 대한 검색 결과에 대하여 관련 링크를 제공하는 경우 검색 페이지의 콘텐츠 중 '더 많은 결과는 좌측의 관련 링크를 참고하십시오.'라는 텍스트를 제공한다면 시각장애인은 관련 링크가 어느 곳에 위치하는지를 인식할 수 없으므로 관련 링크로의 접근이 불가능하다.

개선방법

이를 개선하기 위해서는 검색 페이지 내에 관련 링크로 직접 이동할 수 있는 링크를 제공한다.

잘못된 사례 22) 버튼 모양만을 이용하여 사용법을 알려준 경우

온라인 서식을 작성하는 페이지의 마지막에 세 개의 버튼을 위치시킨 경우를 생각해 보자. 작성 과정에서 작업을 종료하고 제출하기 위한 지시문을 "작업을 저장하지 않고 종료하려면 네모 버튼, 현재까지 작업을 저장하고 종료하려면 세모 버튼, 작성을 계속하려면 둥근 버튼을 누르시오."라고 제시한다면 모양을 구별하지 못하는 화면 낭독 프로그램 사용자는 필요한 작업을 수행할 수 없다.

개선방법

이를 개선하기 위해서는 버튼에 텍스트를 제공하는 등의 추가적인 정보를 제공하고 지시문에 이를 반영해야 한다. 위의 예에서 세 개의 버튼에 각각 '종료', '저장', '계속'이라는 레이블을 제공하고, 지시문을 "작업을 저장하지 않고 종료하려면 '종료' 버튼, 현재까지 작업을 저장하고 종료하려면 '저장' 버튼, 작성을 계속하러

면 '계속' 버튼을 누르시오."와 같이 수정한다.

잘못된 사례 23) 특수 기호만을 이용하여 사용법을 알려준 경우

그래픽 심벌을 조합한 단어는 사용자가 이해하기 어려우므로 사용하지 않아야 한다. 여기에서 그래픽 심벌은 이미지, 텍스트 이미지, 장식용 문자 등으로 받을 수 없는 문자들이다. 예를 들어, 웃는 모습(☺), 화살표(→), 체크표시(✓) 등은 의미가 있는 심벌들이나 보조기술은 이들 그래픽 심벌의 의미를 파악하기 어렵다. 따라서 보조기술이 인식할 수 있기 위해서는 그래픽 심벌을 이미지로 표현하고 이미지에 대한 대체 텍스트를 제공하도록 한다.

- 1) 온라인 쇼핑몰에서 상품 이름 앞에 그래픽 심벌 ☑와 ☒를 이용하여 각각 해당상품이 '구입가능' 또는 '구입불가'임을 알려주는 예는 화면 낭독 프로그램 사용자는 상품이 구입 가능 여부를 알 수 없으므로 잘못된 경우다.

개선방법

이를 개선하기 위해서는 그래픽 심벌을 이미지로 제공하고 이미지에 대체 텍스트를 제공하여 해결한다.

- 2) 다음 그림은 공인인증서 저장매체 선택 목록이다. 그림에서 그래픽 심벌 '▼' 는 이동식 디스크 목록을 열기 위한 버튼이다. 그러나 이 그래픽 심벌은 보조 기술이 의미 없는 문자로 취급하므로 보조기술 사용자는 그 의미를 알 수 없다.



개선방법

이를 개선하려면 그래픽 심벌 '▼'을 이미지로 구현하고 이 이미지에 대체 텍스트를 제공한다. 이 이미지에 대한 대체 텍스트로는 "이동식 디스크 선택"이 한 가지 예가 될 수 있다.

검사항목 1.3.3 (텍스트 콘텐츠의 명도 대비) 텍스트 콘텐츠와 배경 간의 명도 대비는 4.5대 1 이상이어야 한다.

□ 개요

웹 페이지에 보이는 핵심 텍스트 콘텐츠와 배경 간의 충분한 대비를 제공하여, 저시력자, 색각 이상자, 고령자 등도 콘텐츠를 인식할 수 있도록 제공해야 한다. 이를 위하여 보통 크기(18pt 미만, 또는 굵은 14pt 미만)의 텍스트는 4.5:1을 만족해야 한다. 그 이상의 텍스트(18pt 이상, 또는 굵은 14pt 이상)의 텍스트는 3:1을 만족해야 한다.

본문 콘텐츠에 단순히 장식 목적으로만 사용한 텍스트, 로고 또는 상호와 같은 텍스트 이미지, 마우스나 키보드를 활용하여 초점을 받았을 때 색이나 명도 대비가 변화하는 콘텐츠, 사용할 수 없음을 표시하기 위하여 명도 대비를 낮춘 회색의 컨트롤이나 입력 서식 등은 이 검사 항목의 적용을 받지 않는다.

<참고>

이 검사항목은 웹 페이지가 처음 로드된 상태의 페이지에 적용됨. 평가에 사용하는 웹 브라우저의 확대 배율은 기본 또는 100%로 지정함

□ 적용 기술

적용 기술 23) 배경과 텍스트 콘텐츠의 색 지정 방법

배경과 텍스트 콘텐츠 간의 명도 대비를 만족하기 위한 구현 방법은 다음과 같다.

- 1) 단색 배경의 경우 텍스트간의 명도 대비를 만족하도록 배경색과 텍스트 색을 선정한다.
- 2) 배경과 텍스트에 무늬를 주는 경우 상대적인 명도 대비가 4.5:1 이상이 되도록 한다. 예를 들어, 문자의 아랫부분을 어둡게 하고 윗부분을 밝게 표시하는 경우 명도 대비를 만족하기 어렵다. 이 경우, 문자의 배경을 위쪽은 검게 하고 아래쪽은 밝은 색으로 디자인하면 이 검사항목을 만족할 수 있다.

- 3) 배경색과 텍스트를 같은 계열의 색을 사용하여 명도 대비를 만족할 수 없을 경우 텍스트를 구성하는 문자에 테두리(halo)를 표시하여 테두리와 문자간의 명도 대비 차이를 이용하여 텍스트의 구분이 가능하도록 구현한다.

적용 기술 24) 배경과 텍스트 콘텐츠 색을 기본 값으로 지정하는 방법

배경과 텍스트 색을 지정하지 않으면 웹 브라우저가 제공하는 배경색과 전경색이 적용된다. 이 경우, 사용자는 웹 브라우저가 제공하는 배경과 텍스트 색을 자신의 필요에 맞는 CSS를 이용하여 사용할 수 있다.

적용 기술 25) 배경과 텍스트 콘텐츠 색을 변경하는 컨트롤의 제공

웹 페이지에 배경과 텍스트 색을 변경하는 버튼을 제공하고 버튼을 누를 때마다 배경색과 전경색을 변경하는 기능을 제공하는 웹 페이지를 제공할 수 있다.

이 경우에도 기본 확대 배율로 설정된 웹 브라우저에 처음 로딩된 웹 페이지의 명도 대비는 이 검사항목을 만족해야 한다.

<참고> 명도 대비 평가 도구

배경과 텍스트 간의 명도 대비를 검사할 수 있는 평가 도구는 다음과 같음

- 1) Colour Contrast Check(snook.ca) -

http://www.snook.ca/technical/colour_contrast/colour.html

- 2) Colour Contrast Analyzer(Colors on the web) -

<http://www.colorsontheweb.com/colorcontrast.asp>

- 3) Contrast Analyser(The Paciello Group) -

<http://www.paciellogroup.com/resources/contrast-analyser.html>Windows/MACApplication

- 4) Colour Contrast(Juicy Studio) -

<http://juicystudio.com/services/luminositycontrastratio.php>



□ 잘못된 사례 및 개선 방법

잘못된 사례 24) 전경색은 기본 값으로 두고 배경색만을 지정한 경우

웹 콘텐츠에서 전경색과 배경색을 지정하지 않으면 웹 브라우저가 제공하는 전경색과 배경색을 조합을 이용하여 명도 대비를 조절할 수 있다. 웹 콘텐츠의 배경색을 지정하면 웹 브라우저의 배경색 변경 설정 기능이 적용되지 않는다.

- 1) 다음 예는 body 태그의 속성을 이용하여 배경색은 지정하였으나 전경색을 지정하지 않은 잘못된 경우다. 이 사례는 약시자가 선호하는 설정(검정색 배경, 흰색 전경표시 등)을 적용할 수 없다. 이를 개선하기 위해서는 전경색과 배경색을 모두 지정하거나 모두 지정하지 않는 것이 좋다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <html>
    <head>
      <title>웹 사용성 소개</title>
    </head>
    <body bgcolor="white">
      <p> ... document body...</p>
    </body>
  </html>
```

개선방법

전경색과 배경색은 항상 함께 지정하거나 함께 지정하지 않도록 한다.

2) 다음 예는 CSS를 이용하여 배경색을 지정하였으나 전경색을 지정하지 않은 경우다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html>
<head>
<title>배경색의 지정</title>
  <style type="text/css">

    body {background-color:white}
  </style>
</head>
<body>
  <p>배경색을 흰색으로 표시합니다.</p>
</body>
</html>
```

개선방법

전경색과 배경색은 항상 함께 지정하거나 함께 지정하지 않도록 수정한다.

잘못된 사례 25) 배경색은 기본 값으로 두고 전경색만을 지정하는 경우

잘못된 사례 24)와 유사한 경우로, 웹 콘텐츠의 전경색을 지정하고 배경색을 지정하지 않은 경우다.

1) 다음은 배경색은 지정하지 않고 전경색만을 지정한 잘못된 사례이다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html>
<head>
  <title>웹 사용성 소개</title>
</head>
<body color="white">
  <p>... document body...</p>
</body>
</html>
```

개선방법

전경색과 배경색은 항상 함께 지정하거나 함께 지정하지 않도록 변경한다.

- 2) 다음은 CSS를 이용하여 전경색을 지정하였으나 배경색은 지정하지 않는 잘못된 경우다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html>
<head>
  <title>전경색의 설정</title>
  <style type="text/css">
    body {color:white}
  </style>
</head>

<body>
  <p>전경색은 항상 흰색입니다.</p>
</body>
</html>
```

개선방법

전경색과 배경색은 항상 함께 지정하거나 함께 지정하지 않도록 변경한다.

<참고> 배경이미지와 텍스트

배경과 전경색의 명도 대비가 충분하지 않으면 약시자가 텍스트를 인식하기 어렵다. 또한 배경 이미지는 텍스트와 간섭을 일으키지 않아야 한다.

예를 들어, 배경 이미지의 짙은 회색 부분에 검정색 텍스트가 표시되도록 한다면 텍스트와 배경 간의 명도 대비가 작아 텍스트를 인식하기 어렵게 된다. 마찬가지로, 배경 이미지의 무늬와 텍스트 콘텐츠가 간섭을 일으키는 경우에도 텍스트의 인식을 방해하게 된다.

이를 방지하기 위해서는 텍스트와 텍스트가 표시되는 배경 이미지 영역 간의 명도 대비가 충분하도록 제공해야 한다.

검사항목 1.3.4 (배경음 사용 금지) 자동으로 재생되는 배경음을 사용하지 않아야 한다.

□ 개요

웹 페이지에서 자동으로 재생되는 배경음(동영상, 음성, 음악 등)으로 인해 화면 낭독 프로그램을 사용하는 사용자가 콘텐츠를 인식하는 데 방해가 받지 않아야 한다. 또한 콘텐츠는 배경음의 음량을 조절하더라도 화면 낭독 프로그램의 음량에는 영향을 주지 않도록 제공되어야 한다.

이 검사항목을 만족하기 위하여 다음 방법 중 한 가지를 적용한다.

- 1) 자동적으로 재생되는 배경음의 지속시간을 3초 미만으로 제한함
- 2) 3초 이상 재생되는 배경음을 사용할 경우, 반드시 배경음을 제어할 수 있는 수단(멈춤, 일시정지, 음량조절 등)을 웹 페이지의 첫 부분에 제공함

□ 적용 기술

적용 기술 26) 재생 시간이 3초 미만인 배경음의 사용

이 방법은 웹 페이지에 포함되어 자동적으로 재생되는 사운드 콘텐츠의 길이를 3초 미만으로 구현하고, 반복적으로 재생되지 않도록 제공하는 것이다.

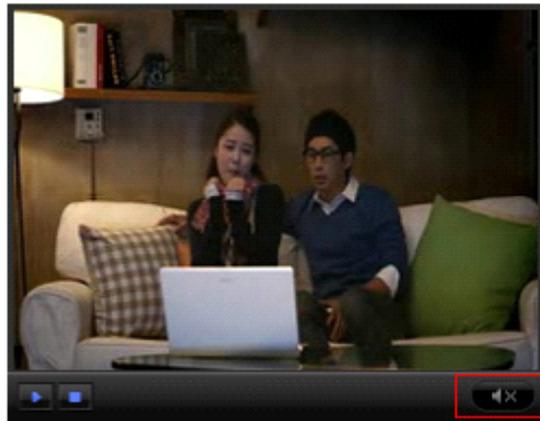
올바르게 구현한 사례는 다음과 같다.

- 웹 페이지를 로딩하면 자동적으로 팡파르가 들린 후에 종료된다.
- 웹 페이지를 로딩하면 대표이사가 "세상을 바꾸는 기업, 0000"라는 멘트가 한 번 제공된다.
- 웹 페이지를 로딩하면 "시작하려면 엔터키를 누르세요."라는 멘트가 한 번 제공된다.
- 웹 페이지를 로딩하면 자동적으로 재생되는 경고 음성을 한번만 제공한다.

적용 기술 27) 배경음을 정지 상태로 제공하는 방법

이 방법은 웹 콘텐츠에 포함된 배경음을 음소거 상태로 제공하고, 사용자 제어에 의하여 배경음을 들을 수 있도록 구현하는 방법이다. 따라서 이 방법은 사용자가 화면 낭독 프로그램을 사용하는데 아무런 영향을 주지 않는다.

- 고래의 대화를 연구하는 학회가 제공하는 웹 사이트가 고래의 울음소리와 물이 철썩이는 소리를 배경음으로 반복해서 제공해야 하려고 한다. 이 경우, 사용자가 배경음을 재생할 수 있도록 재생 버튼을 웹 페이지의 상단에 제공한다. 이때 버튼의 레이블을 배경음이 정지된 상태에서는 '배경음 재생', 배경음이 재생되는 동안에는 '배경음 멈춤'이라고 표시한다.
- 다음 그림에서 하단의 빨간색 네모표시 부분과 같이 배경음 제어 버튼을 제공한다. 또한 음소거 상태에서도 동영상은 계속 제공하는 것이 허용된다.



적용 기술 28) 플래시 콘텐츠의 배경음 자동 실행 방지

플래시 콘텐츠가 로딩 되면 자동적으로 재생되는 배경음은 그 재생을 멈출 수 있는 컨트롤을 사용자가 빨리 발견할 수 있는 위치에 제공해야 한다. 플래시 콘텐츠가 배경음을 제어할 수 있는 컨트롤을 제공하는 방법은 두 가지가 있다.

• 플래시 콘텐츠 컨트롤의 사용 방법

이 방법은 플래시 콘텐츠를 개발할 때 버튼을 추가하여 이 버튼을 누를 때마다 배경음의 정지 또는 재생을 반복하도록 컨트롤을 제공하는 것이다.

다음 코드는 플래시 콘텐츠가 로딩될 때 mp3 파일을 재생하거나 정지시키는 버튼을 제공하는 예다.

```
package wcagSamples {
    import flash.display.Sprite;
    import flash.net.URLRequest;
    import flash.media.Sound;
    import flash.media.SoundChannel;

    import fl.controls.Button;
    import fl.accessibility.ButtonAccImpl;

    import flash.events.MouseEvent;
    public class SoundHandler extends Sprite {
        private var snd: Sound = new Sound();
        private var button: Button = new Button();
        private var req: URLRequest = new URLRequest("http://av.adobe.com/podcast\
            /csbu_dev_podcast_epi_2.mp3");
        private var channel: SoundChannel = new SoundChannel();

        public function SoundHandler() {
            ButtonAccImpl.enableAccessibility();
            button.label = "Stop Sound";
            button.x = 10;
            button.y = 10;
            button.addEventListener(MouseEvent.CLICK, clickHandler);
            this.addChild(button);
            snd.load(req);
            channel = snd.play();
        }
        private function clickHandler(e: MouseEvent): void {
            if (button.label == "정지") {
                button.label = "재생";
                channel.stop();
            } else {
                channel = snd.play();
                button.label = "정지";
            }
        }
    }
}
```

- HTML을 이용한 플래시 재생음 제어

HTML을 이용하여 배경음의 재생 또는 정지 버튼을 구현하고 이 버튼을 누를 때마다 Flash Player JavaScript API를 이용하여 플래시 콘텐츠의 배경음을 제어할 수 있다.

ActionScript 3.0 코드:

```
package wcagSamples {
    import flash.display.Sprite;
    import flash.external.ExternalInterface;
    import flash.net.URLRequest;
    import flash.media.Sound;
    import flash.media.SoundChannel;

    import flash.events.MouseEvent;
    public class SoundHandler extends Sprite {
        private var snd: Sound = new Sound();
        private var soundOn: Boolean = true;
        private var req: URLRequest = new URLRequest("http://av.adobe.com/podcast/\
            csbu_dev_podcast_epi_2.mp3");
        private var channel: SoundChannel = new SoundChannel();

        public function SoundHandler() {
            if (ExternalInterface.available)
                ExternalInterface.addCallback("toggleSound", this.toggleSound);
            snd.load(req);
            channel = snd.play();
        }

        private function toggleSound(enable: Boolean): void {
            if (! enable) {
                channel.stop();
                soundOn = true;
            } else {
                channel = snd.play();
                soundOn = true
            }
        }
    }
}
```

HTML 코드:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="Content-Type"/>
    <title>플래시 배경음 재생 및 정지 예제</title>
    <script src="swfobject.js" type="text/javascript"/>
    <script type="text/javascript">
      function $(id) {
        return document.getElementById(id);
      }

      swfobject.embedSWF("html_control_to_toggle_audio_as3.swf",
        "flashPlaceHolder", "0", "0", "8");
      function init() {
        var soundOn = true;
        $("soundToggle").onclick = function(event){
          soundOn = !soundOn;
          $("flashPlaceHolder").toggleSound(soundOn);
          event.target.value = soundOn ? "정지" : "재생";
        };
      }
      window.onload = init;
    </script>

  </head>
  <body id="header">
    <h1>플래시 배경음 자동 재생 예</h1>
    <p>이 페이지는 오디오 방송 콘텐츠를 자동으로 재생하는 플래시 콘텐츠를 포함하고 있습니다.
    오디오를 정지시키려면 이 페이지의 '재생' 버튼 또는 '정지' 버튼을 누르시오. </p>
    <input id="soundToggle" type="button" value="Stop sound"/>
    <p id="flashPlaceHolder">플래시 콘텐츠를 사용하려면 플래시 플레이어를 설치하십시오. </p>
  </body>
</html>
```

- 화면 낭독 프로그램 설치 여부에 따라 배경음의 자동 재생을 제어하는 방법

배경음의 자동 재생을 억제하는 것은 스크린 사용자의 조작을 방해할 수 있기 때문이다. 따라서 플래시 콘텐츠를 구현할 때 화면 낭독 프로그램의 설치 여부를 검사하여 화면 낭독 프로그램이 설치되지 않은 경우 배경음을 자동적으로 재생되도록 하고 화면 낭독 프로그램이 설치된 경우 배경음을 정지 상태로 제공하는 것이 가능하다.

화면 낭독 프로그램의 설치여부를 검사하기 위하여 사용하는 플래시 속성은

flash.accessibility.Accessibility.active이다. 이 속성이 true이면 화면 낭독 프로그램이 설치되었음을 의미한다.

<참고>

- 1) 센스리더는 배경음 자동 재생 방지기능을 지원하지 않는다.
- 2) 이 속성은 화면 낭독 프로그램뿐만 아니라 MSAA 또는 UI 자동화를 지원하는 모든 보조기술을 화면 낭독 프로그램이 설치된 것으로 간주될 수 있으므로 주의해야 한다.
- 3) 플래시에서 화면 낭독 프로그램의 설치여부를 검사할 때 최소한 5 프레임 동안 검사할 필요가 있다.

다음 ActionScript 3.0 코드는 Accessibility.active 속성이 false이면 자동적으로 mp3 배경음을 재생하도록 구현한 플래시 콘텐츠이다. 다음 코드에서 사용하는 flash.system.Capabilities.hasAccessibility 속성은 MS Windows 환경에서 플래시 플레이어에 동작하고 있는가를 검사할 때 사용한다.

```
//ActionScript 3.0 코드:
package wcagSamples {
    import flash.accessibility.Accessibility;
    import flash.display.Sprite;
    import flash.net.URLRequest;
    import flash.media.Sound;
    import flash.media.SoundChannel;
    import flash.system.Capabilities;
    import fl.controls.Button;
    import fl.accessibility.ButtonAccImpl;
    import fl.controls.Label;
    import flash.events.MouseEvent;

    public class SoundHandler extends Sprite {
        private var snd: Sound = new Sound();
        private var button: Button = new Button();
        private var req: URLRequest = new URLRequest(
            "http://av.adobe.com/podcast/csbu_dev_podcast_epi_2.mp3");
        private var channel: SoundChannel = new SoundChannel();
        private var statusLbl: Label = new Label();
        public function SoundHandler() {
            snd.load(req);
            ButtonAccImpl.enableAccessibility();
            button.x = 10;
            button.y = 10;
            statusLbl.autoSize = "left";
            statusLbl.x = 10;
            statusLbl.y = 40;
            addChild(statusLbl);
        }
    }
}
```

```

button.addEventListener(MouseEvent.CLICK, clickHandler);
this.addChild(button);
if (! Capabilities.hasAccessibility || ! Accessibility.active) {
    channel = snd.play();
    button.label = "정지";
    statusLbl.text = "보조기술을 사용하지 않고 배경음을 자동적으로 제공함";
} else {
    button.label = "재생";
    statusLbl.text = "보조기술을 사용 중임. 배경음을 정지 상태로 제공함";
}
}
private function clickHandler(e: MouseEvent): void
{
    if (button.label == "정지")
    {
        button.label = "재생";
        channel.stop();
    }
    else
    {
        channel = snd.play();
        button.label = "정지";
    }
}
}
}
}

```

적용 기술 29) 실버라이트 콘텐츠의 배경음 자동 실행 방지

실버라이트 콘텐츠가 로딩되면 자동적으로 재생되는 배경음은 그 재생을 멈출 수 있는 컨트롤을 사용자가 빨리 발견할 수 있는 위치에 제공해야 한다. 실버라이트에서 멀티미디어 콘텐츠의 배경음을 제어할 수 있는 컨트롤을 제공하는 방법은 두 가지가 있다.

1) 자동 재생 방지

실버라이트 동영상을 개발할 경우 `MediaElement.AutoPlay` 속성을 이용하여 동영상의 자동 재생 여부를 지정한다. 기본 값은 `MediaElement.AutoPlay` 속성을 `true`로 설정하며 동영상을 자동 재생하도록 한다. 따라서 자동 재생을 방지하기 위해서는 `MediaElement.AutoPlay` 속성을 반드시 `false`로 설정해야 한다.

다음 예는 애니메이션 동영상 파일(xbox.wmv)을 로딩하고 정지 상태로 제공하는 실버라이트 콘텐츠의 코딩 예다.

```

<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition width="*" />
    <ColumnDefinition width="*" />
    <ColumnDefinition width="*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>

  <MediaElement x:Name="media" Source="xbox.wmv" AutoPlay="false"
    width="300" Height="300"
    Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3" />

  <!-- Stops media playback.-->
  <Button Click="StopMedia"
    Grid.Column="0" Grid.Row="1" Content="Stop" />

  <!-- Pauses media playback. -->
  <Button Click="PauseMedia"
    Grid.Column="1" Grid.Row="1" Content="Pause" />

  <!-- Begins media playback. -->
  <Button Click="PlayMedia"
    Grid.Column="2" Grid.Row="1" Content="Play" />
</Grid>

```

2) 음소거

동영상이 자동적으로 재생되더라도 배경음을 음소거 상태로 설정하여 재생음을 들리지 않도록 설정할 수 있다. 이를 위하여 `MediaElement.IsMuted` 속성을 `true`로 설정한다. 만일 `MediaElement.IsMuted` 속성을 지정하지 않거나 `false`로 설정한다면 항상 배경음을 제공하는 상태가 되므로 화면 낭독 프로그램 사용자에게 혼란을 주게 된다.

다음 예는 애니메이션 동영상 파일(`xbox.wmv`)을 로딩하자마자 재생되거나 음소거 상태로 제공되도록 `MediaElement.IsMuted` 속성을 설정한 실버라이트 콘텐츠의 코딩 예다.

```

...
<MediaElement x:Name="media" Source="xbox.wmv" IsMuted="true"
  width="300" Height="300"

```

Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3" />

...

□ 잘못된 사례 및 개선방법

잘못된 사례 26) 배경음이 3초 이상 지속되도록 구현한 경우

블로그 등에서 배경음악을 자동적으로 제공하는 것은 잘못된 경우다. 배경음악은 사용자의 요구로 제공 여부가 결정되어야 한다.

- 1) 회사의 제품을 소개하는 웹 사이트에서 제품의 로고송을 3초 이상 들려주는 것은 잘못된 경우다. 로고송의 지속시간은 3초 미만으로 제한해야 한다.

개선방법

개선 방법으로는 "로고송을 들으시려면 Control, A를 누르시오."라는 음성을 출력하도록 하고 Ctrl+A를 누르면 30초간 지속되는 로고송을 제공할 수 있다.

- 2) 영화 관련 웹 사이트로 콘텐츠가 로드되면 자동적으로 영화의 사운드 트랙이나 관객의 반응을 반복해서 제공하는 경우는 잘못된 사례이다. 다음 웹 콘텐츠는 배경음을 멈출 수 있는 컨트롤을 제공하지 않는다.



개선방법

배경음 컨트롤을 제공하거나 음소거 상태로 웹 콘텐츠를 제공하도록 개선한다.

잘못된 사례 27) 마우스 오버 또는 키보드 초점을 받으면 자동적으로 배경음이 실행되는 경우

초점을 받거나 마우스 오버에 의하여 콘텐츠가 활성화되고 배경음을 자동적으로 제공하는 것도 잘못된 사례이다. 초점의 변화와 마우스 오버에 의한 배경음의 자동 재생은 허용되지 않는다.

<참고> 마우스 오버(Mouse Over)

마우스 오버란 마우스 포인터를 어떤 개체 위에 위치시키는 것을 의미한다. 마우스 포인터를 어떤 개체 위에 위치시키면 OnMouseOver 이벤트가 발생하고 개발자가 지정한 특정한 기능을 수행할 수 있다. 이 사례는 마우스 오버에 의하여 애니메이션 플레이어를 자동적으로 실행하도록 한 경우이다.

다음 예는 마우스 오버에 의하여 애니메이션 플레이어가 자동적으로 실행되고, 동영상과 함께 배경음을 재생하는 경우이다. 우측 그림 하단의 빨간색 네모부분은 배경음을 재생하도록 설정되어 있는 모습을 보여준다.



개선방법

개선하기 위해서는 마우스 오버가 아닌 마우스 클릭(또는 Enter 키)에 의하여 애니메이션 플레이어가 실행되도록 수정해야 한다.

<참고>

마우스 오버에 의하여 멀티미디어 플레이어가 자동적으로 실행되는 것은 검사항목 3.2.1을 위반한 것이므로 허용되지 않는다.

이 경우, 좌측의 애니메이션에 대한 대체 텍스트를 "무한도전으로 이동, 마우스를 클릭하면 동영상 자동재생"이라고 제공한다.

원칙 2. 운용의 용이성(Operable): 사용자 인터페이스 구성 요소는 조작 가능하고 내비게이션 할 수 있어야 한다.

□ 운용의 용이성

웹 사이트에서 제공하는 콘텐츠의 모든 기능은 어떤 사용자나 환경에 상관없이 운용할 수 있어야 한다. 마우스를 사용할 수 없는 사용자나 환경에서는 키보드만으로도 웹 사이트의 기능을 이용할 수 있어야 하며, 키보드를 사용할 수 없는 사용자를 위해서는 마우스만으로도 웹 사이트의 기능을 이용할 수 있어야 한다. 키보드 운용 시에는 시각적으로 인식이 가능하도록 해야 하며 논리적으로 이동할 수 있도록 해야 한다.

온라인 서식을 작성할 경우에도 충분히 작성할 시간이 주어져야 하며, 시간제한이 있거나 자동으로 변하는 콘텐츠에 대해서는 사용자가 제어할 수 있도록 해야 한다.

번쩍거리는 콘텐츠는 제공하지 않는 것이 좋으며, 제공할 경우 사용자에게 미리 알려주거나 회피할 수 있도록 해야 한다.

웹 사이트 내에서 이동과 내비게이션이 용이하도록 콘텐츠를 잘 준비하는 것이 중요하다. 그러기 위해서는 페이지, 프레임, 콘텐츠 블록에는 적절한 제목을 제공해야 한다. 링크 또한 링크의 용도나 목적을 이해할 수 있도록 적절한 정보를 제공해야 한다.

□ 관련지침

운용의 용이성은 사용자가 장애유무 등에 관계없이 웹 사이트에서 제공하는 모든 기능들을 운용할 수 있게 제공하는 것을 의미한다. 운용의 용이성은 키보드 접근성, 충분한 시간 제공, 광고민성 발작 예방, 쉬운 내비게이션의 4가지 지침으로 구성되어 있다.

<표 7> 운용의 용이성 관련 지침 및 검사항목

지침(4개)	검사항목(8개)
<p>2.1(키보드 접근성) 콘텐츠는 키보드로 접근할 수 있어야 한다.</p>	<p>2.1.1(키보드 사용 보장) 모든 기능은 키보드만으로도 사용할 수 있어야 한다.</p>
	<p>2.1.2(초점 이동) 키보드에 의한 초점은 논리적으로 이동해야 하며 시각적으로 구별할 수 있어야 한다.</p>
<p>2.2(충분한 시간 제공) 콘텐츠를 읽고 사용하는 데 충분한 시간을 제공해야 한다.</p>	<p>2.2.1(응답시간 조절) 시간제한이 있는 콘텐츠는 응답시간을 조절할 수 있어야 한다.</p>
	<p>2.2.2(정지 기능 제공) 자동으로 변경되는 콘텐츠는 움직임을 제어할 수 있어야 한다.</p>
<p>2.3(광과민성 발작 예방) 광과민성 발작을 일으킬 수 있는 콘텐츠를 제공하지 않아야 한다.</p>	<p>2.3.1(깜빡임과 번쩍임 사용 제한) 초당 3~50회 주기로 깜빡이거나 번쩍이는 콘텐츠를 제공하지 않아야 한다.</p>
<p>2.4(쉬운 내비게이션) 콘텐츠는 쉽게 내비게이션 할 수 있어야 한다.</p>	<p>2.4.1(반복 영역 건너뛰기) 콘텐츠의 반복되는 영역은 건너뛸 수 있어야 한다.</p>
	<p>2.4.2(제목 제공) 페이지, 프레임, 콘텐츠 블록에는 적절한 제목을 제공해야 한다.</p>
	<p>2.4.3(적절한 링크 텍스트) 링크 텍스트는 용도나 목적을 이해할 수 있도록 제공해야 한다.</p>

검사항목 2.1.1 (키보드 사용 보장) 모든 기능은 키보드만으로도 사용할 수 있어야 한다.

□ 개요

웹 페이지에서 제공하는 모든 기능을 키보드만으로도 사용할 수 있도록 제공해야 한다. 키보드가 아닌 마우스나 태블릿 PC와 같이 정교한 조작을 요구하는 입력장치로만 접근이 가능할 경우 손이 불편하거나 손을 자유롭게 사용할 수 없는 사용자는 접근이 불가능하기 때문이다. 따라서 사용자가 키보드의 버튼 입력만으로도 동등하게 사용할 수 있도록 해야 한다. 여기서 마우스로는 한번으로 조작이 가능한 기능이지만 키보드를 이용할 경우 조작횟수가 늘어나는 것은 큰 문제가 되지 않는다. 이 검사항목이 적용되지 않는 웹 콘텐츠로 사용자의 반응 속도나 지속성이 중요한 요소인 붓질(Painting), 헬리콥터나 비행기 등의 훈련에 사용되는 시뮬레이션 콘텐츠 등과 시각적인 방법으로만 접근이 가능한 지리정보 콘텐츠, 가상현실 콘텐츠 등은 예외로 할 수 있다. 이 경우에도 해당 콘텐츠를 제외한 나머지 인터페이스는 키보드만으로도 사용할 수 있어야 한다.

만약 애플리케이션에 대한 기능일 경우 키보드로 접근도 가능해야 하며 기능도 실행 가능해야 한다.

<참고> 키보드란?

사용자가 텍스트를 입력하기 위하여 사용하는 입력장치를 의미한다. 여기에는 키보드의 자판 입력을 해독하기 위하여 사용되는 소프트웨어도 포함된다. 예를 들어, 키보드의 형태를 가지지 않았지만 기능적으로 키보드를 대신하는 입력장치(예: 노트북 및 PDA 등의 터치패드, 화상키보드, 음성입력장치 등) 등도 키보드로 간주한다. 따라서 '키보드만으로도 사용할 수 있어야 한다'는 의미는 마우스를 전혀 사용할 수 없다는 의미가 아니라 전통적인 자판 형태의 키보드만으로도 사용할 수 있어야 한다는 의미이다.

□ 적용 기술

적용 기술 30) 키보드와 마우스 이벤트 핸들러 제공

마우스로 접근 가능 기능이라면 키보드로도 접근이 가능해야 한다. 가능한 한 키보드와 마우스에 공통적으로 적용되는 이벤트 핸들러를 이용한다. 그렇지 않으면

면 두 가지 방식의 이벤트 핸들러를 함께 제공하도록 한다. 예를 들어, 마우스오버(onmouseover) 이벤트 핸들러나 마우스아웃(onmouseout) 이벤트 핸들러를 이용한 기능이라면, 키보드 초점이 위치할 때(onfocus)와 초점이 빠져나갈 때(onblur)의 이벤트 핸들러도 함께 제공해야 한다.

1) 마우스와 키보드 이벤트 핸들러를 동시에 제공하는 링크

다음 예는 링크에서 onmouseover 이벤트 핸들러와 onmouseout 이벤트 핸들러와 함께 onfocus 이벤트 핸들러와 onblur 이벤트 핸들러를 함께 사용하여 마우스와 키보드가 동일한 기능을 수행하도록 구현한 경우다.

```
<a href="#" onmouseover="process('true', 'on');" onmouseout="process('false', 'off');" onfocus="process('true', 'on');" onblur="process('false', 'off');"></a>
```

<참고> 마우스와 키보드 이벤트 핸들러 비교

onclick 이벤트 핸들러는 키보드로 어떤 객체를 선택했을 경우 발생하며 마우스의 버튼을 눌렀다 놓았을 경우에도 발생하는 이벤트 핸들러이다. 따라서 onclick 이벤트 핸들러를 이용하여 링크를 사용하도록 구현하면 마우스 또는 키보드를 사용하여 사용자의 명확한 입력에 의해 이벤트 핸들러가 발생하도록 제공할 수 있다.

다음 이벤트 핸들러는 마우스와 키보드별로 구분되어 있는 이벤트 핸들러이다.

마우스 이벤트		키보드 이벤트	
이벤트 명	이벤트 발생 시점	이벤트 명	이벤트 발생 시점
mousedown	마우스 버튼을 눌렀을 때	keydown	어떤 키가 눌렸을 때
mouseup	마우스 버튼이 복귀될 때	keyup	어떤 키가 눌렀다가 놓았을 때
mouseover	마우스 커서가 객체 위에 있을 때	focus	어떤 객체가 초점을 받았을 때
mouseout	마우스 커서가 객체영역을 벗어날 때	blur	어떤 객체가 초점을 벗어날 때

2) 마우스와 키보드 이벤트 핸들러를 동시에 제공하는 버튼

다음 예는 버튼에 onmouseover와 onmouseout이 발생하는 기능을 키보드로도 동일한 기능을 수행하도록 구현한 것이다.

```
<input type="button" value="미리보기" onmouseover="process('true', 'on');"
onmouseout="process('false', 'off');" onfocus="process('true', 'on');"
onblur="process('false', 'off');" />
```

<참고> 키보드 접근이 가능한 링크와 버튼

HTML 4.01에서 키보드 접근 및 사용자 조작으로 인한 이벤트 핸들러가 가능한 태그는 <a>, <area>, <button>, <input>, <object>, <select>이 있다.

적용 기술 31) 마우스 드래그와 드롭 기능에서의 키보드 이용 보장

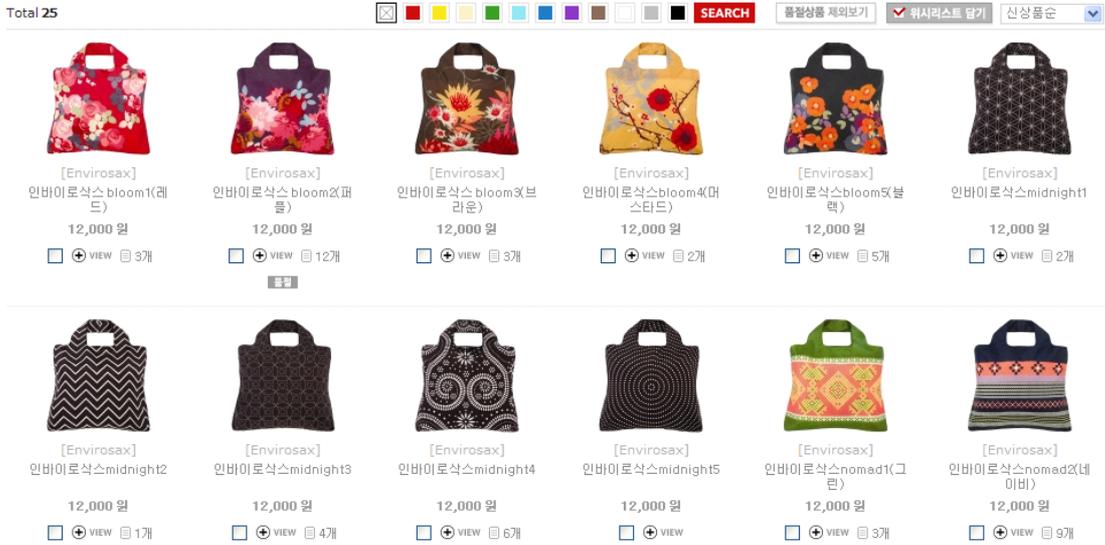
마우스 드래그와 드롭 기능은 마우스로만 구현이 가능한 기능이다. 또한 콘텐츠의 특성에 따라 유사한 기능을 키보드로 수행할 수 있도록 제공해야 한다.

<참고>

마우스를 클릭한 상태에서 포인터를 이동하는 마우스 드래그 기능은 마우스를 클릭한 항목으로부터 마우스 포인터가 위치한 곳에 이르는 항목을 선택하는 과정이다. 또한 드롭 기능은 마우스로 선택한 항목을 마우스 포인터가 지시하는 곳으로 이동시키는 기능이다. 키보드를 이용하여 마우스 드래그와 드롭 기능을 동일하게 수행할 수는 없다. 그러나 키보드를 이용하여 잘라내기와 붙여넣기를 수행함으로써 마우스 드래그와 드롭 기능을 대신할 수 있다.

1) 체크 상자를 이용한 상품 선택

쇼핑몰에서 상품을 선택하여 장바구니에 담는 기능은 키보드를 이용하여 수행할 수 있어야 한다. 예를 들어, 모든 상품에 장바구니에 담기용 체크 상자를 제공하고 '장바구니에 담기' 버튼을 클릭하면 체크상자가 '체크된' 상품을 장바구니에 담도록 구현하는 것도 가능하다.



2) 상품 선택 버튼을 이용한 방법

또 다른 방법은 상품별로 '장바구니에 담기' 버튼을 제공하고 이 버튼을 누를 때마다 해당 상품을 장바구니 목록에 포함되도록 구현하는 것도 가능하다.

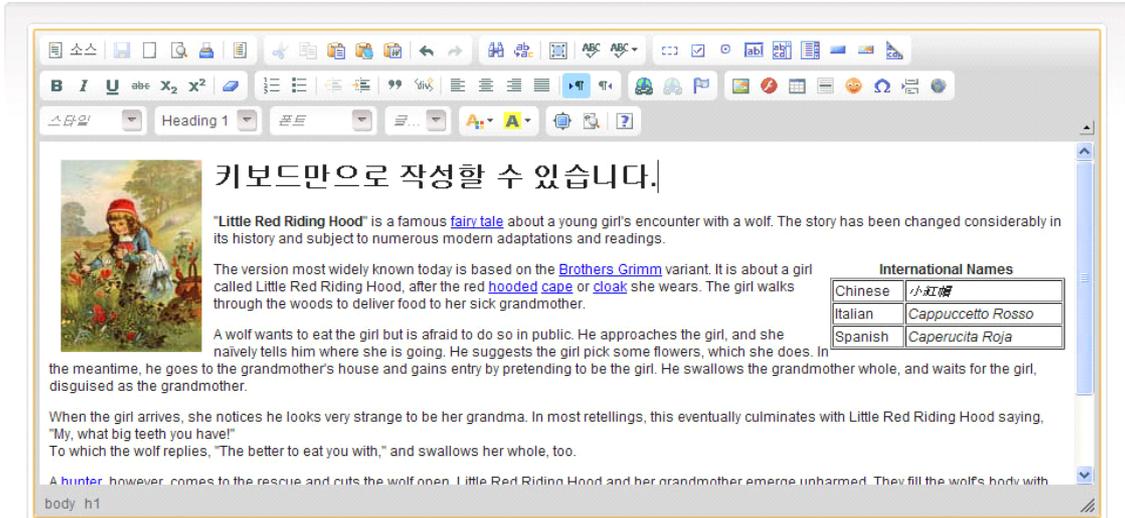
상기 1), 2) 방법의 기능에 추가하여 마우스 드래그 및 드롭 기능을 이용하여 상품을 선택하여 장바구니에 담는 기능을 함께 제공하면 마우스를 사용할 수 있는 사용자의 사용성도 함께 높일 수 있다. 여기서 중요한 점은 마우스 드래그 및 드롭 방법만을 사용하도록 구현해서는 안된다는 것이다.

키보드만으로 모든 기능을 사용할 수 있어야 할 뿐만 아니라 마우스만으로도 모든 기능을 이용할 수 있도록 구현하는 것은 접근성을 향상시키는 매우 좋은 방법이다.

적용 기술 32) 웹 애플리케이션 제공 시 키보드 이용 보장

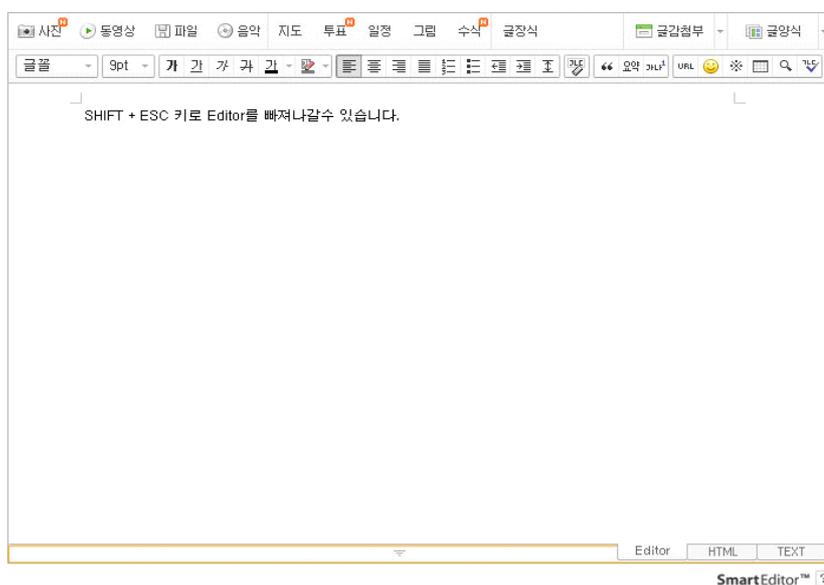
HTML의 기본 기능만으로는 제공이 어려운 다양한 기능을 포함한 애플리케이션일수록 키보드 접근이 떨어지는 경우가 있다. 최근 클라우드 컴퓨팅의 발전으로 여러 가지 기능의 웹 애플리케이션을 제공하는 경우가 점차 늘어나고 있고, 대표적인 예가 웹 에디터이다.

웹 에디터는 기본적으로 모든 기능이 키보드로 이동 가능해야 하며 실행 또한 가능해야 한다. 그리고 기능 영역과 작성 영역 간의 이동 역시 가능해야 한다.



탭 키를 이용한 내비게이션은 기본적으로 지원하는 키보드 내비게이션 기능이 다. 그러나 복잡한 애플리케이션에서는 단축키를 제공하고 미리 알려줄 필요가 있다. 단축키를 지원하면 웹 콘텐츠나 웹 애플리케이션을 더 효과적으로 사용할 수 있으며 키보드와 마우스를 사용하는 모든 사용자에게 편리함을 제공한다.

단축키를 제공하는 경우 사용자에게 단축키에 대한 정보를 제공해야 한다.



```
<!-- //재편집 안내 문구 -->
<iframe src="/lib/smartereditor2/6160/smart_editor2_inputarea.html"
id="se2_iframe" name="se2_iframe" class="se2_input_wysiwyg" width=
"400" height="300" frameborder="0" scrolling="yes" title="글쓰기 영역
: 글쓰기 영역에서 빠져 나오시려면 Shift+ESC키를 누르세요" style="height:
412px; display: block; ">
```

예를 들어, 단축키를 버튼의 툴 팁으로 제공하거나 컨트롤 레이블에 단축키를 표시하는 방법이 그것이다. 이러한 단축키 표시 방법을 access key 또는 니모닉(mnemonic)이라고 한다.

<참고>

단축키란?

단축키를 구현할 경우 웹 브라우저와 운영체제가 사용하고 있는 다양한 단축키와 충돌이 일어나지 않도록 세심한 주의를 기울여야 한다. 특히 웹 브라우저는 업데이트가 수시로 발생하기 때문에 새로운 단축키가 추가될 가능성이 있다.

또한 사용 가능한 모든 웹 브라우저와의 단축키 호환성을 검사하여야 한다. 충돌을 방지하기 위해서는 CTRL, SHIFT, ALT 등의 키를 사용하지 않는 것이 좋다. 그러나 문자를 입력해야 하는 경우 이 방법도 실제적이지 않다.

단축키를 정의하여 사용하는 방법은 충돌을 일으킬 가능성이 낮은 키 조합, 예를 들어 CTRL+SHIFT 키와 숫자 키의 조합을 이용하는 것이다. 숫자키를 이용한 단축키는 다국적용의 콘텐츠나 애플리케이션을 구현하는 경우에 효과적이다.

클라우드 컴퓨팅이란?

서버에 애플리케이션을 저장해 두고 사용자가 필요할 때 이를 장치로 불러와서 사용하는 웹 기반 소프트웨어 서비스이다. 웹 에디터를 온라인으로 제공하는 것도 클라우드 컴퓨팅의 일종이다.

키보드 이동 순서

키보드 내비게이션 순서는 원칙적으로 웹 콘텐츠에 포함된 컨트롤이나 사용자 입력의 코딩 순서를 따른다. 만일 그 순서를 변경하고자 할 경우 해당 객체의 tabindex 속성 값을 원하는 키보드 내비게이션 순서로 제공한다. 일부 태그의 tabindex 속성을 0으로 설정한 하면 이 태그의 키보드 내비게이션 순서는 코딩 순서에 따른다.

<참고> 정적 요소의 키보드 접근 허용

1) 웹 콘텐츠

이미지와 같은 정적 태그를 마우스를 이용하여 상호작용이 가능한 컨트롤로 구현하는 경우 초점을 받아 사용할 수 있도록 구현하려면 해당 태그의 tabindex 속성을 설정하고 onclick 이벤트를

핸들러에 의하여 함수를 호출하도록 구현해야 한다. 이 과정에서 tabindex 속성을 설정하더라도 다른 컨트롤이나 사용자 입력의 키보드 내비게이션 순서는 검사항목은 2.1.2에서 제시한 이동 순서를 유지해야 한다.

2) 실버라이트 콘텐츠의 경우

이미지를 이용하여 사용자 정의 컨트롤을 만들고 IsTabStop 값을 true로 설정하여 사용자 정의 컨트롤이 키보드 초점을 받을 수 있도록 한다. 사용자 정의 컨트롤이 초점을 받았음을 화면상으로 구분하기 위해서는 VisualStateManager 클래스를 사용한다.

더 쉬운 방법은 이미지를 버튼 컨트롤에 포함시켜 이미지 버튼으로 구현하는 것이다. 다음 소스코드는 이미지(sample.jpg)를 클릭하면 어떤 함수(doProcess)를 수행하는 예다.

```
<Button Click="doProcess"><Image Source="sample.jpg"/></Button>
```

적용 기술 33) 실버라이트 플러그인의 키보드 접근 제공

실버라이트 플러그인 내의 컨트롤이 키보드 접근이 가능하도록 만들기 위해서는 플러그인으로 초점이 이동할 수 있도록 구현해야 한다. 만일 웹 페이지가 HTML 콘텐츠와 실버라이트 플러그인으로 구성되어 있다면 플러그인으로서의 초점 이동이 가능하므로 실버라이트 플러그인의 키보드 접근이 가능하다. 그러나 웹 페이지를 하나의 실버라이트 플러그인으로 구성한다면 플러그인이 로딩될 시점에 초점이 플러그인의 첫 번째 컨트롤로 이동하도록 다음과 같이 구현해야 한다.

```
C# 버전:  
LayoutRoot.Loaded += (sender, e) =>  
    System.Windows.Browser.HtmlPage.Plugin.Focus();  
  
Visual Basic 버전:  
AddHandler LayoutRoot.Loaded,  
    Sub()  
        System.Windows.Browser.HtmlPage.Plugin.Focus()  
    End Sub
```

□ 잘못된 사례 및 개선방법

잘못된 사례 28) 마우스로만 접근 가능한 이미지 버튼을 제공하는 경우

tab 키로 이동이 가능한 기능을 제공할 수 있는 태그는 링크와 버튼이다. 그러나 이미지나 기타 태그에 마우스만으로 이용이 가능한 기능을 제공할 경우 키보

드를 이용하는 사용자는 그 기능을 이용할 수 없다.

다음은 이미지(image/print.gif)를 마우스로 클릭하면 인쇄하도록 구현한 것이다. 그러나 키보드를 이용하여 인쇄할 수 있는 다른 대체 수단을 제공하지 않았다면 이 구현 방법은 중요한 기능(인쇄)을 가진 이미지가 키보드 초점을 받을 수 없으므로 중요한 기능에 키보드 접근이 불가능한 잘못된 사례이다.

```

```

개선방법

키보드로도 접근이 가능하기 위해서는 이미지 버튼으로 구현해야 한다.

이미지 버튼 구현 사례

```
<button onclick="page_print();" ></button>
```

위 개선 방법으로 제공이 불가능한 경우라면 이미지에 tabindex를 설정하여 키보드 접근이 가능하도록 지정할 수 있다. 그러나 tabindex는 다른 키보드 접근 태그의 이동 순서에 방해되지 않는 방법으로 제공해야 한다.

<참고>

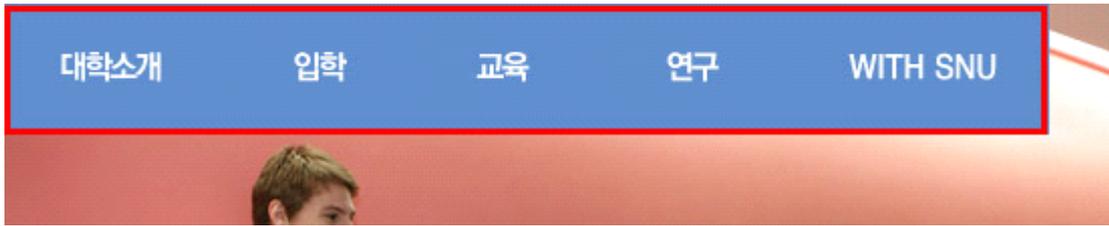
기본적으로 tabindex="0"이 설정되면 코드상의 순서에 맞게 키보드 이동순서가 설정되게 된다.

```

```

잘못된 사례 29) 키보드로 접근할 수 없는 플래시 메뉴

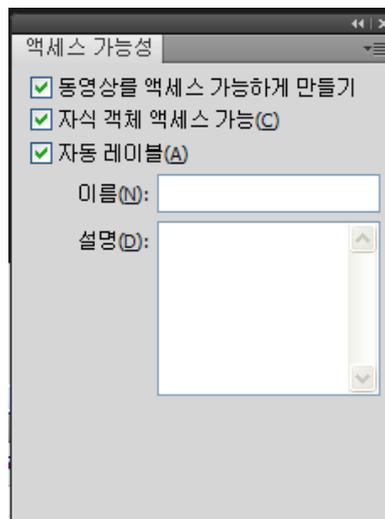
플래시로 메뉴를 개발하는 경우 다음 그림과 같이 키보드를 이용하여 초점을 메뉴항목으로 이동시킬 수 없도록 구현하면 마우스로는 사용이 가능하나 키보드로는 사용이 불가능하다.



마우스 장치 독립적인 이벤트 핸들러 제공

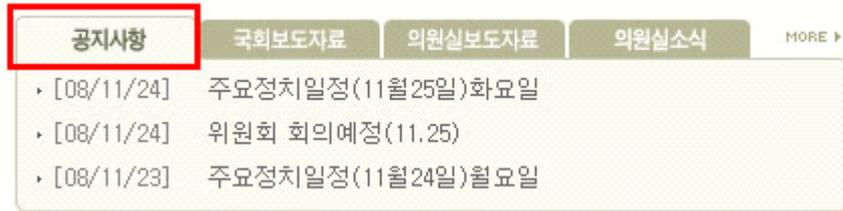
개선방법

키보드로 접근 가능한 메뉴를 구현하기 위해서는 액세스 가능성 패널에서 전체 무비에 대한 액세스 가능성 패널에서 '동영상을 액세스 가능하게 만들기'와 '자식 객체 액세스 가능' 체크 상자를 모두 체크상태로 설정한다. 여기에서 자식 객체란 플래시로 구현한 각 메뉴 및 서브메뉴 아이템을 의미한다.



잘못된 사례 30) 마우스로만 접근할 수 있는 링크를 제공한 경우

메뉴에 따라서 선택할 수 있는 항목이 화면에 표시되고, 항목을 선택할 수 있도록 다음과 같이 메뉴 목록을 구현하는 경우 메뉴 선택과 항목 선택을 마우스로만 가능하도록 구현하면 키보드 사용자의 접근이 불가능하게 된다.



마우스 장치 독립적인 이벤트 핸들러 제공

위 그림의 '공지사항' 메뉴를 다음과 같이 마우스용 이벤트만을 사용하여 링크로 구현하면 마우스로만 접근이 가능하게 된다.

```
<a href="#" onmouseover="show('show')" onmouseout="show('hide')">

</a>
```

이 소스코드는 마우스 포인터를 '공지사항' 메뉴 위로 이동시키면 메뉴의 색이 변화하고 해당 메뉴에 속한 선택 항목이 표시된다. 그러나 키보드 초점 변화에 의해서는 아무런 변화가 일어나지 않는다. 키보드 초점에 관한 검사 항목은 2.1.2를 참고하라.

개선방법

위 소스코드를 키보드 초점을 받을 때 메뉴의 색이 변화하고 해당 메뉴에 속한 선택 항목이 표시되도록 하려면 다음과 같이 onfocus 이벤트 핸들러와 onblur 이벤트 핸들러에 의해서도 각각 onmouseover 이벤트 핸들러와 onmouseout 이벤트 핸들러와 동일한 기능을 수행하도록 해야 한다.

```
<a href="#notice" onmouseover="show('show')" onmouseout="show('hide')"
onfocus="show('show')" onblur="show('hide')">

</a>
```

더 바람직한 방법은 마우스 클릭과 키보드 선택에 의하여 공통적으로 발생하는 onclick 이벤트를 사용하여 세부 항목이 나타나도록 하는 것이다.

```
<a href="#notice" onclick="show('show')">

</a>
```

잘못된 사례 31) 키보드만으로 이전 입력 서식 이동이 불가능한 경우

다음은 주민등록번호를 입력하는 예로, 주민등록번호의 앞 6자리를 입력하면 초점이 자동적으로 주민등록번호 뒷 7자리로 이동하는 예다.

본인의 이름 및 주민등록번호를 정확히 입력해주세요.

이름	<input type="text"/>	주민등록번호	<input type="text"/>	-	<input type="text"/>	<input type="button" value="확인"/>
----	----------------------	--------	----------------------	---	----------------------	-----------------------------------

이 경우, 주민등록번호의 앞 6자리를 입력하면 초점이 자동적으로 주민등록번호 뒤 7자리를 입력하는 입력상자로 이동하도록 구현되어 있어 두 번째 입력 상자에 초점이 있을 경우 shift + tab 키를 이용하여 초점을 첫 번째 입력상자로 이동시키더라도 초점이 자동적으로 두 번째 입력 상자로 이동한다.

개선방법

위의 사례에서 주민등록번호 첫 번째 6자리를 입력 후 다음 입력란으로 초점이 자동으로 이동되면 첫 번째 6자리 입력을 잘못했을 경우나 이름을 수정해야 할 경우 키보드로는 돌아갈 수 없는 문제가 발생한다. 이를 방지하려면 첫 번째 입력 상자에 6자리 번호를 입력하더라도 자동적으로 초점이 두 번째 입력 상자로 이동하지 않도록 구현하거나 이동하더라도 다시 뒤로 돌아갈 수 있도록 구현되어야 한다.

더 좋은 방법은 다음처럼 주민등록번호 정보를 하나의 입력필드에 입력할 수 있도록 제공하는 것이다.

환불/조회

해지환급금 보증금

○ 해지환급금

주민등록 번호

주민번호는 '-'없이 입력하세요.

검사항목 2.1.2 (초점 이동) 키보드에 의한 초점은 논리적으로 이동해야 하며 시각적으로 구별할 수 있어야 한다.

□ 개요

키보드 내비게이션 과정에서 키보드 초점을 받은 링크, 컨트롤 및 입력 서식은 초점을 받지 않은 객체들로부터 구분될 수 있어야 한다.

또한 Tab 키와 Shift + Tab 키에 의한 초점의 이동 순서는 논리적이며 일관성이 있어야 한다. 키보드 내비게이션의 순서는 일반적으로 웹 페이지의 좌측 상단 영역에서 우측 하단 영역으로 이동하는 것이 원칙이다.

<참고> 논리적인 키보드 내비게이션

논리적이며 일관성 있는 초점의 이동 순서란 다음 모든 조건을 만족하는 경우다.

- Tab 키와 Shift + Tab 키에 의한 초점 이동 순서가 순환적이며, 반복적이어야 함
- Tab 키와 Shift + Tab 키에 의한 키보드 내비게이션 과정에서 웹 페이지에 포함된 모든 컨트롤, 링크, 사용자 입력으로 초점 이동이 가능해야 함
- Tab 키와 Shift + Tab 키에 의한 키보드 내비게이션 과정에서 사용자가 사용할 수 없는 컨트롤, 링크, 사용자 입력 등의 객체로 초점이 이동하지 않아야 함
- Tab 키와 Shift + Tab 키에 의한 키보드 내비게이션 과정에서 웹 페이지를 구성하는 두 영역 사이에 반복적인 초점 이동이 일어나지 않아야 함
- 사용자 입력이 초점을 받았을 경우 아무런 정보를 입력하지 않아도 Tab 키 또는 Shift + Tab 키를 이용하여 다음 컨트롤, 링크 또는 사용자 입력으로 초점 이동이 가능해야 함

□ 적용 기술

적용 기술 34) 콘텐츠 화면 순서와 동일한 키보드 내비게이션 순서 제공

HTML의 경우, 기본적인 초점 순서는 태그들이 콘텐츠 소스에서 나타나는 순서를 따른다. HTML 소스의 순서가 웹 페이지 상에서의 시각적인 순서와 일치한다면, Tab 키를 눌러서 콘텐츠를 이동하는 순서는 시각적 레이아웃 순서와 일치하게 된다. 그러나 소스 순서가 시각적 순서와 일치하지 않는다면, 키보드 내비게이션 순서는 시각적으로 표시된 콘텐츠와 일치하지 않아 사용자에게 혼란을 줄 수

있다. 따라서 콘텐츠를 디자인할 때 컨트롤, 링크, 입력 서식 등의 인터랙티브 요소들이 기본적인 탭 순서를 따라갈 경우 콘텐츠 내에서의 순서와 관계에 걸맞게 위치하도록 배치해야 한다. 또는 콘텐츠의 논리적인 이동 순서를 돕기 위해 tabindex로 제어하는 방법도 있다.

<참고>

HTML에서 tabindex 속성은 두 가지 기능을 한다. 첫 번째는 태그가 초점을 가질 수 있게 하는 것이고, 두 번째는 초점 순서에서 하나의 위치를 태그에 할당하는 것이다. tabindex="0" 값은 태그가 초점을 가질 수 있도록 하지만 초점 순서는 코드상의 순서를 따르게 된다. 초점 순서는 tabindex의 양수 값이 작은 태그부터 큰 태그로 이동한다. 문서 객체 모델(DOM)의 태그 순서와 일치하지 않는 순서로 tabindex값을 설정한다는 것은 설정한 순서가 보조기술 사용자들에게는 정확하지 않다는 의미일 수 있다. 키보드 이동 순서가 다르게 되는 주된 요인은 콘텐츠의 이동 순서 배치를 구성 태그에 맞게 CSS로 하지 않고 HTML이나 XHTML에서 tabindex속성으로 지정하기 때문이다. 따라서 tabindex를 설정한 순서는 시각적인 표현 순서와 다를 수 있다.

적용 기술 35) 시각적으로 표시 가능한 초점 표시 방법

키보드로 웹 페이지를 이동할 때 현재 초점이 어디에 있는지 알 수 있어야 한다. 약시자나 뇌병변 장애인, 상지 장애인 등 마우스와 같은 포인팅 장치를 이용하지 않는 사람들은 키보드와 같은 순차적인 탐색 장치가 매우 중요한 도구가 된다.

또한 일부 모바일 브라우저에서도 마우스 포인터가 해당 위치에 초점이 옮겨졌다는 것을 표시하거나 한 번의 손가락 터치에 의해 초점을 받았다는 것을 표시하기 위해 해당 링크나 입력 서식에 시각적인 테두리를 표시하도록 하고 있다.

그러나 이러한 시각적인 단서가 명확하게 표시되지 않아 실질적으로 키보드를 이용하면서 초점의 이동 내역과 현재 초점 위치를 파악하기가 쉽지 않다. 이 경우, 다음 두 가지 방법으로 초점 위치를 표현해 주는 것이 좋다.

1) CSS의 가상 클래스 ‘:focus’ 제공

CSS의 outline 속성을 이용해 초점 위치를 명확히 표시해 주는 방법으로 주로 키보드 사용자들에게 큰 도움을 준다.

CSS는 주로 정적인 꾸미기를 위해 사용하지만 초점 받은 태그에 대한 화면 표시를 정의하기 위해 :focus를 사용할 수 있다. 이때 :focus와 동일한 기능을 수행

하지만 마우스만 지원하는 `:hover`도 같이 사용하는 것이 좋다.

```
a:hover,  
a:focus { border:1px solid #ee6600; }
```

<참고> 웹 브라우저의 기본적인 초점 표시 방법



대부분의 브라우저에서는 초점이 받은 링크에 대해 위와 같이 시각적으로 표시해준다.

위의 그림에서 많이 쓰이는 브라우저인 인터넷 익스플로러 8과 파이어폭스는 초점이 받은 곳에 가는 점선 테두리를 표시해 주는데, 이것은 약시자들이 인식하기에 충분하지 않다. 따라서 다음과 같은 CSS 코드를 적용하면 초점 위치가 시각적으로 더 명확하게 표시되게 할 수 있다. 단, 인터넷 익스플로러 6과 7에서는 `outline` 속성을 지원하지 않으므로 적용되지 않는다.

```
:hover, :active, :focus {  
    outline-width: 2px !important;  
    /* outline 속성은 border와 달리 추가 공간을 차지하지 않는다. */  
    outline-style: solid !important;  
    -moz-outline-radius: 4px;  
    /* 파이어폭스에서 테두리를 둥글게 만들기 */  
}
```



CSS를 이용해 초점을 시각적으로 더 두드러지게 만든 경우

<참고> outline 표시문제

CSS를 잘못 정의하면 초점 받은 태그의 초점표시(outline 등)가 화면에 표시되지 않는다. 대표적으로 overflow:hidden 속성을 사용할 경우 발생할 수 있으며 webkit 엔진의 safari나 chrome과 같은 브라우저에서 임의로 outline 속성을 제거하면 나타나지 않는다. 이 경우, outline을 표시하지 않을 때 개발자가 초점을 표시하는 방법을 제공해야 한다.

2) 스크립트를 사용하여 배경색이나 경계선 변경

다음 방법은 스크립트를 이용하여 링크가 초점을 받으면 배경이 노란색으로 표시되고 초점을 잃게 되면 정상적으로 바뀌도록 제공할 수 있는 방법이다.

```

<script>
function toggleFocus(e1)
{
  e1.style.backgroundColor = e1.style.backgroundColor=="yellow" ? "inherit" :
"yellow";
}
</script>

<a href="example.html" onfocus="toggleFocus(this)"
onblur="toggleFocus(this)">focus me</a>

```

두 방법 모두 공통적으로 초점을 화면에 뚜렷이 표시할 수 있는 방법으로, 추가적인 style(CSS)를 제공하고 있다.

적용 기술 36) 플래시에서 키보드 내비게이션 순서 제공 방법

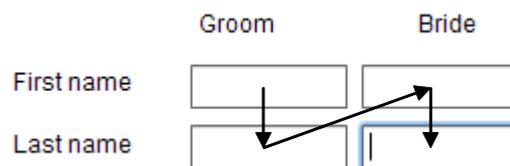
플래시 콘텐츠의 Tab 키에 의한 내비게이션 순서는 기본적으로 콘텐츠를 설계할 때 컨트롤을 추가한 순서와 동일하다. 만일 컨트롤의 추가 순서를 달리하여 컨트롤의 키보드 내비게이션 순서를 변경할 경우 키보드 내비게이션 순서가 화면상의 이동 순서와 동일하도록 XML코드를 변경하거나 모든 컨트롤의 TabIndex값을 사용자가 설정해 주어야 한다.

<참고>

플래시 무비에서 키보드 내비게이션 순서를 지정하기 위해서는 모든 정적 텍스트 태그를 동적 텍스트 변환하고, 다음의 두 가지 조건을 반드시 만족하도록 구현해야 한다.

- 1) 무비에 포함된 모든 텍스트 객체는 입력(input) 또는 동적 텍스트(dynamic text)로 정의한다. 이 경우, 글꼴 외곽선이 .swf 파일에 포함되므로 플래시 무비 파일의 크기가 커지게 되므로 주의해야 한다.
- 2) 무비에 포함된 모든 심볼 객체에는 객체이름을 지정해 주어야 한다.

예를 들어, 네 개의 입력상자(TextInput)를 2열로 배치하고 다음 그림의 화살표 방향에 따라 신랑의 이름과 성, 신부의 이름과 성의 순서로 키보드 내비게이션이 수행되도록 TabIndex 값을 설정할 수 있다.



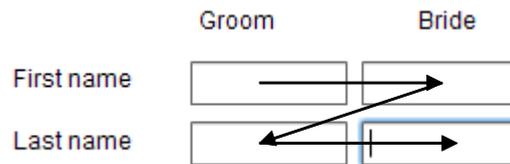
```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx"
    minWidth="955" minHeight="600">
    <fx:Declarations>
        <!-- Place non-visual elements (e.g., services, value objects) here
-->
    </fx:Declarations>
    <s:Label x="125" y="74" text="Groom&#xd;" width="49" height="17"/>
    <s:Label x="33" y="103" text="First name" width="80" height="17"/>
```

```

<s:Label x="33" y="134" text="Last name" width="80" height="17"/>
<s:Label x="234" y="74" text="Bride" width="49" height="17"/>
<s:TextInput x="121" y="99" width="78" id="GroomFirstName" tabIndex="10"/>
<s:TextInput x="206" y="99" width="78" id="BrideFirstName" tabIndex="30"/>
<s:TextInput x="121" y="129" width="78" id="GroomLastName" tabIndex="20"/>
<s:TextInput x="206" y="129" width="78" id="BrideLastName" tabIndex="40"/>
</s:Application>

```

위 소스코드에서 4개의 입력상자(TextInput)의 TabIndex 속성을 설정하지 않으면 키보드 내비게이션 순서는 다음 그림과 같이 입력상자의 생성 순서와 같이 신랑의 이름, 신부의 이름, 신랑의 성, 신부의 성의 순이 된다.



적용 기술 37) 실버라이트에서 키보드 내비게이션 순서 제공 방법

실버라이트 콘텐츠에서 컨트롤의 Tab 키 내비게이션 순서는 플래시 콘텐츠의 경우와 같이 콘텐츠를 설계할 때 컨트롤을 추가한 순서와 동일하다. 만일 컨트롤의 추가 순서를 달리하여 컨트롤의 키보드 내비게이션 순서를 변경할 경우 키보드 내비게이션 순서가 화면상의 이동 순서와 동일하도록 XAML를 변경하거나 모든 컨트롤의 TabIndex 값을 사용자가 설정해 주어야 한다.

예를 들어, 플래시 예와 같이 네 개의 입력상자(TextBox)를 2열로 배치하고 신랑의 이름과 성, 신부의 이름과 성의 순서로 입력상자 간의 키보드 내비게이션이 가능하도록 구현한 소스는 다음과 같다. 여기서 TabIndex 값을 생략하면 신랑의 이름, 신부의 이름, 신랑의 성, 신부의 성의 순서로 키보드 내비게이션이 일어난다.

```

<!--Custom tab order.-->
<Grid>
  <Grid.RowDefinitions>...</Grid.RowDefinitions>
  <Grid.ColumnDefinitions>...</Grid.ColumnDefinitions>

  <TextBlock Grid.Column="1" HorizontalAlignment="Center">Groom</TextBlock>

```

```

<TextBlock Grid.Column="2" HorizontalAlignment="Center">Bride</TextBlock>

<TextBlock Grid.Row="1">First name</TextBlock>
<TextBox x:Name="GroomFirstName" Grid.Row="1" Grid.Column="1"
    TabIndex="1"/>
<TextBox x:Name="BrideFirstName" Grid.Row="1" Grid.Column="2"
    TabIndex="3"/>

<TextBlock Grid.Row="2">Last name</TextBlock>
<TextBox x:Name="GroomLastName" Grid.Row="2" Grid.Column="1"
    TabIndex="2"/>
<TextBox x:Name="BrideLastName" Grid.Row="2" Grid.Column="2"
    TabIndex="4"/>

</Grid>

```

□ 잘못된 사례 및 개선방법

잘못된 사례 32) 초점을 일부러 보이지 않도록 한 경우

웹 디자이너들 중 일부는 링크나 입력 필드, 버튼 등에 초점이 가면 시각적으로 테두리가 생기는 것을 일부러 없애는 경우가 있다. 그러나 시각적인 테두리는 키보드로 웹 페이지를 탐색하는 사람들에게는 현재의 초점의 위치를 보여주는 매우 중요한 정보가 된다. 요즈음에 나온 브라우저들(예: 구글 크롬, 애플 사파리, 오페라 등)에서는 기존의 브라우저들(예: 인터넷 익스플로러, 파이어폭스)보다 오히려 더 과장되게 초점의 위치를 보여주도록 기본 값이 되어 있다. 초점을 받은 위치에 초점을 없애게 되면 키보드 내비게이션이 전혀 불가능하게 되어 키보드에 의존하는 대부분의 장애인들이 웹을 사용할 수 없게 될 뿐만 아니라 일반 사용자들도 매우 중요한 단서인 초점 위치를 잃어버려 사용성을 떨어뜨리는 요인이 된다.

서식 제출 버튼에 초점이 보이지 않도록 잘못 제작한 소스

```
<input type="submit" onFocus="this.blur();">
```

링크가 있는 이미지가 초점을 받았다는 것을 알 수 없도록 한 경우

```
<a onFocus="this.blur()" href="Page.html"></a>
```

링크 태그에 `onfocus="this.blur();"` 를 설정한 경우 키보드로 해당 태그에 접

근할 수 없을 뿐만 아니라 마우스로 접근할 경우에도 현재 위치를 시각적으로 알 수 없는 문제가 있다.

숙박 정보 | 음식 정보 | 엔터테인먼트정보 | 관광지 정보 | 투어정보

링크에 `onfocus="this.blur();" 사용`

```
<a href="accommodation.html" target="_blank" onfocus="this.blur();">숙박 정보</a>
```

개선방법

`onfocus=this.blur()`를 사용하면 마우스로 접근이 가능하지만 시각적으로 현 초점이 어디인지를 알 수 없고 키보드로 해당 태그에 접근이 되지 못한다.

뿐만 아니라 페이지 전체를 키보드로 내비게이션 할 수 없는 문제가 발생하기 때문에 `onfocus=this.blur()`은 사용하지 않아야 한다.

```
<a href="accommodation.html" target="_blank">숙박 정보</a>
```

잘못된 사례 33) 키보드 이동순서가 비논리적인 경우

다음 예제는 `tab` 키로 접근이 가능한 태그에 임의의 `tabindex`를 설정하여 키보드 이동 순서를 논리적이지 못하게 한 경우로, 이 경우 임의로 지정한 `tabindex` 속성값이 0이 아닌 다른 값이 지정되면 키보드 이동 순서가 순서대로 이동하기 전에 `tabindex`가 지정된 곳으로 먼저 이동되는 문제가 발생하게 된다.



개선방법

특정 태그에 tabindex를 별도로 설정하게 되면 전체 구성의 키보드 이동 순서가 화면 구성순서와 맞지 않게 된다. 따라서 tabindex는 가급적 사용하지 않는 것이 좋으며, 만약 tabindex를 사용해서 키보드로 접근이 필요한 경우라면 tabindex 값을 0으로 설정하여 전체 이동 순서에 문제가 되지 않도록 하는 것이 좋다.

검사항목 2.2.1 (응답시간 조절) 시간제한이 있는 콘텐츠는 응답시간을 조절할 수 있어야 한다.

□ 개요

시간제한이 있는 콘텐츠는 그 변화를 사용자가 제어할 수 없는 콘텐츠이다. 예를 들어, 일정한 시간마다 그 내용의 일부 또는 전체가 자동적으로 갱신되는 콘텐츠, 일정한 시간이 경과하면 다른 웹 페이지로 이동하도록 구성된 콘텐츠, 사용자가 제어하지 않아도 자동적으로 스크롤되는 콘텐츠, 일정시간 후 사라지거나 접근이 차단되는 콘텐츠 등이 시간제한이 있는 콘텐츠이다.

시간제한이 있는 콘텐츠는 시간적인 제약이 있음을 사용자에게 반드시 알려주어야 한다. 또한 다음과 같은 예외의 경우에 해당하는 콘텐츠를 제외한 모든 콘텐츠는 시간제약을 회피하거나 우회할 수 있는 수단을 함께 제공해야 한다.

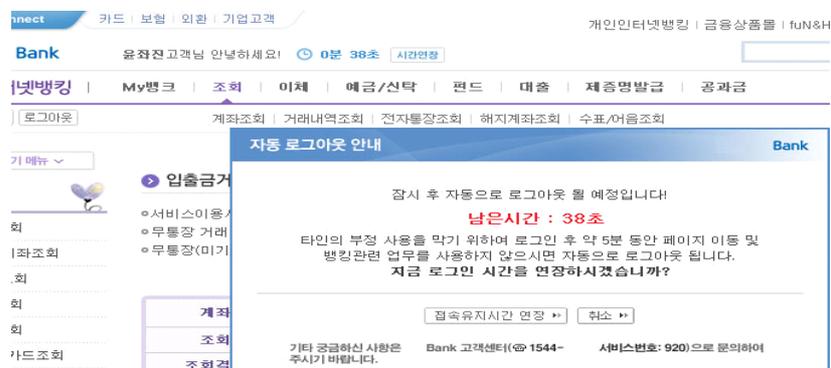
<참고> 시간제한 예외 콘텐츠

경매나 실시간 게임, 듣기평가용 콘텐츠 등과 같이 원천적으로 콘텐츠의 이용에 따르는 시간 조절을 허용할 수 없는 콘텐츠

□ 적용 기술

적용 기술 38) 제한 시간 연장 방법 제공

제한 시간을 가진 기능을 제공할 때 사용자가 제한 시간을 연장할 수 있도록 해야 한다.



이때 사용자가 연장할 시간을 직접 지정하거나 일정시간을 반복적으로 연장할 수 있는 방법을 사용할 수 있다.

적용 기술 39) 메타 태그를 이용한 페이지 재 이동 방법 제공

메타(meta) 태그를 이용하여 클라이언트측 자동 재이동(redirect)하는 경우 메타 태그가 설정되는 페이지의 content 값이 0보다 크면 재이동하는 페이지가 브라우저의 히스토리에 남게 되어 페이지 뒤로 이동할 경우 바로 이전 페이지가 재이동 페이지로 이동되는 문제가 있다.

```
<meta http-equiv="refresh" content="0;URL='redirect.html'" />  
<p><a href="redirect.html">redirect.html</a> 페이지로 이동합니다.</p>
```

메타 태그로 설정된 재이동 페이지는 시간을 연장하거나 해제할 수 없기 때문에 메타 페이지는 자동으로 재이동이 되지 못한 경우를 위한 최소한의 재이동 관련 정보만 제공하고 content 값은 반드시 0으로 설정해야 한다.

메타 태그를 이용한 페이지 이동에 content 값이 0보다 크다면 브라우저 페이지 history에 남기 때문에 이전 페이지로 돌아가기 힘들기 때문이다. 특히 시각장애인들이 인지하기 힘들다.

□ 잘못된 사례 및 개선방법

잘못된 사례 34) 페이지 재이동 시 회피 수단을 제공하지 않는 경우

1) 쇼핑몰 페이지 자동 이동 사례

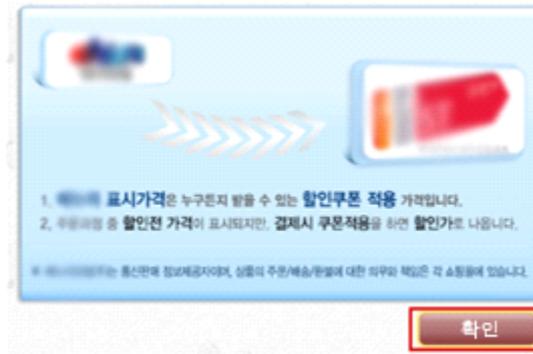
다음 그림은 자바스크립트를 이용하여 자동으로 페이지를 재이동하는 경우로, 이 경우 사용자는 재이동되는 시간동안 콘텐츠를 인식하지 못할 수 있다.



```
<SCRIPT>
setTimeout("location.href='http://www.example.com'", 5000);
</SCRIPT>
```

개선방법

자동으로 페이지가 이동하는 방식은 제공하지 않는 것이 좋다. 반드시 사용해야 하는 경우라 하더라도 다음 그림과 같이 확인 버튼을 이용하여 사용자 선택에 의해 페이지를 이동할 수 있도록 제공해야 한다.



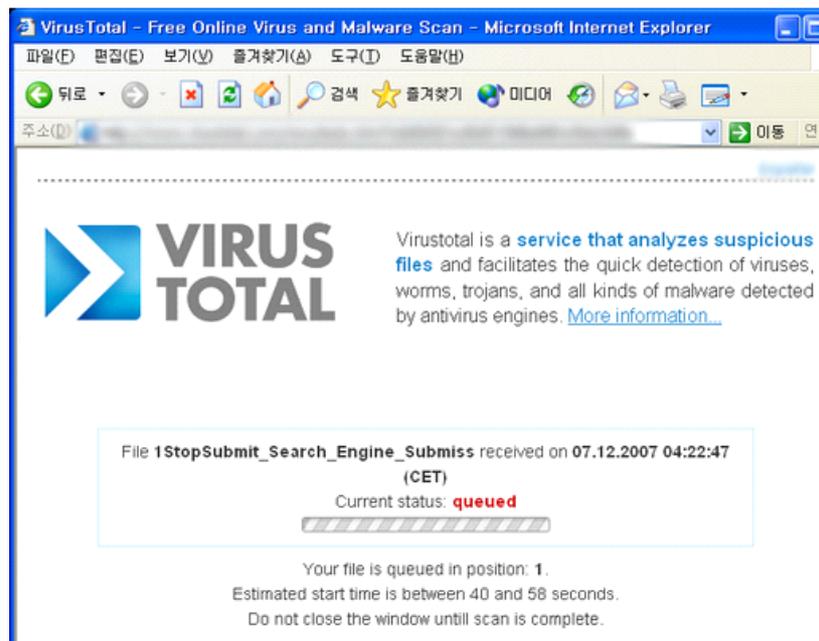
```
<p>
1. ○○○ 표시가격은 누구든지 받을 수 있는 할인쿠폰 적용 가격입니다.
2. 주문과정 중 할인전 가격이 표시되지만, 결제시 쿠폰적용을 하면 할인가로 나옵니다.
</p>
...
<a href="http://shopping.co.kr/?itemId=2344679gh79">확인</a>
```

2) 바이러스 자동 체크 사례

웹 페이지에 접속하자마자 특정 서비스(예를 들어, 온라인 바이러스 체크 등)를 수행할 경우 사용자는 자신이 이동하려 했던 페이지가 아니라 다른 페이지로 이동이 되어버리게 되어 이용에 혼란을 줄 수 있다.

개선방법

사용자가 바이러스 체크를 시작할 수 있도록 컨트롤을 제공해 주어야 한다. 또한 사용자가 어떤 서비스를 선택한 경우에도 해당 서비스를 수행하는 과정이 복잡하고 오랜 시간이 걸린다면 서비스를 중간에 멈추고 목적지 URL로 되돌아 갈 수 있는 수단을 제공해야 한다.



검사항목 2.2.2 (정지 기능 제공) 자동으로 변경되는 콘텐츠는 움직임을 제어할 수 있어야 한다.

□ 개요

저시력자나 지적장애인 등은 이동하거나 스크롤되는 콘텐츠를 인지하기가 어렵다. 때문에 웹 콘텐츠는 사용자가 이동이나 스크롤을 일시 정지시키거나 지나간 콘텐츠 또는 앞으로 나타날 콘텐츠를 사용할 수 있도록 "앞으로 이동", "뒤로 이동", "정지" 등의 컨트롤을 제공해야 한다.

<참고>

자동으로 변경되는 콘텐츠란 웹 콘텐츠에서 스크롤 및 자동 갱신되는 콘텐츠를 말하며, 자동으로 변화하는 배너, 텍스트 스크롤 등이 있다.

□ 적용 기술

적용 기술 40) 변화하는 콘텐츠를 일시정지 시키고, 일시정지 된 곳으로부터 다시 시작할 수 있도록 제공

갱신되는 콘텐츠를 읽고 있는 사용자는 충분한 읽을 시간을 얻기 위해 일시정지시킬 수 있어야 한다. 또한 멈추었다 다시 시작하면 읽고 있던 그 이후부터 변경되어야 한다.



다음 예를 참고하라.

- 페이지에 자동 스크롤되는 뉴스 배너가 있다. 그것을 읽고자 하는 사용자는 ESC 키를 눌러서 스크롤을 잠시 멈출 수 있고, 방향키를 다시 누르면 스크롤이 다시 시작된다.

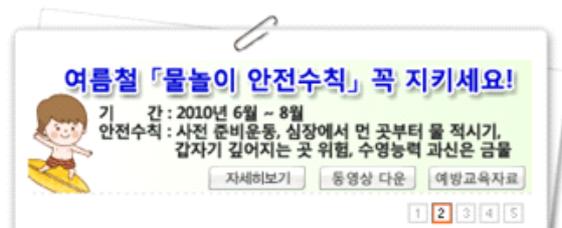
적용 기술 41) 전체 배너를 보여줄 수 있는 방법 제공

변화하는 콘텐츠를 제어할 수 있는 방법을 제공하지 않는 경우 전체 콘텐츠를 보여줄 수 있는 방법을 제공할 수 있다.



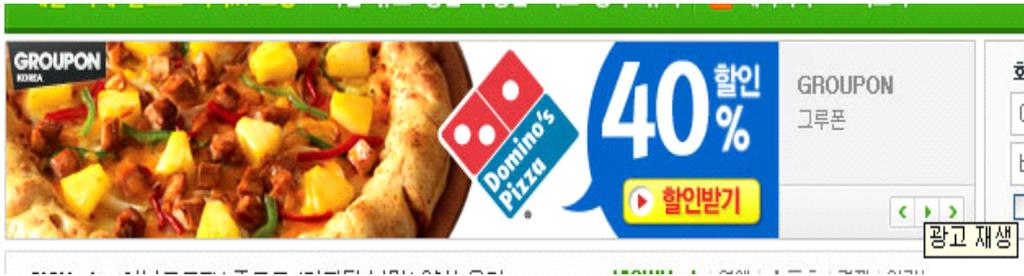
적용 기술 42) 전체 배너의 리스트 제공

변화하는 배너의 정지나 이전, 다음 기능을 제공할 수 없다면 사용자가 개별 배너를 직접 선택할 수 있는 방법을 제공해야 한다. 다음 그림은 5개의 배너를 직접 선택할 수 있도록 1~5까지 번호별로 링크를 제공하고 있다.



적용 기술 43) 사용자 요구에 의한 업데이트 방법 제공

사용자의 요구와 무관하게 콘텐츠가 자동적으로 업데이트되는 기능의 경우 사용자는 불편함을 겪게 된다. 따라서 업데이트를 자동적으로 발생시키는 대신, 사용자가 자신의 필요에 따라 업데이트를 요청할 수 있는 기능을 제공해야 한다.

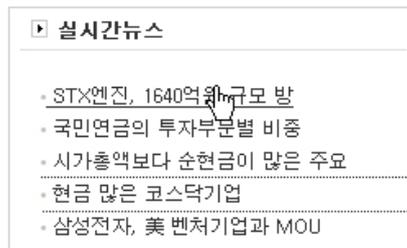


또한 자동으로 업데이트되는 콘텐츠를 사용자는 멈출 수 있어야 한다. 이를 위해 "일시정지" 버튼을 제공하여 자동으로 업데이트되고 있는 콘텐츠를 사용자가 원하는 만큼 정지시킨 뒤 이용이 다 끝난 뒤 다시 업데이트될 수 있도록 해야 한다.

□ 잘못된 사례 및 개선방법

잘못된 사례 35) 마우스로만 정지할 수 있는 콘텐츠를 제공한 경우

다음 그림은 실시간 뉴스 기사가 상하 스크롤로 갱신되는 경우이다. 이 경우, 사용자는 뉴스를 보던 도중에 내용이 변경되어 이전 뉴스를 찾게 되지만 사용자 제어 기능은 단지 해당 기사 링크에 마우스를 올렸을 경우에만 스크롤이 멈추도록 되어 있으며, 키보드 초점을 받은 경우 스크롤이 멈추지 않아 키보드 접근이 불가능한 사례이다.



개선방법

링크에 마우스를 올렸을 경우에만 작동되는 스크립트는 키보드로 접근했을 경우에도 동일하게 동작하도록 개선되어야 한다. 또한 텍스트 배너의 변화가 멈춘 다음에는 지나간 배너로 이동하거나 다음 배너로 이동할 수 있는 방법도 제공해

야 한다.

잘못된 사례 36) 자동으로 변하는 배너

다음 그림은 여러 개의 이미지 배너가 일정 시간이 지나면 순서대로 변경되어 보여주는 경우다.

사용자가 변화하는 배너를 멈추기 위해서는 해당 배너 위로 마우스를 올려 두어야만 멈출 수 있어 키보드로는 멈출 수 없는 문제가 있다.



개선방법

우선 마우스뿐만 아니라 키보드 초점을 받았을 경우에도 스크롤이 멈추도록 개선해야 하며, 또한 사용자가 배너를 선택할 수 있는 방법을 함께 제공해야 한다. 사용자가 배너를 선택할 수 있는 가능한 수단은 다음과 같다.

- 앞으로 이동, 뒤로 이동 등의 스크롤 제어 기능 제공
- 사용자 선택이 가능한 배너 번호와 번호에 동기되는 배너의 표시 기능 제공
- 전체 배너를 표시하는 방법 제공

검사항목 2.3.1 (깜빡임과 번쩍임 사용 제한) 초당 3~50회 주기로 깜빡이거나 번쩍이는 콘텐츠를 제공하지 않아야 한다.

□ 개요

깜빡이는 콘텐츠는 사용자의 시선을 끌기 위한 목적으로 주로 사용되는 콘텐츠이다. 그러나 깜빡임의 주기와 크기, 변화의 정도에 따라 광과민성 증후군이 있는 사용자에게 발작이나 졸도 등을 유발할 수 있으므로 깜빡이는 콘텐츠의 사용에 유의하여야 한다.

광과민성 발작은 주로 어린이들에게 많이 일어나며 특히 빛에 민감한 어린이들에게 일어날 가능성이 많다. 증상은 어지럼증을 유발하며, 심하면 의식을 잃고 온몸이 경련을 일으키기도 한다.

그러나 깜빡임이 있는 모든 콘텐츠가 위험한 것은 아니다. 문제가 되는 것은 게임, 애니메이션 등과 같이 명도 대비나 채도 대비(특히 빨간색과 파란색 계열)가 큰 화면이 연속적으로 오랜 시간 빠르게 반복(주파수 초당 3~50회)되는 경우에만 문제가 된다. 초당 2회 이하로 깜빡이는 콘텐츠는 문제를 야기하지 않는다. 3D 애니메이션의 경우 깜빡임 주파수와 관계없이 발작을 일으킬 가능성이 있다는 연구 결과가 있으므로 웹 페이지를 통하여 3D 동영상을 제공할 경우 주의하여야 한다.

(출처: 3D 입체영상 '포켓몬 쇼크' 재연될라, 이투데이 2011년 1월 13일,
<http://kr.news.yahoo.com/service/news/shellview.htm?articleid=20110113110000417f7&linkid=4&newsetid=1352>)

<참고> 포켓몬 사례

1997년 일본의 유명 애니메이션 '포켓몬스터'의 38번째 에피소드인 '전능 전사 포리곤'의 한 장면에서 빨간색과 파란색이 교대로 반복되는 섬광 장면이 나오자 TV로 이를 시청하던 어린이들이 발작을 일으킨 사고가 있었다. 이 사고로 어린이 685명이 구급차로 병원에 실려 갔으며 이들 중 150여 명은 병원에 입원한 바 있다. 일본에서는 어린이들의 발작 원인으로 애니메이션의 섬광 이미지로 인한 광과민성 발작 현상으로 결론내고 관련 업계와 방송국에 주의 조치를 내린 바 있다.

□ 적용 기술

적용 기술 44) 깜빡이는 시간이 3초 이내인 콘텐츠만 제공

페이지에 포함된 모든 콘텐츠는 움직임, 번쩍임, 깜빡임이 없도록 한다. 사용자의 시각적인 유인 효과를 위해서 콘텐츠의 움직임, 번쩍임, 깜빡임을 사용하는 경우 이들 움직임이 3초 이내에 자동적으로 멈출 수 있도록 제공한다.

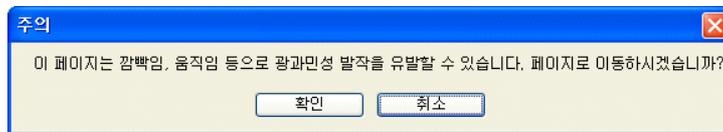
콘텐츠의 움직임, 번쩍임, 깜빡임이 콘텐츠를 사용하거나 이해하는 데 필수적인 경우 사용자가 그 기능을 활성화시키고 멈출 수 있는 컨트롤을 제공한다. 이 경우에도 처음 로드될 때에는 콘텐츠의 움직임, 번쩍임, 깜빡임이 정지 상태이거나 로드 후 그 움직임이 3초 이내이어야 한다.

움직임, 번쩍임, 깜빡임이 있는 페이지 경우에도 3초 이내로만 깜빡이는 콘텐츠를 제공한다면 움직임, 번쩍임, 깜빡임에 대한 불편을 줄일 수 있다. 이러한 방법을 이용해 3초 이내의 깜빡이는 콘텐츠를 제공하면 사용자의 시각을 끌 수 있어 팝업 창을 대체하는 용도로 사용될 수도 있다.

```
<style type="text/css">
#blink_contents{width:100px;height:100px;background-color:red;visibility:visible;
}
</style>
</head>
<body>
<div id="blink_contents">...</div>
<script type="text/javascript">
var i = 0;
function popupBlink(){
var blinkContents = document.getElementById("blink_contents");
if(i < 3){
blinkContents.style.visibility = blinkContents.style.visibility == "" ?
"hidden" : "";
i=i+1;
}else{
blinkContents.style.visibility = "";
window.clearInterval(blinking);
}
}
var blinking = window.setInterval(popupBlink, 500);
</script>
</body>
```

적용 기술 45) 경고 페이지 제공

콘텐츠의 움직임, 번쩍임, 깜빡임 등이 페이지를 사용하거나 이해하는 데 필수적이며, 사용자가 그 기능을 활성화시키거나 멈출 수 있는 컨트롤을 제공하기 어려운 경우 다음과 같은 메시지 상자를 이용하여 사용자에게 이들 움직임, 번쩍임, 깜빡임이 있는 페이지로 이동함을 알리고 사용자가 이 페이지로 이동할 것인가를 결정할 수 있도록 한다.



□ 잘못된 사례 및 개선방법

잘못된 사례 37) 깜빡이는 콘텐츠가 많은 경우

다음 그림은 페이지에 사용된 콘텐츠 중 광고배너, 이미지 등이 과도하게 움직임이 많은 경우이다. 이 경우, 사용자는 움직이는 콘텐츠가 많아 주위가 산만해지고 집중력이 떨어질 수 있다.



깜빡임 콘텐츠가 많은 페이지

개선방법

<개선 방법 1>

가장 바람직한 방법은 웹 페이지를 움직임이나 깜빡임이 없도록 구현하는 것이다. 특히 쇼핑몰에서 제공하는 상품 이미지를 플래시나 gif 애니메이션으로 구성하지 않고 정적 이미지로 제공한다.

<개선 방법 2>

그러나 움직임이나 깜빡이는 콘텐츠를 제공해야 하는 경우 콘텐츠의 움직임과 깜빡임은 3초 미만이 되도록 한다. 만일 깜빡임이 3초 이상 지속되는 경우 해당 웹 페이지에 접속되기 전에 광과민성 발작을 유발할 수 있음을 사용자에게 경고하여 사용자가 사전에 깜빡이는 콘텐츠를 피할 수 있도록 제공해야 한다.

<참고> 깜빡이는 콘텐츠 제공에 대한 참고 내용

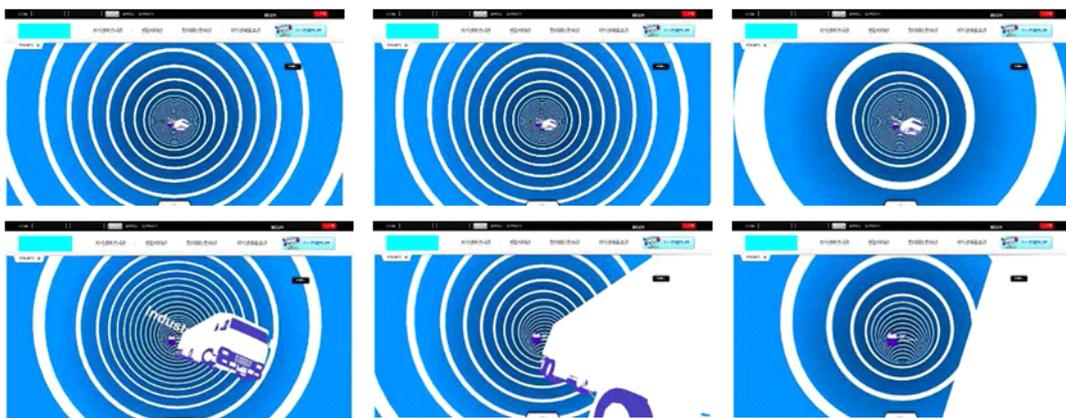
<http://www.w3.org/TR/2007/WD-UNDERSTANDING-WCAG20-20071211/seizure.html>

<http://www.w3.org/TR/2007/WD-UNDERSTANDING-WCAG20-20071211/seizure-three-times.html>

<http://www.w3.org/TR/2007/WD-UNDERSTANDING-WCAG20-20071211/seizure-does-not-violate.html>

잘못된 사례 38) 발작 가능성이 있는 동영상을 제공한 경우

다음 그림은 움직이는 영상을 플래시 콘텐츠로 제공하는 경우이다. 이 콘텐츠의 경우 파란색과 흰색 고리가 빠르게 교차되며 회전되어 확대되는 전체 화면에 광과민성 발작증상을 일으킬 가능성이 있다.



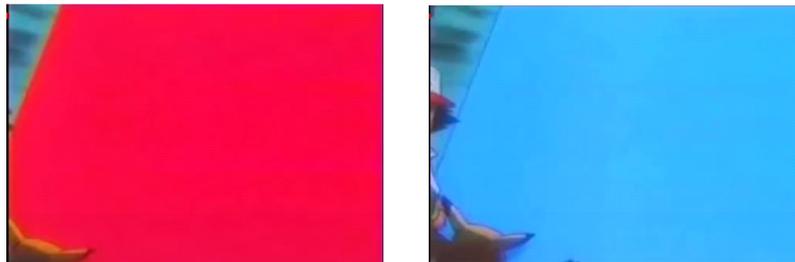
개선방법

많은 움직임이 있거나 화면의 변화가 빠른 콘텐츠를 제공하기 전에 사용자가 광과민성 증후에 의한 발작을 피할 수 있도록 사전에 경고해야 한다.

또한 사용자 요구에 의하여 이 콘텐츠를 사용하더라도 이용 도중 즉시 중단할 수 있는 기능을 제공해야 하며, 이 기능은 마우스, 키보드 등 어느 입력장치로도 접근할 수 있도록 제작되어야 한다.

잘못된 사례 39) 번쩍임이 지속되는 영상을 제공한 경우

빨간색과 파란색이 빠른 속도로 주기적인 반복이 일어나는 애니메이션, 동영상 등은 광과민성 증후군이 있는 사용자에게 발작 증세를 일으킬 수 있다. 다음 화면은 1997년 일본에서 어린이들에게 큰 위험을 제공한 포켓몬스터 제 38화의 문제 장면이다. 그림과 같이 빨간색과 파란색이 빠른 속도로 화면에 나타남으로 인하여 광과민성 증후군에 의한 발작 증세를 유발할 수 있다.



개선방법

채도 대비가 큰 화면의 반복적인 변화를 제공해야 하는 콘텐츠는 초당 3회 이상 깜빡이지 않도록 해야 한다. 또한 섬광 이미지가 지속되는 시간을 3초 미만으로 되도록 한다.

검사항목 2.4.1 (반복 영역 건너뛰기) 콘텐츠의 반복되는 영역은 건너뛸 수 있어야 한다.

□ 개요

웹 콘텐츠는 사용성을 높이기 위하여 각 페이지를 내비게이션 영역, 정보 제공 영역 등으로 구분하여 그 위치와 모양을 일정하게 유지한다. 이 중 내비게이션 영역은 동일한 콘텐츠를 모든 페이지에 걸쳐 반복적으로 제공하므로 반복 영역이라고도 한다. 내비게이션 영역은 주로 메뉴, 링크 모음 등으로 구성된다.

화면 낭독 프로그램 사용자의 입장에서는 웹 페이지를 리프레시하거나 리로딩 할 때마다 페이지의 앞부분에 제공되는 반복 영역을 반복적으로 듣게 된다. 따라서 자주 방문하는 웹 사이트의 경우 반복 영역을 반복해서 듣는 것이 매우 불편한 일이다. 이러한 불편을 해결하기 위하여 웹 콘텐츠는 반복 영역을 화면 낭독 프로그램이 읽어주기 전 또는 읽는 도중에 사용자가 핵심 콘텐츠로 이동할 수 있는 수단을 제공해야 한다.

□ 적용 기술

적용 기술 46) 반복적인 콘텐츠 영역의 시작 부분에 건너뛰기 링크 제공

콘텐츠 영역을 우회하는 건너뛰기 링크(skip navigation)를 제공하면 해당 영역을 건너뛰고 다음 영역으로 사용자 제어를 이동시킬 수 있으므로 키보드 사용자의 사용성을 높일 수 있다. 어떤 영역에 건너뛰기 링크를 제공하려면 그 영역의 첫 번째 링크의 목적지를 다음 영역의 첫 번째 항목으로 지정해야 한다.

링크, 버튼으로 구현된 건너뛰기 링크를 각 페이지 상단에 제공할 경우 반복 영역을 건너뛰어 원하는 정보로 이동할 수 있다. 건너뛰기 링크는 반복 영역이 있는 모든 페이지의 상단의 일정한 곳에 제공해야 하며 사용법과 모양이 모든 페이지에 걸쳐 동일해야 한다.

정보 제공 영역을 여러 개로 구분하여 다수의 콘텐츠를 제공하는 경우 여러 개

의 건너뛰기 링크를 제공할 수 있다. 그러나 건너뛰기 링크도 반복 영역의 하나이므로 여러 개의 건너뛰기 링크를 제공하면 화면 낭독 프로그램이 건너뛰기 링크를 읽어주어 사용자에게 또 다른 불편함을 주므로 그 수를 3개 이하로 줄일 필요가 있다. 다음과 같은 경우 건너뛰기 링크를 제공할 수 있다.

1) 링크 목록간 건너뛰기 기능 제공

하나의 웹 페이지에 여러 개의 링크 목록영역을 포함하고 있는 경우 각 링크 목록영역별로 건너뛰기 링크를 제공할 수 있다. 예를 들어, 사이트 맵, 주 메뉴, 기관 소개, 기관 연락처 등으로 구성된 영역과 소절(subsection)로 이동하기 위한 링크 목록을 제공하는 콘텐츠에서 페이지의 첫 부분에 '기관 소개로 이동' 링크를 제공하고, 두 번째 링크 목록의 첫 부분에는 '소절 건너뛰기'를 위한 링크를 제공한다.

2) 도서 색인

색인을 포함하여 구성된 온라인 도서 시스템에서 페이지의 좌측에 색인을 제공하는 경우 색인의 첫 번째 링크를 '색인 건너뛰기'로 구성하면 반복적으로 제공하는 색인영역을 우회할 수 있다.

3) 모든 링크 건너뛰기 기능 제공

어떤 기관의 웹 페이지 첫 부분이 메뉴, 내비게이션 바(navigation bar), 사이트 맵(site map), 기관에 대한 정보, 기관 연락처 등으로 구성되어 있을 경우 건너뛰기 링크를 메뉴 시작 전에 제공하고 링크 텍스트를 '모든 링크 건너뛰기'라고 제공한다. 건너뛰기 링크를 실행시키면 메뉴, 내비게이션 바, 사이트 맵을 우회하여 기관에 대한 정보 영역으로 직접 이동할 수 있다.

<참고>

모든 페이지가 반복 영역 건너뛰기가 반드시 필요하지는 않는다. 콘텐츠 블록마다 링크의 수가 많거나 메뉴나 링크 등의 반복되는 콘텐츠가 많은 복잡한 페이지일 경우에만 반복 영역 건너뛰기가 필요하다. 또한 반복 영역 건너뛰기 자체도 너무 많이 제공되지 않아야 한다.

적용 기술 47) 건너뛰기 링크의 화면 표시

건너뛰기 링크는 화면 낭독 프로그램을 사용하는 시각장애인뿐만 아니라 키보

드를 사용하는 사용자가 핵심 콘텐츠로 이동하기 위한 매우 유용한 수단이다. 따라서 건너뛰기 링크는 화면에 보이도록 구현하는 것이 좋다.

디자인의 특성상 건너뛰기 링크를 보이지 않도록 구현하더라도 키보드 내비게이션에 의하여 초점을 받을 수 있도록 해야 하며, 키보드 초점을 받으면 건너뛰기 링크가 표시되어야 한다.

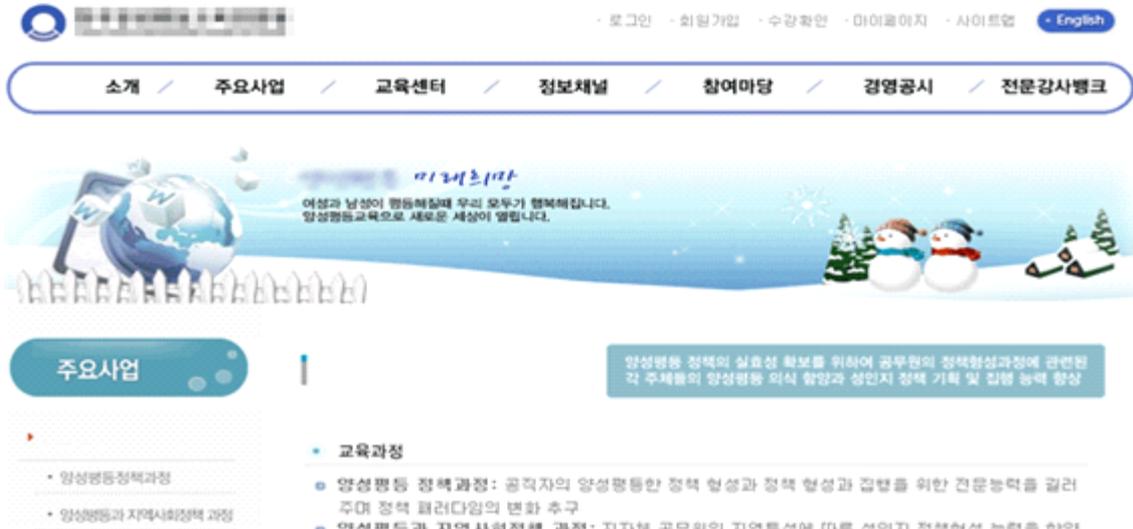
다음 예는 키보드 초점을 받으면 건너뛰기 링크가 화면에 나타나도록 구현한 CSS 코드이다. 건너뛰기 링크가 화면에 나타나는 CSS 파일을 사용자에게 추가로 제공하는 것도 좋은 방법이다.

```
<style>
.contents_nav_link{visibility:hidden;position:absolute;}
.contents_nav_link:focus{visibility:visible;position:static;}
</style>
<a href="#contents" class="contents_nav_link">본문 바로가기</a>
```

□ 잘못된 사례 및 개선방법

잘못된 사례 40) 건너뛰기 링크를 제공하지 않은 경우

다음 그림은 건너뛰기 링크가 제공되지 않아 본문의 핵심 콘텐츠로 바로 초점할 수 없는 경우이다. 이 경우 화면 낭독 프로그램을 이용하는 사용자나 몸이 불편하여 키보드만으로 접근할 수밖에 없는 사용자는 페이지 상단의 메뉴, 왼쪽의 주요 사업 링크 등 공통적으로 제공되는 반복되는 링크를 항상 지나가야 하기 때문에 불편하다.



주요 콘텐츠로 바로 이동이 불가능한 페이지

개선방법

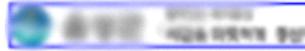
따라서 반복적으로 사용되는 상단 메뉴와 왼쪽 링크를 우회하여 주요 콘텐츠로 이동할 수 있는 링크나 버튼을 제공해야 한다. 또한 건너뛰기 링크(버튼)는 화면에 표시하여 키보드 사용자도 사용할 수 있도록 제공하는 것이 바람직하다.

`본문 바로가기`

잘못된 사례 41) 건너뛰기 링크를 과도하게 제공한 경우

다음 그림은 공통적인 사용이라 생각하여 복사, 붙여넣기 방식으로 건너뛰기 링크를 제공하거나 건너뛰기 링크를 너무 작은 단위로 나누어 제공한 경우다. 이 경우, 해당 페이지에 없는 기능을 제공하여 사용자에게 혼란을 줄 수 있고, 많은 양의 링크가 제공되어 이 영역이 오히려 반복영역으로 인식될 수 있는 문제가 있다.

- [콘텐츠 바로가기](#)
- [대메뉴 바로가기](#)
- [하위메뉴 바로가기](#)
- [사이트 정보 바로가기](#)
- [베너 바로가기](#)
- [페이지 만족도 평가 바로](#)



- [English](#)
- [中國語](#)
- [日本語](#)
- [아랍어](#)
- [문화관광](#)
- [홈페이지 도움말](#)
- [사이트맵](#)

과도한 건너뛰기 링크 제공

개선방법

따라서 건너뛰기 링크는 주요 콘텐츠를 이동하는 기능이나 반복되는 링크를 건너뛸 수 있는 목적으로만 제공하는 것이 좋다.

또한 핵심 콘텐츠에 한해서만 바로 이동할 수 있는 링크를 제공해야 한다.

<참고>

건너뛰기 링크의 용도 표현

건너뛰기 링크에 대한 대체 텍스트 또는 링크 텍스트는 다음 예와 같이 어떤 영역을 우회하는지 또는 목적지가 어디인지를 명확히 표시해야 한다.

- 1) "OO 링크 영역 건너뛰기"
- 2) "OO을 건너뛰어 OO으로 이동" 등

건너뛰기 링크를 제공할 경우

건너뛰기 링크를 제공할 경우 "OO 링크 집합을 건너뛰기" 또는 "OO을 건너뛰어 OO으로 감" 등의 방식으로 어떤 영역을 건너뛰는지 어디로 건너뛰는지에 대해 명확히 알려 주어야 한다.

검사항목 2.4.2 (제목 제공) 페이지, 프레임, 콘텐츠 블록에 는 적절한 제목을 제공해야 한다.

□ 개요

적절한 제목은 페이지, 프레임, 콘텐츠 블록을 유추하여 사용자가 웹 콘텐츠를 운용하기 쉽게 도와주기 위해 제공한다. 만약 페이지의 제목을 제공하지 않거나 모두 동일한 제목으로 통일될 경우 화면 낭독 프로그램을 이용하는 사용자나 인지능력이 낮은 사용자는 페이지가 변경되어도 어떤 페이지인지 판단하기 어렵게 된다.

<참고> 적절한 제목이란?

제목은 불필요한 특수기호를 반복적으로 사용하지 말아야 하고 간단명료해야 한다. 그리고 해당 웹 사이트의 어느 페이지와도 다른 제목이어야 한다. 프레임의 경우 아무런 내용이 없는 프레임이라면 "빈 프레임" 등으로 제목을 제공해야 하고, 콘텐츠 블록에 대하여 <h1>, <h2> 태그 등을 부여하여 제목과 본문을 구분할 수 있도록 제공하는 것이 좋다.

□ 적용 기술

적용 기술 48) 웹 페이지의 제목 제공

모든 HTML(XHTML 포함)을 이용하여 구성된 웹 페이지에는 문서의 목적을 정의하는 title 태그를 반드시 제공해야 한다. title 태그를 제공하면 본문을 검색하지 않아도 문서의 용도를 직관적으로 알 수 있다.

다음 코드는 페이지 제목을 'World Wide Web'으로 정의한 예다.

```
<html>
  <head>
    <title>world wide web</title>
  </head>
  <body>
    ...
  </body>
</html>
```

적용 기술 49) 콘텐츠 블록의 제목 제공

헤딩 태그는 콘텐츠의 구조를 조직적으로 전달하기 위하여 사용한다. 따라서 사용자는 헤딩 태그를 이용해 웹 문서의 구조를 파악하거나 다음 헤딩에 대한 정보를 쉽게 파악할 수 있다.

```
<h2>뉴스</h2>
  <div>
    <ul>...</ul>
    <ul>...</ul>
  </div>
<h2>쇼핑</h2>
  <div>
    <ul>...</ul>
    <ul>...</ul>
  </div>
```

만일 화면 상으로 표시되는 모습을 헤딩 태그를 사용하여 구현한 것처럼 꾸민다면 사용자가 문서의 구조를 파악하거나 콘텐츠를 탐색하는 것이 어려워 혼란을 줄 수 있다. 따라서 글씨체 속성을 이용하여 콘텐츠의 구조를 표현하는 것은 좋은 방법이 아니다.

적용 기술 50) 프레임 제목 제공

frame 태그를 사용할 경우 해당 frame의 title 속성을 제공하여 프레임 간의 내비게이션을 용이하도록 해야 한다.

웹 페이지를 frameset으로 구성하고 title 태그를 제공하더라도 각 frame별로 title 속성을 제공해야 한다. 그 이유는 프레임의 내비게이션 시 title 태그가 현재 프레임의 위치를 알려주는 중요한 정보로 이용되기 때문이다.

<참고> 프레임 제목

프레임 제목은 프레임 내에 포함된 콘텐츠를 유추할 수 있는 간결한 제목을 제공해야 한다. "메

인 메뉴", "참고 문서" 등의 프레임 제목은 프레임이 제공하는 콘텐츠의 용도를 유추할 수 있으므로 적절하다. 그러나 "top 프레임", "main 프레임" 등과 같은 제목은 프레임의 용도가 아닌 프레임의 위치 정보를 알려줄 뿐이므로 바람직하지 않다.

□ 잘못된 사례 및 개선방법

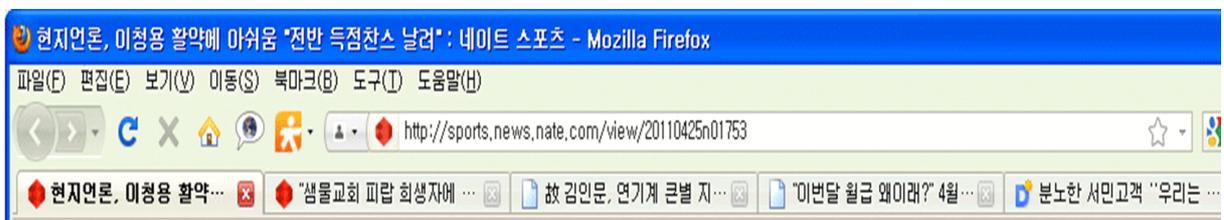
잘못된 사례 42) 콘텐츠가 다른 페이지에 동일한 제목을 제공한 경우

다음 예는 페이지마다 콘텐츠가 다르지만 페이지 title 태그 내용을 동일하게 제공하는 경우로, 이 경우 화면 낭독 프로그램을 이용하는 사용자나 인지장애를 가진 사용자는 접근하는 페이지의 변화된 부분을 인식하기 어렵게 된다.



개선방법

페이지마다 각각 고유한 title 태그 내용을 제공해야 한다. 신문사 사이트라면 다음 그림처럼 신문기사의 중요 제목을 title 태그 내용으로 제공해 주는 것이 좋다.



잘못된 사례 43) 프레임 제목이 누락된 경우

프레임에 타이틀을 제공하지 않은 경우 화면 낭독 프로그램을 이용하는 사용자

나 인지장애를 가진 사용자는 프레임을 구별할 수 없으므로 프레임 간의 이동이 어려워 결과적으로 해당 콘텐츠를 운용하기 어려워진다.

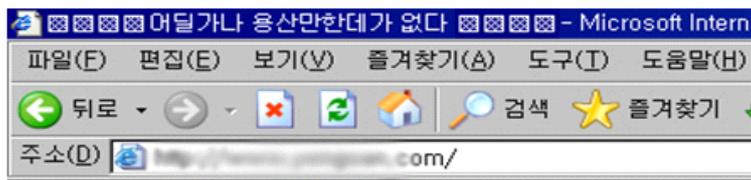
개선방법

프레임 제공 시 프레임 내에서 제공하는 콘텐츠의 용도 또는 목적을 토대로 간결한 내용으로 제목을 제공해야 한다. 또한 "top 프레임", "main 프레임" 등과 같이 어떤 콘텐츠인지 알 수 없는 내용보다는 "로그인 프레임", "광고 프레임" 등과 같이 그 의미를 명확히 알 수 있는 제목을 제공하면 페이지를 이용하기가 편리하게 된다.

```
<iframe width="280" height="150" frameborder="0" scrolling="no" marginwidth="0" marginheight="0" title="광고 프레임" src="http://nv2.ad.naver.com/adshow?unit=002X" name="f280150" id="f280150">광고 : <a href="http://nv2.ad.naver.com/adshow?unit=002X">http://nv2.ad.naver.com/adshow?unit=002X</a></iframe>
```

잘못된 사례 44) 특수문자(☒☒☒☒)기호를 제공한 경우

다음 예는 페이지 타이틀에 특수문자(☒☒☒☒)를 사용하여 제공하는 경우로, 이 경우 화면 낭독 프로그램을 이용하는 사용자나 인지장애를 가진 사용자는 페이지 제목을 명확하게 이해하지 못할 수 있다. 때에 따라서는 반복적인 특수문자로 인하여 화면 낭독 프로그램의 고장을 유발할 수도 있으므로 주의해야 한다.



페이지 타이틀에 특수문자 사용

개선방법

페이지 타이틀 제공 시 표현을 위한 특수문자의 사용은 피하고 간결하고 웹 페이지가 제공하는 콘텐츠를 유추할 수 있도록 제공해야 한다.

검사항목 2.4.3 (적절한 링크 텍스트) 링크 텍스트는 용도나 목적을 이해할 수 있도록 제공해야 한다.

□ 개요

링크 텍스트를 "여기를 클릭하세요."나 "여기"와 같은 애매모호한 표현을 사용하면 시각장애인이나 인지장애인 뿐만 아니라 비장애인들도 링크를 클릭했을 때 어떤 반응이나 어떤 페이지로 이동될 지 예측하기 어렵다.

따라서 링크 텍스트를 직관적이고 의미 있게 구성하여 사용자가 링크의 용도나 목적지를 명확히 알 수 있게 제공해야 한다.

올바른 링크 텍스트 제공은 사용자가 원하지 않는 링크를 방문해 보지 않거나 키보드를 여러 번 입력하지 않아도 되고 원하는 콘텐츠가 있는 링크를 바로 이용할 수 있게 된다.

□ 적용 기술

적용 기술 51) 링크의 목적을 설명하는 링크 텍스트 제공

링크를 포함하는 주변 문장과 조합하여 그 목적이나 용도를 알 수 있도록 제공할 수 있다. 이러한 방법은 사용자로 하여금 이 링크가 어떤 목적을 가지는지 어떤 페이지로 이동하는지를 구별할 수 있도록 도와준다. 링크 텍스트를 단순히 URL 경로를 제공하는 것은 사용자에게 충분한 설명을 하지 못한다.

1) 메이플라워호에 대한 웹 페이지로 이동하는 링크

```
<p>"미국으로의 첫 번째 이주민들은 <a href="mayflower.html">메이플라워호</a>를 타고 왔다."</p>
```

2) 기사에 링크를 걸 경우

다음 예와 같이 해당 링크가 사용되는 웹 페이지의 문맥에 따라 목적에 맞는

단어에 링크를 걸어주는 경우도 있다.

보도내용 링크

<p>"00신문의 보도 내용에 따르면 00교육청에서는 2012년도 연간 계획을 수립하였다."</p>

00교육청 링크

<p>"00신문의 보도 내용에 따르면 00교육청에서는 2012년도 연간 계획을 수립하였다."</p>

2012년도 연간계획 링크

<p>"00신문의 보도 내용에 따르면 00교육청에서는 2012년도 연간 계획을 수립하였다."</p>

<참고> 링크의 목적을 설명하는 링크 텍스트 제공

링크의 목적을 설명하는 링크 텍스트를 제공하면 사용자는 링크를 웹 페이지의 다른 링크와 구별할 수 있고, 그 링크를 따라갈지의 여부를 결정하는 데 도움을 준다.

목적지의 URL 주소를 링크 텍스트로 제공하는 것은 충분한 설명이 되지 못한다.

□ 잘못된 사례 및 개선 방법

잘못된 사례 45) 목적이나 용도를 알기 어려운 링크 텍스트

다음 예는 페이지 이동을 하기 위한 링크를 텍스트로 제공한 경우로, 이 경우 해당 사이트에 대한 목적이나 링크의 용도를 알기 힘들어 어떠한 페이지인지 어떠한 기능을 하는 링크인지를 클릭해서 해당 링크 페이지로 이동하기 전까지는 알 수 없게 된다.

그 밖의 것들

- 다른 테스트를 하고 싶으시다면 [여기를 클릭해주세요](#).
- 기타 문의 사항이 있으시면 [여기를 클릭해주세요](mailto:mltaylor@us.ibm.com).

<그림 26> 목적이나 용도를 알기 어려운 링크 텍스트 (여기를 클릭해주세요)

개선방법

링크 텍스트를 직관적이고 의미 있게 구성하여 사용자가 링크의 용도나 목적지를 명확히 알 수 있도록 개선해야 한다.

또한 위 예제에서는 "여기를 클릭해 주세요."보다는 "다른 테스트를 하고 싶으시다면 클릭해 주세요." 라는 문장 전체를 링크 텍스트로 사용하는 것이 좋다. 이외 추가 설명이 필요하다면 title 속성을 사용하여 제공할 수 있다.

- title 속성을 사용하는 사례

```
<div class="page">  
  
<a href="page_prev.php?page=1" class="pre" title="이전 검색결과 페이지">이전</a>  
  
<a href="page_next.php?page=3" class="next" title="다음 검색결과 페이지">다음</a>  
  
</div>
```

원칙 3. 이해의 용이성(Understandable): 콘텐츠는 이해할 수 있어야 한다.

□ 이해의 용이성

웹에서 제공되는 정보와 사용자 인터페이스는 이해할 수 있게 구성이 되어야 한다. 여기에서 이해라는 것은 자극으로부터 시각과 청각 등 여러 감각 기관을 통해 감각을 느끼고, 사물을 지각(perception)한 후에 과거의 경험 등에 견주어 내용을 파악하고 해석하는 것을 말한다. 원칙 1(인식의 용이성)에서는 장애인들이 웹 콘텐츠를 일차적으로 잘 감각하고 지각할 수 있도록 작성하는 방법에 대한 지침을 담고 있는 반면, 원칙 3(이해의 용이성)에서는 그렇게 지각된 웹 콘텐츠가 실제 어떤 내용이고, 어떤 의미를 담고 있는지 이해할 수 있도록 작성하는 방법을 담고 있다. 즉, 웹 콘텐츠나 인터페이스가 감각적으로는 인식이 되었으나 무슨 내용인지 알 수가 없게 다른 언어로 되어 있거나 순서가 뒤죽박죽이어서 이해가 안 되거나 무엇을 해야 할 지 안내가 충분하지 않을 수 있다. 이러한 것을 방지하기 위해서 원칙 3(이해의 용이성)에서는 네 가지 지침과 여섯 개의 검사항목을 제시하고 있다.

우선, 제공되는 내용이 사용자의 이해 수준이나 사용 언어에 구애받지 않고 쉽게 읽을 수 있어야 한다. 이를 위해서는 우선 제공되는 정보의 언어가 무엇인지 명확히 표시하여 시각적으로 그것을 보는 사람뿐만 아니라 보조 기기나 검색 엔진 등 다른 방법으로 웹에 접근하는 경우에도 내용을 이해할 수 있어야 한다.

또한 제공되는 내용이 시각적으로는 논리적인 것 같지만, 사실 그것을 표시하기 위해 사용된 소스가 논리적인 순서로 작성되지 않았다면 제공되는 내용을 순차적으로 읽어나가는 시각장애인용 화면 낭독 프로그램에서 내용을 이해하기 어렵게 된다. 마찬가지로 복잡한 데이터를 표 형식으로 제공할 때 그 표를 풀어 헤치고 데이터를 순서대로 나열해도 여전히 내용을 이해할 수 있는지 생각해 보아야 한다.

사용자 인터페이스를 설계할 때에도 그것이 이해할 수 있는지 점검해야 한다. 예를 들어, 사용자로부터 입력을 받는 서식을 만들 때 해당 서식이 무엇을 입력해야 하는 것인지 알 수 있도록 레이블을 잘 붙여주어야 한다. 또한 사용자가 값을 실수로 잘못 입력했을 경우에도, 그것을 쉽게 고칠 수 있는 방법을 제공해야 한다.

□ 관련 지침

이해의 용이성은 사용자가 장애유무 등에 관계없이 웹 사이트에서 제공하는 콘텐츠를 이해할 수 있도록 제공하는 것을 의미한다. 이해의 용이성은 가독성, 예측 가능성, 콘텐츠의 논리성, 입력 도움의 4가지 지침으로 구성되어 있다.

<표 8> 이해의 용이성 관련 지침 및 검사항목

지침(4개)	검사항목(6개)
3.1(가독성) 콘텐츠는 읽고 이해하기 쉬워야 한다.	3.1.1(기본 언어 표시) 주로 사용하는 언어를 명시해야 한다.
3.2(예측 가능성) 콘텐츠의 기능과 실행결과는 예측 가능해야 한다.	3.2.1(사용자 요구에 따른 실행) 사용자가 의도하지 않은 기능(새 창, 초점 변화 등)은 실행되지 않아야 한다.
3.3(콘텐츠의 논리성) 콘텐츠는 논리적으로 구성해야 한다.	3.3.1(콘텐츠의 선형화) 콘텐츠는 논리적인 순서로 제공해야 한다.
	3.3.2(표의 구성) 표는 이해하기 쉽게 구성해야 한다.
3.4(입력 도움) 입력 오류를 방지하거나 정정할 수 있어야 한다.	3.4.1(레이블 제공) 입력 서식에는 대응하는 레이블을 제공해야 한다.
	3.4.2(오류 정정) 입력 오류를 정정할 수 있는 방법을 제공해야 한다.

검사항목 3.1.1 (기본 언어 표시) 주로 사용하는 언어를 명시해야 한다.

□ 개요

웹 브라우저는 웹 페이지를 구성하는 텍스트 콘텐츠의 언어 정보를 바탕으로 텍스트 콘텐츠를 화면에 표시하거나 보조기기로 제공한다. 화면 낭독 프로그램을 사용하는 경우, 텍스트 콘텐츠의 언어 정보를 화면 낭독 프로그램으로 제공하여 정확한 발음이 가능하도록 화면 낭독 프로그램을 제어하기도 한다. 따라서 웹 페이지를 구성하는 기본 언어는 정확히 명시해야 한다.

□ 적용 기술

적용 기술 52) <html> 태그에 주 사용 언어 지정

웹 페이지에서 주로 사용하는 언어는 페이지의 상단에 html 태그에 lang 속성을 이용하여 다음과 같이 지정한다.

주로 사용되는 언어가 영어일 때

```
<html lang="en">
```

주로 사용되는 언어가 한국어일 때

```
<html lang="ko">
```

주로 사용되는 언어가 일본어일 때

```
<html lang="ja">
```

주로 사용되는 언어가 스페인어일 때

```
<html lang="es">
```

이렇게 페이지에서 주로 사용하는 언어를 지정해 놓으면, 웹 페이지에 접근하는 여러 가지 기계에서 언어를 인식하게 된다. 화면 낭독 프로그램과 같은 보조기기는 지정된 언어에 맞는 TTS(Text-to-Speech) 엔진을 가동시키고, 자동으로 웹의 정보를 수집하여 검색을 도와주는 검색 엔진 로봇은 해당 언어의 검색 인덱스에 페이지를 분류해 넣는다. 즉, 사용자가 검색할 때 특정 언어를 지정한 웹 검색 결과만 나오도록 할 때 검색 엔진은 lang 속성 값을 여러 가지 판단의 근거 중 하나로 활용할 수 있다.

이 때 lang의 속성값에는 ISO 639-1에서 지정한 두 글자로 된 언어 코드를 넣어야 한다. 주의할 점은 한국어 코드는 kr이 아니라 ko다. 자세한 것은 ISO 639-1에 대한 위키 페이지(http://en.wikipedia.org/wiki/ISO_639-1)와 언어 코드 목록(http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)을 참고하라.

여기에서 주의할 것은 주로 사용되는 언어를 설정하는 것은 해당 페이지의 인코딩 방식을 정하는 것과는 다르다는 것이다. 즉, <head>의 맨 첫 줄에 <title>을 지정하기 전에 <meta> 태그를 이용해서 문서의 인코딩 방식을 지정하지 않으면 브라우저가 콘텐츠를 바르게 표시하지 못하는 경우가 생긴다. 인코딩 방식을 지정하는 문제는 화면 낭독 프로그램과 같은 보조기기에서 뿐만 아니라 시각적으로 내용을 표시하는 브라우저에도 영향을 주므로 반드시 제공해야 한다.

- HTML 4.01의 경우,

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>페이지의 제목</title>
```

- HTML5의 경우,

```
<head>
  <meta charset="UTF-8">
  <title>페이지의 제목</title>
```

와 같이 인코딩 방식을 지정한다.

<참고> 페이지 중간에 언어가 바뀔 때

지침에서는 해당 페이지의 주요 사용 언어만을 최소한으로 지정할 것을 요구하고 있다. 이렇게 함으로써, 화면 낭독 프로그램이 페이지의 내용을 해당 언어로 정확하게 읽어주도록 지시하는 역할을 한다. 그러나 페이지 안에서 여러 개의 언어가 번갈아가면서 쓰이는 경우, 변경되는 언어가 나올 때마다 언어를 지정해 주는 것이 좋다.

<p>중국어로는 "안녕하세요"라고 인사할 때 你好라고 말합니다.</p>

만약 화면 낭독 프로그램에 중국어 음성 엔진이 탑재되어 있다면 위의 你好를 "니하오"와 같은 중국어 발음으로 읽어주겠지만, lang 속성을 지정하지 않았다면 그냥 한국식 발음으로 "이호"라고 읽어주게 된다.

주로 사용하는 언어 또는 부분적으로 쓰이는 언어를 바르게 지정함으로써 해당 언어에 맞는 인용 부호나 구두점 등을 바르게 표시할 수 있다.

<p>그는 한국어로 <q lang="ko">웹은 보편적으로 접근 가능한 것입니다.</q>라고 말하였다.

그는 독일어로 <q lang="de">Web ist allgemein zugänglich</q>이라고 말하였다.

그는 프랑스어로 <q lang="fr">Web est accessible à tous</q>이라고 말하였다.</p>

위와 같이 해당 언어가 사용된 부분에 부분적으로 또는 페이지 전체적으로 언어를 명시해 주면, 브라우저는 다음과 같이 화면에 표시해 준다.

그는 한국어로 "웹은 보편적으로 접근 가능한 것입니다."라고 말하였다.

그는 독일어로 „Web ist allgemein zugänglich"이라고 말하였다.

그는 프랑스어로 « Web est accessible à tous »이라고 말하였다.

검사항목 3.2.1 (사용자 요구에 따른 실행) 사용자가 의도하지 않은 기능(새 창, 초점 변화 등)은 실행되지 않아야 한다.

□ 개요

서식 컨트롤이나 사용자 입력이 초점을 받았을 때 의도하지 않은 기능이 자동적으로 실행되지 않도록 콘텐츠를 개발해야 한다. 사용자가 마우스로 클릭하거나 키보드를 이용하여 입력한 후 기능이 실행되어야 하며, 사용자가 예측할 수 없는 상황에서 정보를 제공하지 않아야 한다. 특히 사용자가 인지하지 못한 상황에서 새 창, 팝업 창 등을 제공하지 않아야 한다.

□ 적용 기술

적용 기술 53) 초점 변화에 의해 맥락을 변화시키지 말고, 해당 인터페이스가 활성화(activate)되었을 때 기능을 실행

보통 사용자는 해당 요소를 클릭하거나, 키보드의 엔터키를 누름으로서 기능이 실행될 것으로 예측한다. 즉, 단지 키보드의 초점을 가져가거나 마우스의 포인터를 위치시켰다고해서 급격한 변화가 일어나리라고 기대하지 않는다. 사용자의 편의성을 높여주고 마우스 클릭 수를 줄여준다는 목적으로 해당 인터페이스가 초점을 받았을 때 급격한 맥락의 변화를 초래하는 기능을 수행하면 일반 사용자들도 혼란스러울 뿐만 아니라 화면 확대기나 키보드를 위주로 사용하는 장애인들, 한번의 클릭이 초점의 이동을 뜻하는 모바일 브라우저 사용자들의 웹 페이지 탐색을 어렵게 만든다. 다음과 같은 예를 생각해 볼 수 있다.

- 버튼과 새 창을 여는 기능을 연계시킬 경우, 사용자가 버튼을 클릭하거나 또는 스페이스 바를 눌렀을 경우에만 새 창을 연다. 그렇지 않고 버튼에 onfocus 속성을 이용해 새 창을 여는 이벤트를 연계하면 안 된다.
- 여러 개의 페이지로 구성된 입력 서식이 있다고 가정하면, 첫 번째 입력 페이지에서 입력을 완료하고 [다음 페이지] 버튼 또는 [첫 번째 페이지 입력

완료]라는 버튼으로 탭을 이동하자마자 다음 페이지로 자동으로 가도록 하면 안 된다. 반드시 사용자가 해당 버튼을 눌렀을 때에만 두 번째 페이지로 이동해야 한다.

적용 기술 54) 명확한 서식 제출(submit) 버튼 제공

서식을 입력하고 나서 사용자의 명확한 요구와 값 제출에 의해서만 HTTP 요구(request)가 일어나고 맥락의 변화가 일어나도록 한다. HTML에서 제공하는 submit 버튼을 쓰지 않고, 자바스크립트로만 서식값이 전송되도록 하는 것은 잘못이다. 다음과 같이 명확한 서식값 제출 버튼을 제공하는 것이 좋다.

```
<form action="http://www.example.com/cgi/subscribe/" method="post">
  <p>메일링 리스트에 가입하려면 이메일 주소를 넣어주세요.</p>
  <label for="address">이메일 주소:</label><input type="text"
  id="address" name="address" />
  <input type="submit" value="가입" />
</form>
```

마찬가지로, 목록 선택 상자를 제공할 경우 반드시 선택값을 사용자가 확인하고 제출할 수 있는 submit 버튼을 따로 제공해야 한다.

```
<form action="http://www.example.com/cgi/redirect/" method="get">
  <p>이동할 사이트를 고르세요.</p>
  <select name="dest">
    <option value="/index.html">홈</option>
    <option value="/blog/index.html">블로그</option>
    <option value="/tutorials/index.html">사용법</option>
    <option value="/search.html">검색</option>
  </select>
  <input type="submit" value="페이지 이동" />
</form>
```

적용 기술 55) 새 창 열림을 사전에 알림

링크를 선택했을 때 아무런 경고 없이 새 창이 열리게 되면, 시각적인 콘텐츠를 잘 인식하지 못하는 사용자들이나 인지 장애 또는 학습 장애인들에게 혼란을 주게 된다. 또한 일반 사용자들은 새 창이 열림으로써 앞으로 가기와 뒤로 가기와 같은 일반적인 브라우저의 이동 버튼을 사용할 수 없게 만드므로 되도록이면 사

용하지 않는 것이 좋다. 그러나 불가피하게 링크를 새 창으로 열어야 한다면 미리 새 창이 열릴 것이라는 것을 예측할 수 있게 해주는 것이 좋다. 가장 간결하고 쉬운 방법은 html의 링크 텍스트 안(이미지의 경우, alt 속성 값)에 새 창이 열릴 것이라고 표시해주거나, title 속성을 이용해 툴 팁(화면 낭독 프로그램의 경우, title 속성 값을 읽어주거나 읽어주지 않도록 설정이 가능함)으로 새 창이 열릴 것을 알려주는 방법이 있다.

```
<a href="knitting.html" target="_blank">뜨개질 포털 사이트
  (새 창으로 열림)</a>

<a href="korean.html" target="_blank"></a>

<a href="help.html" target="_blank" title="새 창으로 열림">도움말</a>
```

CSS를 이용해 해당 링크에 초점이 이동하거나 마우스 포인터가 위치했을 때 새 창이 열릴 것이라고 알려주는 방법도 있다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>새 창이 열리는 것에 대한 경고</title>
    <style type="text/css">
      body {
        margin-left:2em;
        margin-right:2em;
      }
      :focus { outline: 0; }
      a.info {
        position:relative;
        z-index:24;
        background-color:#ccc;
        color:#000;
        text-decoration:none
      }
      a.info:hover, a.info:focus, a.info:active {
        z-index:25;
        background-color:#ff0
      }
      a.info span {
        position: absolute;
        left: -9000px;
```

```

width: 0;
overflow: hidden;
}
a.info:hover span, a.info:focus span, a.info:active span {
display:block;
position:absolute;
top:1em; left:1em; width:12em;
border:1px solid #0cf;
background-color:#cff;
color:#000;
text-align: center
}
div.example {
margin-left: 5em;
}
</style>
</head>
<body>
<h1>새 창 열림 경고</h1>
<p> 더 자세한 정보는 <a class="info"
href="popup_advisory_technique.html" target="_blank">
<strong>적용 기술 웹 사이트</strong><span>새 창에 열림.</span></a>에 있습니다.
</p>
</body>
</html>

```

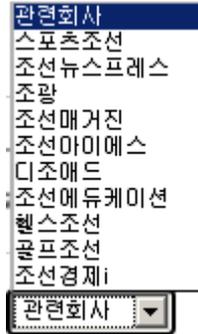
새 창 열림 경고

더 자세한 정보는 **적용 기술 웹사이트**에 있습니다.
새 창에 열림.

CSS를 이용해 새 창이 열리는 것을 미리 알려주는 경우

□ 잘못된 사례 및 개선 방법

잘못된 사례 46) 목록 선택 상자에서 초점 변경만으로 새 창이 열리는 경우



이동 버튼이 없는 목록 선택 상자

```
<select onchange="copyrtChgUrl(this.value)">
  <option value="0">관련회사</option>
  <option value="http://www.chogwang.com/">조광</option>
  <option value="http://golf.chosun.com/">골프조선</option>
</select>
```

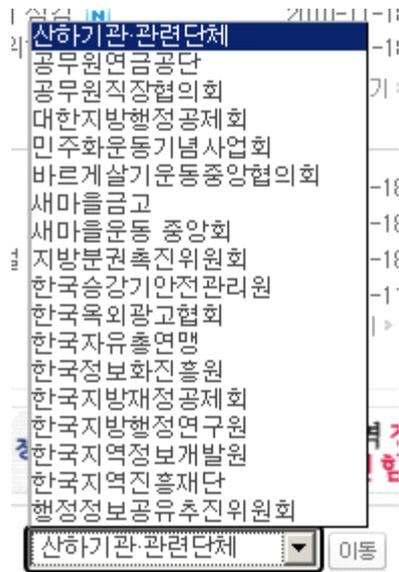
2011년 11월 21일 <http://www.chosun.com> 에서 발췌

위의 예에서는 이동하고자 하는 관련 회사를 선택하자마자 자바스크립트 함수가 실행되고 새 창에 해당 사이트가 열린다. 이 경우, 시각장애인 및 지체장애인 등 마우스를 이용하기 힘든 장애인뿐만 아니라 마우스에 능숙하지 않은 비장애인들도 자신의 선택을 되돌릴 수 없어서 문제가 된다. 키보드를 사용할 경우, 선택을 포기하고 다음으로 이동하기 위해 탭 키를 누르는 것만으로 의도하지 않게 선택이 이루어지게 된다. 화면 낭독 프로그램을 사용할 경우 첫 번째 항목을 읽어주는 것 자체가 바로 선택이 되어 다음 목록을 선택할 수 없게 된다. 따라서 선택 상자에서 여러 개의 항목 중에 하나를 선택한 다음에는 반드시 그 선택을 확인하는 버튼이나 절차가 있어야 한다.

개선방법

```
<form action="goto_site.jsp">
<select onchange="copyrtChgUrl(this.value)">
  <option value="0">관련회사</option>
  <option value="http://www.chogwang.com/">조광</option>
  <option value="http://golf.chosun.com/">골프조선</option>
</select>
<input type="submit" value="이동">
```

목록 선택 후 별도의 [이동] 버튼을 제공하여 개선한 사례



이동 버튼이 있는 목록 선택 상자

```

<form name="selectForm1" action="/gpms/view/jsp/goUrl.jsp"
onsubmit="goSelectUrl(1);return false;">
<label for="select1" style="display:none;">산하기관·관련단체</label>
<select name="url" id="select1" title="산하기관 및 관련단체 선택">
  <option value='' selected >산하기관·관련단체</option>
  <option value='http://www.gepcor.or.kr' >공무원연금공단</option>
  <option value='http://www.i-mopas.or.kr' >공무원직장협의회</option>
</select>
<input type="image" src="/gpms/resource/images/common/general/btn_go.gif"
alt="이동" title="새 창으로 이동">
</form>

```

2010년 11월 21일 <http://mopas.go.kr> 에서 발췌

위의 예에서는 산하 기관 및 관련 단체를 선택한 이후 이동 버튼을 눌러 최종 선택을 결정하게 되므로, 모든 항목을 선택할 수 있고, 실수로 선택을 잘못 했다 라도 되돌릴 수 있게 된다. 이동 버튼의 경우도 <form> 요소의 action 속성 값을 이용해 서버측에서 이동에 대한 스크립트를 제공하는 것이 바람직하다. 보조적으로 onsubmit이라는 이벤트에 대해서 자바스크립트 함수로 똑같은 기능을 제공해도 된다.

잘못된 사례 47) 체크 상자의 선택만으로 페이지가 다시 로드되는 경우

쇼핑몰 사이트에서 구매하고자 하는 물건의 여러 가지 옵션을 조합해서 구매의 범위를 좁히고자 할 때, 체크 상자를 다 선택하기 전에 하나의 체크 상자를 체크하는 것만으로 페이지 내용이 급격히 바뀌면 안 된다.



체크 상자를 선택하는 것만으로 페이지의 내용이 바뀌는 경우

개선 전 소스

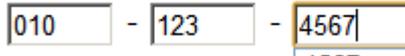
```
<input id='d1_438' item_id='438' item_name='로지텍' section_type='1'
section_id='1' onclick="cs.click_item( this.id );" type="checkbox" />
<label for='d1_438' style='cursor:pointer;'
onclick='event.cancelBubble=true'><span class='checked_on'>로지텍</span></label>
```

위의 페이지에서 로지텍이라는 제조사를 선택하자마자 바로 로지텍 마우스의 목록이 다음에 나오게 된다. 만약 키보드를 사용하는 시각장애인이 사실은 로지텍과 마이크로소프트, 3M을 복수로 선택하고 싶었다고 하자. 첫 번째 제조사를 선택하자마자 확인 과정 없이 바로 페이지가 다시 로드되어, 키보드를 사용하는 사람은 페이지의 맨 위에서부터 다시 탐색을 시작해서 제조사 선택하는 곳에 이르는 과정을 여러 번 반복해야 한다. 따라서 서식 값을 별도로 제출(submit)하지 않고 자동으로 내용이 급격히 바뀌는 것이 편해 보일 수 있지만, 일부 사용자들에게는 엄청난 불편함을 초래하게 된다는 것을 알아야 한다.

개선방법

체크 상자를 다 선택한 후 별도의 [확인]이나 [제출] 버튼을 누르도록 하여, 사용자의 선택이 전송된 후 상품 목록이 화면에 나오도록 한다.

잘못된 사례 48) 텍스트 입력 서식에 값을 넣으면 자동으로 제출되는 경우



전화 번호 입력 필드 예

개선 전 소스

```
<form method="get" id="form1">
  <input type="text" name="text1" size="3" maxlength="3"> -
  <input type="text" name="text2" size="3" maxlength="3"> -
  <input type="text" name="text3" size="4" maxlength="4"
  onchange="form1.submit();">
</form>
```

2010년 11월 20일 <http://www.w3.org/TR/2008/WD-WCAG20-TECHS-20081103/F36>에서 발췌

전화번호와 같이 텍스트 입력을 받을 때에도 입력 후 별도의 [제출], [확인]과 같은 버튼을 제공해야 한다. 그렇지 않고 위의 예처럼 마지막 번호를 입력하자마자 값이 제출이 되어버리는 경우, 입력 오류를 수정하기가 매우 어려워진다.

개선방법

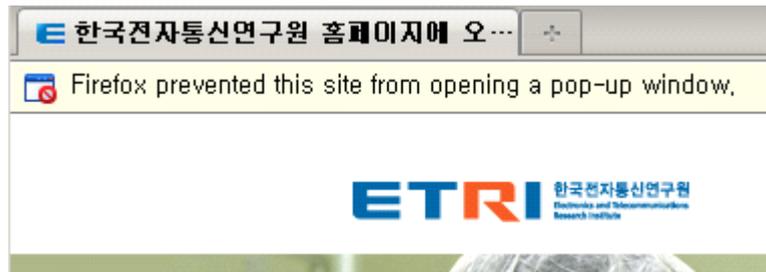
마지막 필드 값에 변화가 생길 때 자동으로 값이 제출되는 스크립트를 제거하고 다음과 같이 별도의 [제출] 버튼을 두고 서식 값을 제출하도록 한다.

```
<form method="get" id="form1" action="http://example.com/handle">
  <input type="text" name="text1" size="3" maxlength="3"> -
  <input type="text" name="text2" size="3" maxlength="3"> -
  <input type="text" name="text3" size="4" maxlength="4">
  <input type="submit" value="Submit" />
</form>
```

잘못된 사례 49) 페이지가 로드될 때 자동으로 팝업 창이 열리도록 한 경우

국내 웹 페이지들에서는 한시적인 사안을 홍보하거나 전달할 때 사용자가 원하지 않는 팝업 창을 많이 사용한다. 그러나 예측되지 않은 팝업은 사용자를 혼란스럽게 할 뿐만 아니라 시각장애인, 인지 능력이 떨어지는 사람, 운동 능력이 떨어지는 사람들에게 웹 탐색을 어렵게 만드는 요인이 된다. 또한 대부분의 브라우저

들이 팝업 창을 차단 옵션을 가지고 있어서 실질적인 광고, 홍보 효과도 떨어진 다.



브라우저에 의해 차단된 팝업 창

개선 전 소스

```
if ( notice_getCookie("CookieName5") != "done" ) {  
  
window.open('http://poll.kofst.or.kr:8080/Email_poll_101015_pop.aspx', 'CookieName  
5', 'status=no,width=480, height=320,resizable=no,left=0,top=215,scrollbars=no')  
  
}
```

개선방법

팝업 창을 쓰지 않고 원래의 메인 페이지 안에서 중요한 내용을 방문자들이 잘 볼 수 있도록 눈에 잘 띄거나 중요한 위치에 배치함으로써 실질적인 홍보 효과와 접근성을 높일 수 있다. 웹 페이지 기획자라면 팝업 창으로 띄우는 내용이 정말로 긴급하게 모든 방문자에게 알려야 하는 것인지 다시 고려해 볼 필요가 있다.

검사항목 3.3.1 (콘텐츠의 선형화) 콘텐츠는 논리적인 순서로 제공해야 한다.

□ 개요

웹 페이지에 제공되는 콘텐츠는 시각적으로 보았을 때 논리적인 순서로 보이지만, 그것을 선형적으로 풀어보면 논리가 어긋나지는 않은지 주의해야 한다. 화면 낭독 프로그램과 같은 보조 기기는 전체 콘텐츠를 보통 소스 코드 순서대로 풀어서 접근하게 된다. 이x 때 시각적으로 보였던 순서와 달리 논리적인 순서가 흐트러질 경우, 보조 기기 사용자들은 내용을 이해할 수 없게 된다.

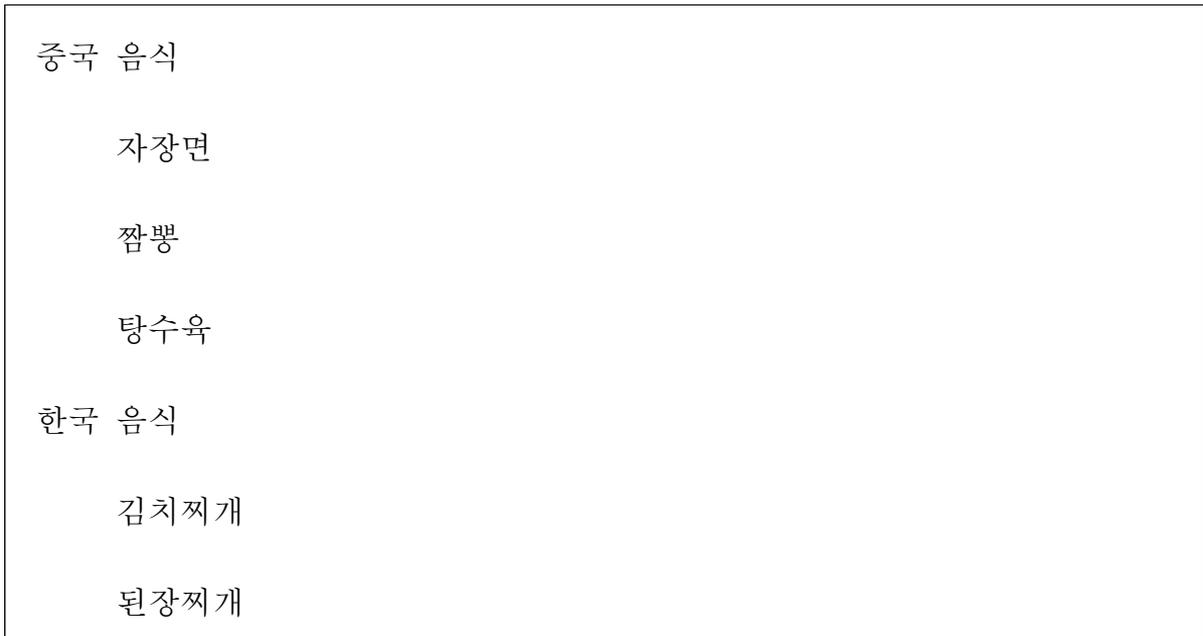
□ 적용 기술

적용 기술 56) 콘텐츠를 의미 있는 순서로 배열

웹 페이지의 콘텐츠는 마크업 언어(HTML)에서 논리적으로 작성되어야 한다. 일단 마크업 언어에서 구조적으로 이상이 없이 작성이 되면, 그것을 시각적으로 꾸미기 위해서 위치를 재조정하거나 보이는 순서까지 바꾸는 것은 CSS에서 할 수 있다. 따라서 항상 마크업 언어가 순서에 맞게 의미 있게 작성되어 있는지 주의해야 한다. 다음의 예는 중국 음식과 한국 음식을 마크업 언어로 구조적으로 의미 있게 표시한 것이다.

```
<div class="box">
  <d1>
    <dt class="menu1">중국 음식</dt>
    <dd class="item1">자장면</dd>
    <dd class="item2">짬뽕</dd>
    <dd class="item3">탕수육</dd>
    <dt class="menu2">한국 음식</dt>
    <dd class="item4">김치찌개</dd>
    <dd class="item5">된장찌개</dd>
  </dt>
</d1>
</div>
```

이것은 브라우저에서 보통 다음과 같이 표시된다.



이것을 시각적으로 다르게 보이게 하려면 CSS를 적용하면 된다. 예를 들어, 다음과 같이 중국 음식과 한국 음식을 나란히 보이도록 CSS를 적용할 수 있다.

```
<style type="text/css">
    .item1 {
        left: 0;
        margin: 0;
        position: absolute;
        top: 7em;
    }
    .item2 {
        left: 0;
        margin: 0;
        position: absolute;
        top: 8em;
    }
    .item3 {
        left: 0;
        margin: 0;
        position: absolute;
        top: 9em;
    }
    .item4 {
        left: 14em;
        margin: 0;
    }
```

```

position: absolute;
top: 7em;
}
.item5 {
left: 14em;
margin: 0;
position: absolute;
top: 8em;
}
.menu1 {
background-color: #FFFFFF;
color: #FF0000;
font-family: sans-serif;
font-size: 120%;
left: 0;
margin: 0;
position: absolute;
top: 3em;
}
.menu2 {
background-color: #FFFFFF;
color: #FF0000;
font-family: sans-serif;
font-size: 120%;
left: 10em;
margin: 0;
position: absolute;
top: 3em;
}
#box {
left: 5em;
position: absolute;
top: 5em;
}
</style>

```

CSS 적용 결과, 브라우저에는 같은 내용이 다음과 같이 다르게 표현된다. 그렇다고 해서 이 내용을 화면 낭독 프로그램을 통해 읽었을 때 중국 음식 → 한국 음식 → 자장면 → 김치찌개와 같은 순서로 나오지는 않는다. 여전히 중국 음식 → 자장면 → 짬뽕 → 탕수육과 같은 순서로 읽어주기 때문에 사용자는 정확히 제작자가 의도한 내용을 파악할 수 있게 된다.

적용 기술 58) 동적으로 생성된 요소는 그것을 유발시킨 요소 바로 뒤에 위치

어떤 경우 동적으로 콘텐츠를 보이게 하거나 안 보이게 할 필요가 있다. 가장 대표적인 경우가 풀다운 메뉴다. 이 때 하위 메뉴 항목들은 반드시 그것의 직속 상위 메뉴 다음에 나와야 한다. 그렇지 않고 상위 메뉴 항목들이 먼저 다 나온 다음에 하위 항목들이 맨 나중에 한꺼번에 나온다면, 어떤 하위 항목이 어떤 상위 항목에 속하는지 논리적으로 파악할 수 없게 된다. 단지 시각적으로 그렇게 보이는 것이 중요한 것이 아니라, DOM(Document Object Model)에서 하위 항목이 반드시 그것을 불러온 상위 항목 바로 다음에 위치해야 한다. 이렇게 함으로써 웹 페이지를 선형화시켰을 때에도 내용을 정상적으로 의미 있게 이해할 수 있고, 탭으로 이동할 때에도 정상적인 순서로 작동을 한다.

다음 예는 IMEI 코드가 무엇인지 모르는 사용자를 위해 IMEI 코드가 무엇인지 숨겨진 도움말을 보여주는 화면이다. 도움말을 보이게 하는 버튼 물음표 버튼 바로 밑으로 실제 도움말이 코드상으로도 위치해 있고, 탭 키로 이동을 할 때에도 물음표 버튼 다음에 도움말 안에 있는 링크로 이동해 정상적인 순서를 따르게 된다. 만약 도움말이 페이지의 맨 끝에 있도록 소스를 작성했다면, 문서의 의미가 훼손되었을 것이고, 탭 키를 통한 탐색도 어려웠을 것이다.

Phone Details

IMEI Code: 

PAC Code: 

숨겨진 도움말을 호출하는 링크 (도움말은 숨겨진 상태)

Phone Details

IMEI Code: 

Explanation of IMEI Code

The International Mobile Equipment Identity (IMEI) number is a unique 15-digit code used to identify an individual GSM mobile telephone. The number can be found on most mobiles by typing in *#06#. If this combination doesn't work on your mobile phone, please call our support centre on +44 (0) 1252 xxxx xxx.

[Back to IMEI input field.](#)

PAC Code: 

숨겨진 도움말을 호출하는 링크 (바로 다음에 도움말이 보임)

```

<p>
  <label for="imei">IMEI Code: </label>
  <input type="text" id="imei" />
  <a href="#imeihelp"></a>
</p>
<div id="imeihelp">
  <h3>Explanation of IMEI</h3>
  <p>Help content here...</p>
  <p><a href="#imei">Back to IMEI input field</a>.</p>
</div>

```

출처: <http://juicystudio.com/experiments/numbertransfer.php>, 2011년 6월 21일

위의 도움말 제공 방식에 대해 도움말이 IMEI Code를 입력한 다음에 나오기 때문에 시각장애인들이 도움말이 있다는 사실을 알기 어렵다는 지적도 있다. 그런 경우, 도움말 링크를 <label>안에 넣어 다음과 같이 표현해도 된다.

```

<p>
<label for="imei">

```

```
IMEI Code:
<a href="#imeihelp"></a>
</label>
<input type="text" id="imei" />
</p>
```

□ 잘못된 사례와 개선 방법

잘못된 사례 50) 상위 메뉴와 하위 메뉴의 읽는 순서가 잘못된 경우



상위 메뉴와 하위 메뉴 배치

개선 전 소스

```
<ul id="menu">
  <li class="selected"><a href="">Quick Guide</a></li>
  <li><a href="faq.html">FAQ, Q&A</a></li>
  <li><a href="breeze.html">Technical Guide</a></li>
  <li><a href="reports.html">Reports</a></li>
  <li><a href="timezone_table.html">ETC</a></li>
</ul><!--menu end-->
-
<ul id="submenu">
  <li><a href="introduction.html">Introduction</a></li>
  <li class="selected"><a href="learner_guide.html">Learner's Guide</a></li>
  <li><a href="admin_guide.html">Administrator's Guide</a></li>
</ul>
```



선형화하면 상위 메뉴가 다 나온 다음 하위 메뉴가 나온다.

위의 그림에서 상위 메뉴 항목인 Quick Guide에 대한 하위 메뉴 항목은 Introduction, Learner's Guide, Administrator's Guide 세 가지가 있다. 따라서 소스 코드에서도 Quick Guide → Introduction → Learner's Guide → Administrator's Guide 순서로 항목이 배치되어야 한다. 그러나 소스에서 상위 메뉴가 모두 끝난 다음에 하위 메뉴가 배치되었다. 즉, 시각적인 배치를 위해 소스 코드도 논리적인 순서를 어기고 시각적인 배치 순서에 맞춘 것이다. 이렇게 하면, 논리적으로 어떤 것이 상위 메뉴이고 그것의 하위 메뉴가 무엇인지 구분하기가 어렵다.

개선방법

따라서 다음과 같이 HTML 코드를 바꾸고, 그것을 보여주는 방식은 별도로 CSS에서 조정해야 한다.

```
<ul class="menu">
  <li class="selected"><a href="">Quick Guide</a></li>
  <ul>
    <ul class="submenu">
      <li><a href="introduction.html">Introduction</a></li>
      <li class="selected"><a href="learner_guide.html">Learner's
Guide</a></li>
      <li><a href="admin_guide.html">Administrator's Guide</a></li>
    </ul>
    <li><a href="faq.html">FAQ, Q&A</a></li>
    <li><a href="breeze.html">Technical Guide</a></li>
    <li><a href="reports.html">Reports</a></li>
    <li><a href="timezone_table.html">ETC</a></li>
  </ul>
```

</u1>

잘못된 사례 51) 배치용 표를 잘못 사용한 경우

다음 표는 어떤 기관의 연혁을 보여주기 위해 사용한 경우다.

2007년	기관 설립
2008년	초대 원장 취임
	홈페이지 개설
2009년	매출 1억원 달성
	서초동 사옥으로 이전

개선 전 소스

```
<table>
<tr>
  <td>
    2007년<br>
    2008년<br><br>
    2009년
  </td>
  <td>
    기관 설립<br>
    초대 원장 취임<br>
    홈페이지 개설<br>
    매출 1억원 달성<br>
    서초동 사옥으로 이전
  </td>
</tr>
</table>
```

단지 시각적으로 보이는 요소들을 배치하기 위해 논리적으로 다른 콘텐츠를 하나로 합쳐서는 안 된다. 위의 표에서 2007년, 2008년, 2009년은 각각 다른 콘텐츠인데, 표를 이용해 한 개의 셀에 들어가 있다. 화면 낭독기 프로그램으로 이것을 펼쳐서 읽으면 다음과 같이 뜻을 알아볼 수 없는 내용이 된다.

2007년, 2008년, 2009년, 기관 설립, 초대 원장 취임, 홈페이지 개설, 매출 1억원 달성, 서초동 사옥으로 이전

개선방법

표를 사용해야 한다면 다음과 같이 논리적인 내용 단위로 재구성해 주는 것이 좋다.

2007년	기관 설립
2008년	초대 원장 취임 홈페이지 개설
2009년	매출 1억원 달성 서초동 사옥으로 이전

이렇게 논리적인 단위로 표를 재구성하면, 표를 선형화했을 때 읽는 순서가 바르게 된다.

2007년, 기관 설립, 2008년, 초대 원장 취임, 홈페이지 개설, 2009년, 매출 1억원 달성, 서초동 사옥으로 이전

다음과 같이 <th>를 써서 연도를 표시하고 <td>를 써서 일어난 사건들을 구분하여 데이터를 담는 표 형식으로 구성할 수 있다. 그러나 굳이 데이터 형식의 표를 사용하지 않고 일반 배치(layout) 목적의 표를 써서 <td>만으로 구성하더라도 읽는 순서에는 문제가 생기지 않는다.

개선 후 소스 1

```
<table>
<caption>기관 연혁</caption>
<tr>
  <th>2007년</th>
  <td>기관 설립</td>
</tr>
<tr>
  <th>2008년</th>
  <td><ul>
    <li>초대 원장 취임</li>
    <li>홈페이지 개설</li>
  </ul>
</td>
</tr>
<tr>
```

```

<th>2009년</th>
<td><ul>
  <li>매출 1억원 달성</li>
  <li>서초동 사옥으로 이전</li>
</ul>
</td>
</tr>
</table>

```

이러한 종류의 연혁을 표현하기 위해 굳이 표를 쓰지 않고, 정의 목록 (definition list)을 사용하여 용어와 용어에 대한 정의처럼 다음과 같이 구성할 수도 있다.

개선 후 소스 2

```

<dl>
  <dt>2007년</dt>
  <dd>기관 설립</dd>
  <dt>2008년</dt>
  <dd>초대 원장 취임</dd>
  <dd>홈페이지 개설</dd>
  <dt>2009년</dt>
  <dd>매출 1억원 달성</dd>
  <dd>서초동 사옥으로 이전</dd>
</dl>

```

검사항목 3.3.2 (표의 구성) 표는 이해하기 쉽게 구성해야 한다.

□ 개요

복잡한 데이터를 표로 제공할 경우, 시각장애인 등도 이해할 수 있도록 표의 이해를 돕기 위한 내용 및 구조에 대한 정보를 제공해야 한다. 행과 열이 얽힌 표에 담긴 정보를 음성으로 듣고 이해하는 것은 쉽지 않다. 따라서 HTML에서는 표의 실제 데이터와 그것의 헤더(제목 열과 제목 행)를 짝지어 주는 몇 가지 방법을 제공한다. 그렇게 하면, 복잡한 표를 풀어서 읽을 때에도 내용을 이해할 수 있게 된다.

□ 적용 기술

적용 기술 59) 표 제목을 <caption>으로 제공

웹 페이지에는 제목을 <title> 요소로 제공하고, 각 섹션에는 제목을 헤딩(h1, h2 등) 요소로 제공하며, 서식 컨트롤들을 모아놓은 것에는 제목을 <legend>로 제공한다. 데이터를 담고 있는 표에도 표의 내용을 대표하는 제목을 넣어 주어야 한다. 그냥 표 위쪽에 텍스트로 제목을 넣어주면, 그것이 표의 제목인지 아닌지 기계나 보조 기술이 판단할 수 없으므로, 반드시 <table> 요소 안쪽에 <caption> 요소를 이용해 제목을 넣어준다.

서울과 부산의 야채 가격 (단위: 원)

구 분		상추	당근	오이
서울	도매가	100	200	300
	소매가	105	220	305
부산	도매가	90	150	250
	소매가	110	180	310

위와 같은 표에서 "서울과 부산의 야채 가격 (단위: 원)"이라는 표의 제목은 다음과 같이 <caption>으로 표기해야 한다.

```
<table>
<caption>서울과 부산의 야채 가격 (단위: 원)</caption>
  중간 생략
</table>
```

<caption> 요소를 쓸 경우, 브라우저에서 기본적으로 적용하는 스타일이 제작자의 의도와 맞지 않아, 일부러 <caption>의 내용을 감추고 별도로 제목을 표시하거나 <caption> 대신 그냥 일반 텍스트로 표 바깥에 제목을 표시하는 것은 잘못된 것이다. <caption> 내용의 글자 크기, 모양, 문단 배치, 색깔 등을 조정하는 것은 다음 예처럼 CSS를 이용해서 얼마든지 가능하다.

<caption> 요소의 스타일을 지정한 CSS 예

```
caption {
  font-size: 1em;
  font-family: 돋움, Dotum, sans-serif;
  font-weight: bold;
  color: #fff;
  background-color: #603;
  text-align: left;
}
```

드문 경우, <caption>을 대체하여 적절한 제목 요소(<h1>, <h2>, <h3> 등)를 표 제목으로 사용할 수도 있다. 전체적인 흐름 속에서 제목 요소가 표 제목을 나타낸다는 것이 비교적 명확한 경우 사용할 수 있지만, <caption>과 달리 표와 표 제목이 프로그램적으로 결합되지(programmatically determined) 않으므로, 만약 사용자가 표 정보에 직접적으로 접근하는 경우, 표 제목을 파악하기 어려워할 수 있다.

마지막으로, <caption>이나 summary를 쓰지 않고, 제목 요소도 사용하지 않고 단지 일반적인 문단 요소(<p>)를 표 제목으로 사용하는 것은 좋은 방법이 아니다.

적용 기술 60) 표의 구조 또는 내용에 대한 요약을 summary로 제공

데이터를 담고 있는 표의 내용을 해체하여 그것을 아무리 논리적인 순서대로

나열을 잘 한다고 하더라도 복잡한 표의 내용을 이해하기는 쉽지 않다. 따라서 데이터를 나열하기 전에 표에 어떤 데이터를 담고 있고, 전체적으로 표의 구조가 어떻게 되어있는지 요약해서 설명을 해주면 표의 내용을 파악하기가 훨씬 쉬워진다. <caption>과 달리 표의 내용과 구조를 요약 설명해 주는 summary는 <table> 요소의 속성으로 지정하면 되고, 일반적인 시각적인 브라우저에서 표시되지는 않으므로, 자세하게 표의 특성을 서술식으로 기술해 주는 것이 좋다.

만약 다음과 같은 표가 있다면,

서울과 부산의 야채 가격 (단위: 원)

		상추	당근	오이
서울	도매가	100	200	300
	소매가	105	220	305
부산	도매가	90	150	250
	소매가	110	180	310

표의 구조나 내용을 summary로 요약한 예

```
<table summary="서울과 부산의 도매와 소매 야채 가격을 비교한 표로 열이 두 단계 깊이로 되어 있다. 즉, 서울, 부산을 구분하는 단계와 도매가와 소매가를 구분하는 단계가 있음.">
<caption>서울과 부산의 야채 가격 (단위: 원)</caption>
중간 생략
</table>
```

적용 기술 61) 표의 헤더 셀과 데이터 셀의 관계 정의

표의 제목과 요약 설명을 제공하는 것만으로 표의 내용을 정확히 이해할 수는 없다. 중요한 것은 표에 담긴 데이터를 이해할 수 있게 하려면, 표의 헤더 셀과 데이터 셀을 명확히 구분해 주어야 한다. 헤더 셀이란 보통 표의 첫째 행(줄)이나 첫째 열에 표시되는 다른 데이터들에 대한 제목 셀을 뜻한다. HTML 마크업에서는 헤더 셀에 대해서는 일반적인 <td>가 아닌 <th>라는 특수한 요소를 쓰도록 하고 있다. 다음과 같은 표에서는 구분, 중간 고사, 기말 고사가 헤더 셀에 해당한다.

구분	중간 고사	기말 고사
김철수	50	60
박영희	70	80

헤더 셀을 <th>로 구분한 표의 예

```
<table summary="중간 고사와 기말 고사 성적을 개인별로 보여주는 표">
  <caption>개인별 시험 성적표</caption>
  <thead>
    <tr>
      <th>구분</th>
      <th>중간 고사</th>
      <th>기말 고사</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>김철수</th>
      <td>50</td>
      <td>60</td>
    </tr>
    <tr>
      <th>박영희</th>
      <td>70</td>
      <td>80</td>
    </tr>
  </tbody>
</table>
```

그러나 표가 복잡해지면 단순히 <th>를 지정하는 것으로 표의 헤더 셀과 데이터 셀을 짝지을 수 없는 경우가 생긴다. 이 경우 다음과 같은 두 가지 방법 중 하나로 헤더 셀과 데이터 셀을 짝지어준다.

- 첫째, 헤더 셀에 고유한 id 속성 값을 준 다음, 그것과 연관되는 내용에도 headers라는 속성을 통해 똑같은 값을 갖게 함으로써 연관성이 있다는 것을 나타내는 방법이다.
- 둘째, 헤더 셀이 영향력을 미치는 범위를 scope 속성을 이용해 표시해 주는 방법이다.

```

<!-- 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>

```

```

<!-- 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>

```

이렇게 헤더 셀과 데이터 셀을 짝지어 주게 되면, 화면 낭독 프로그램에서는 보통 다음과 같은 순서로 내용을 읽어주기 때문에 표의 내용을 이해하기가 수월해진다.

표 제목 : 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

□ 잘못된 사례

잘못된 사례 52) 표를 이미지로 표현하고, 표 제목을 제공하지 않은 경우

우리나라 신문 기사에서 자주 발견되는 사례는 데이터를 담고 있는 표를 이미지로 제공하는 것이다. 다음 사례에서는 건보정책 소위 개정안의 개정 전후 변경되는 주요 내용을 비교해서 보여주는 표다. 표 자체도 대체 텍스트가 전혀 없는 이미지로 되어 있고, 표 제목도 <caption>이나 summary, title, 또는 제목 요소(<h1>, <h2> 등)를 전혀 사용하지 않고 그냥 일반 문단 요소를 강조해서 사용하고 있다.

```
<p><strong>■ 건보정책 소위 개정안</strong></p>
<div class="articlePhotoC">
<div class="articlePhotoCarea" style="width: 500px;">

```

기사입력 2011-03-25 03:00:00 | 기사수정 2011-03-25 03:00:00 C f t

■ 건보정책 소위 개정안

	감기환자 약값 본인부담액 변화 (단위: 원)			MRI 환자 부담금 변화		
	인상 전	인상 후	인상 액	현행	5월 이후	절감액
상급종합병원	4850(본인부담률 30%)	⇒ 8080(본인부담률 50%)	▲ 3230	18만7668원	⇒ 13만1843원	▼ 5만5825원
종합병원	3420(-)	⇒ 4560(+ 40%)	▲ 1140	15만375원	⇒ 10만5644원	▼ 4만4731원
병원	2550(-)	⇒ 2550(+ 30%)	0	11만5488원	⇒ 8만1134원	▼ 3만4354원
의원	2320(-)	⇒ 2320(+ 30%)	0	8만5181원	⇒ 5만9844원	▼ 2만5337원

자료: 보건복지부(2009년 진료비 평균으로 처방일에 따라 실제 본인부담 금액은 차이 있음)

표를 이미지로 표현하고, 표 제목을 제공하지 않은 사례



실제 데이터를 <table> 요소를 사용해 표현해야 한다. 다음은 데이터를 표를 이용해 다시 재구성한 예시다.

```

<!-- 위 생략 -->
  <style type="text/css">
    * {font-family: "맑은 고딕", sans-serif;font-size:12px;}
    table {color:black;background-color:#fff;}
    caption {font-weight:bold;font-size:1.2em;}
    td {padding:0.3em;border-bottom:1px solid #ccc}
    .first, first th, .first td {border-top:2px solid
#ccc;font-size:1.1em}
    .last, .last th, .last td {border-bottom:2px solid #ccc}
    .hospital {background-color:#ddd; color:black;border-bottom:2px
solid #ccc;}
    .before {border-left:2px solid #ccc}
    .after {color:#000; background-color:inherited}
    .after:before {content:" => "}
    .increase {color:red; background-color:inherited}
    .increase:before {content:"▲"}
    .decrease {color:#0aa; background-color:inherited}
    .decrease:before {content:"▼"}
    caption {text-align:left;margin-bottom:1em}
    caption:before {content:"■ "}
  </style>

<!-- 중간 생략 -->

<table>
<caption>건보정책 소위 개정안</caption>
<tr>
  <th rowspan="2" class="last first">구분</th>
  <th colspan="3" scope="col" class="before first">감기 환자 약값 본인 부담액
변화</th>
  <th colspan="3" scope="col" class="before first">MRI 환자 부담금 변화</th>
</tr>
<tr class="last">
  <th scope="col" class="before">인상 전</th>
  <th scope="col">인상 후</th>
  <th scope="col">인상액</th>
  <th scope="col" class="before">현행</th>
  <th scope="col">5월 이후</th>
  <th scope="col">절감액</th>
</tr>

```

```

<tr>
  <th scope="row" class="hospital">상급 종합병원</th>
  <td class="before">4850원(본인 부담률 30%)</td>
  <td class="after">8080원(본인 부담률 50%)</td>
  <td class="increase">3230원</td>
  <td class="before">18만7668원</td>
  <td class="after">13만1843원</td>
  <td class="decrease">5만5825원</td>
</tr>
<tr>
  <th scope="row" class="hospital">종합병원</th>
  <td class="before">3240원(본인 부담률 30%)</td>
  <td class="after">4560원(본인 부담률 40%)</td>
  <td class="increase">1140원</td>
  <td class="before">15만375원</td>
  <td class="after">10만5644원</td>
  <td class="decrease">4만4731원</td>
</tr>
<tr>
  <th scope="row" class="hospital">병원</th>
  <td class="before">2550원(본인 부담률 30%)</td>
  <td class="after">2550원(본인 부담률 30%)</td>
  <td class="increase">0원</td>
  <td class="before">11만5488원</td>
  <td class="after">8만1134원</td>
  <td class="decrease">3만4354원</td>
</tr>
<tr class="last">
  <th scope="row" class="hospital">의원</th>
  <td class="before">2320원(본인 부담률 30%)</td>
  <td class="after">2320원(본인 부담률 30%)</td>
  <td class="increase">0원</td>
  <td class="before">8만5181원</td>
  <td class="after">5만9844원</td>
  <td class="decrease">2만5337원</td>
</tr>
</table>

<!-- 아래 생략 -->

```

■ 건보정책 소위 개정안

구분	감기 환자 약값 본인 부담액 변화			MRI 환자 부담금 변화		
	인상 전	인상 후	인상액	현행	5월 이후	절감액
상급 종합병원	4850원(본인 부담률 30%)	⇒ 8080원(본인 부담률 50%)	▲3230원	18만7668원	⇒ 13만1843원	▼5만5825원
종합병원	3240원(본인 부담률 30%)	⇒ 4560원(본인 부담률 40%)	▲1140원	15만375원	⇒ 10만5644원	▼4만4731원
병원	2550원(본인 부담률 30%)	⇒ 2550원(본인 부담률 30%)	▲0원	11만5488원	⇒ 8만1134원	▼3만4354원
의원	2320원(본인 부담률 30%)	⇒ 2320원(본인 부담률 30%)	▲0원	8만5181원	⇒ 5만9844원	▼2만5337원

이미지로 된 표를 다시 마크업으로 표현한 예

검사항목 3.4.1 (레이블 제공) 입력 서식에는 대응하는 레이블을 제공해야 한다.

□ 개요

웹에서는 서식을 통해 사용자로부터 다양한 방법으로 값을 입력받아 웹 서버에 전송하고, 그 결과에 따라 다른 페이지를 보여준다. 서식 입력은 웹에서 가장 일반적인 상호작용 방법이다. 그러나 장애인들에게는 서식에 자료를 입력하는 것이 그냥 주어진 내용을 읽는 것보다 훨씬 어려운 일에 속하며, 많은 시간을 요구한다. 따라서 입력 서식을 사용할 때 현재 위치에 대한 정확한 안내, 특정한 서식으로의 빠른 이동, 정확한 맥락의 제공, 입력할 값의 범위에 대한 안내, 오류를 방지하거나 복구할 수 있는 방법 제공, 작업을 다시 하거나 취소할 수 있는 방법 제공, 충분한 시간적 여유를 제공 등을 통해 장애인의 서식 접근성을 확보할 수 있도록 주의해야 한다.

□ 적용 기술

적용 기술 62) 입력 서식과 <label>의 명시적 관계 제공

사용자가 무엇인가를 입력하거나 선택하도록 요구하는 입력 서식에는 반드시 무엇을 입력해야 하는지 정확히 안내해 주는 레이블, 즉 입력 서식에 대한 제목이 있어야 한다. 입력 서식에 대한 레이블은 서식 가까이에 제공하는 것도 중요하지만 보다 중요한 것은 레이블이 해당 서식과 프로그램적으로 짝지어지도록 마크업 언어를 사용하는 것이다. HTML에서는 <label> 요소를 이용해 서식의 레이블을 표시하고, 레이블과 서식을 짝짓기 위해 레이블의 for 속성과 서식의 id 속성 값을 일치시키도록 하고 있다.

다음 그림에서는 Last name이라는 레이블을 제공하고 있으며, 레이블의 for="lname"과 입력 서식의 id="lname"의 값이 일치하도록 하여 해당 입력 서식과 레이블이 짝지어지도록 하고 있다. 이러한 짝짓기는 텍스트 입력 서식(input type="text") 뿐만 아니라 목록 선택(select), 배타적인 목록 선택(input type="radio"), 체크상자(input type="checkbox"), 긴 글 입력 서식(textarea)

등 대부분의 서식에 적용된다.

* First name:

* Last name:

* Preferred content: ?

서식 제목을 명시적인 레이블로 제공한 예

```
<label for="lname">Last name:</label>
<input type="text" name=".ln" value="" size="30" maxlength="50" id="lname">
<label for="intl">Preferred content:</label>
<select name=".intl" id="intl">
```

<참고> 라디오 버튼을 위한 레이블, id, name 부여

라디오 버튼은 두 개 이상의 선택 옵션 중 하나만을 선택하도록 설계된 입력 서식이다. 따라서 각각의 버튼에 대해 고유한 id를 제공하는 것은 물론이고, 선택 시 서로 배타적인 버튼들은 공통의 name 속성을 가지고 있어야 한다.

즉, 다음과 같은 라디오 버튼을 설계한다면, Threads와 Posts 버튼은 각각 show0과 show1이라는 고유한 id 값을 가지고 있지만, 또한 show라는 공통의 name 값을 가지고 있다. 물론 <label> 요소의 for 속성 값은 버튼의 고유한 id 값과 일치하도록 부여해야 한다.

Show Results as Threads Posts

라디오 버튼과 레이블을 짝지은 예

```
<label for="show0"><input type="radio" name="show" value="0" id="show0"
checked="checked" />Threads</label>
<label for="show1"><input type="radio" name="show" value="1" id="show1"
/>Posts</label>
```

<참고> 명시적 레이블링(explicit labeling)과 암묵적 레이블링(implicit labeling)

예전에는 <label> 요소를 쓸 때 레이블이 입력 서식을 감싸도록 하는 암묵적인 레이블링을 허용하거나 권장하였다. 즉,

```
<label>주소: <input type="text" name="address" id="address" size="50"></label>
```

와 같이 함으로써 접근성을 보장하도록 하였다. 그러나 이러한 방법은 <label>과 입력 서식이

물리적으로 떨어질 수밖에 없는 상황(예를 들어, 표의 서로 다른 셀에 들어가야 하는 상황)에서 쓰기 어려울 뿐만 아니라 현재의 화면 낭독 프로그램에서 제대로 레이블을 인식하지 못하는 경우가 많다. 따라서 <label>을 사용할 때 for 속성을 사용하여 어떤 입력 서식과 짝이 지어지는지 명확하게 표시하는 명시적인 레이블링 방법을 사용해야 한다.

```
<label for="address">주소:</label> <input type="text" name="address" id="address" size="50">
```

적용 기술 63) <label>을 제공할 수 없는 입력 서식에 title 제공

가끔 시각적인 디자인을 고려하여 레이블을 표시하기 어려운 때가 있다. 이 경우, 서식 요소의 title 속성을 이용하여 서식의 제목을 넣어주면 대부분의 보조 기술에서 이것을 인식할 수 있고, 시각적인 브라우저에서도 툴 팁으로 표시되어 사용자에게 도움을 준다.

다음 예에서와 같이, 검색 창 옆에 검색 대상의 종류를 선택하는 목록 상자를 제공하는 경우 별도로 검색 대상이라고 레이블을 표시할 공간이 없으므로, 목록 상자의 레이블을 title로 제공할 수 있다.

```
<label for="searchTerm">검색:</label>  
<input id="searchTerm" type="text" size="30" value="" name="searchTerm">  
<select title="검색 대상" id="scope">  
  <option value="title">제목</option>  
  <option value="keyword">키워드</option>  
</select>
```



서식 제목을 명시적으로 나타내기 어려운 경우 (검색 대상)

전화번호를 입력하는 경우, 여러 개의 텍스트 필드에 값을 넣게 되는데, 지역 번호, 국번, 끝 번호를 입력하기 위한 필드라는 것을 레이블로 표시해 줄 수 없으므로 title 속성을 이용해서 표시해 주면 된다.

전화번호 각각의 번호 필드에 title을 넣어준다.

```
<fieldset style="padding:0.5em">
  <legend>전화번호</legend>
  <input id="areaCode" name="areaCode" title="지역 번호" type="text" size="3"
value="" >-
  <input id="exchange" name="exchange" title="국번" type="text" size="3"
value="" >-
  <input id="lastDigits" name="lastDigits" title="끝번호 4자리" type="text"
size="4" value="" >
</fieldset>
```

불가피하게 서식을 표 안에 넣어야 하는 경우가 있다. 예를 들어, 여러 개의 문항으로 이루어진 설문을 구성할 때 다음 그림과 같이 표를 만들고 각각의 셀마다 하나씩 라디오 버튼을 배치해야 한다면, 각 버튼마다 title을 이용해 정확한 문항과 응답이 무엇인지 알려주어야 한다. 주의할 것은 데이터 테이블에서 제목행과 제목열을 표시하는 마크업(<th> 등)을 사용했다고 하더라도 일반 데이터 셀에 서식 요소를 넣었을 경우 반드시 title을 넣어주지 않으면 서식에 무슨 값을 넣어야 하는지 알 방법이 없다는 것이다. 다음 그림에서 맨 윗 줄 왼쪽에서 두 번째 라디오 버튼의 경우, title="1. 교육 시설 - 중립"을 넣어줌으로써, 사용자가 해당 라디오 버튼을 선택할 경우 1번 문항에 대해 "중립"이라고 응답한다는 것을 알 수 있게 한다.

다음 항목에 얼마나 만족하십니까?

항목	매우 불만족	중립	매우 만족
1. 교육 시설	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
2. 강사 실력	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

```
<table border="1">
<caption>다음 항목에 얼마나 만족하십니까?</caption>
<tr>
  <th>항목</th>
  <th>매우 불만족</th>
  <th>중립</th>
```

```

        <th>매우 만족</th>
</tr>
<tr>
    <th>1. 교육 시설</th>
    <td><input type="radio" name="no1" id="no1-1" value="1" title="1. 교육 시설 -
매우 불만족"></td>
    <td><input type="radio" name="no1" id="no1-2" value="2" title="1. 교육 시설 -
중립"></td>
    <td><input type="radio" name="no1" id="no1-3" value="3" title="1. 교육 시설 -
매우 만족"></td>
</tr>
<tr>
    <th>2. 강사 실력</th>
    <td><input type="radio" name="no2" id="no2-1" value="1" title="2. 강사 실력 -
매우 불만족"></td>
    <td><input type="radio" name="no2" id="no2-2" value="2" title="2. 강사 실력 -
중립"></td>
    <td><input type="radio" name="no2" id="no2-3" value="3" title="2. 강사 실력 -
매우 만족"></td>
</tr>
</table>

```

적용 기술 64) 여러 개의 유사한 입력 서식 묶기

여러 개의 서식 컨트롤이 한꺼번에 나올 때, 특히 라디오 버튼이나 체크 상자가 같은 이름을 같은 필드 값을 제출하는 것이라면, 그것들을 <fieldset>이라는 요소를 사용해 묶어주는 것이 좋다. <fieldset> 요소의 시작 부분에는 항상 해당 그룹의 제목을 <legend>를 이용해 표시해 주어야 한다.

서식 컨트롤이 매우 많으며, 이것들이 논리적으로 몇 개 집단으로 묶는 것이 전체적으로 서식들을 이해하는 데 도움이 된다면, 묶어주는 것이 좋다. 다음 예에서는 사는 곳과 관련된 입력 필드들을 하나로 묶고, 근무지와 관련된 필드들을 하나로 묶어주었다. 이렇게 해주면, 일반적인 화면 낭독 프로그램에서는 각 서식의 <label>을 읽어주기 전에 <legend>값을 앞에 덧붙여주어, 사용자는 어떤 맥락에서 어떤 값을 입력하는지 잊지 않고 응답할 수 있다.

<fieldset>을 이용해 유사한 서식을 묶어준 예

```
<form action="http://example.com/adduser" method="post">
  <fieldset>
    <legend>집 주소</legend>
    <label for="raddress">주소: </label>
    <input type="text" id="raddress" name="raddress" />
    <label for="rzip">우편번호: </label>
    <input type="text" id="rzip" name="rzip" />
  </fieldset>
  <fieldset>
    <legend>회사 주소</legend>
    <label for="caddress">주소: </label>
    <input type="text" id="paddress" name="caddress" />
    <label for="czip">우편번호: </label>
    <input type="text" id="czip" name="czip" />
  </fieldset>
</form>
```

□ 잘못된 사례

잘못된 사례 53) 이메일 입력 서식의 레이블을 잘못 제공한 경우

우리나라의 많은 사이트에서 이메일 주소를 입력하도록 할 때 사용자 편의성을 높이기 위해 이메일 주소를 직접 입력하게 하는 방법과 자주 사용되는 이메일 주소의 도메인을 선택하는 방법을 함께 제공하고 있다. 이 경우, 각 입력 필드에 정확히 <label>이나 title을 통해 알려주지 않는 경우가 많아 사용자들이 전체 이메일 주소를 입력해야 하는지 아니면 부분적으로만 입력해야 하는지 알 수가 없다.

다음 한 사례에서는 서식 컨트롤의 레이블을 caption이라는 잘못된 속성을 통해 제공하고 있으며, caption의 값도 모든 입력 필드가 “이메일”로 동일하게 제공하여 사용자가 구체적으로 이메일의 어떤 부분까지 입력해야 하는지 명확하게 알 수 없다.


```

<select name="pw_email_list" style="width: 107px; height: 22px;"
onchange="check_pw_mail(this.form);">
<option selected="selected">선택하세요</option>
<option value="naver.com">naver.com</option>
<!-- 중간 생략 -->
<option value="self">직접입력</option>
</select>

```

개선방법

이메일 주소를 입력받을 때 다음 사례처럼 이메일의 앞부분과 뒷부분을 나누지 않고 한꺼번에 입력받는 방법이 제일 간단하다. 이렇게 했을 때 사용자들이 이메일 주소를 잘못 입력할 가능성이 있는데, 그것은 실제 사용자가 입력한 이메일로 확인 메일을 보내서 검증받도록 하면 간단하다.

```

<label for="email-coldRegistrationForm">Email:</label>
<div class="fieldgroup">
  <span class="error" id="email-coldRegistrationForm-error"></span>
  <input name="email" value="" id="email-coldRegistrationForm"
maxlength="128" data-ime-mode-disabled="" type="text">
</div>

```

굳이 이메일의 앞부분과 뒷부분을 나누어서 입력하도록 하려면 다음과 같이 명확하게 레이블을 제공해 줄 필요가 있으며, 굳이 이메일 서비스 목록을 선택 상자에서 제공해 주는 것은 혼란을 가중시키기 때문에 삭제하였다.

이메일 주소 @

```
<label for="email-id">이메일 주소</label>
<input type="text" name="email-id" id="email-id" title="이메일 ID (@ 앞부분)">
@
<input type="text" name="email-domain" id="email-domain" title="이메일 도메인 (@
뒷부분)">
```

검사항목 3.4.2 (오류 정정) 입력 오류를 정정할 수 있는 방법을 제공해야 한다.

□ 개요

입력 서식 작성 시 오류가 발생하지 않도록 미리 어떤 값을 넣어야 하는지 안내를 해주고, 만약 그래도 실수로 오류가 발생하는 경우, 어디에서 어떤 오류가 났는지 사용자가 알 수 있어야 하고, 이를 정정할 수 있는 방법을 제공해야 한다. 오류가 생겼을 때 많이 쓰는 방법은 서식 전체를 다시 보여주고, 오류가 발생한 필드에 별도의 표시를 해는데, 이것만으로는 충분하지 않다. 화면 낭독 프로그램 사용자의 경우, 오류가 생긴 필드에 순차적으로 접근하기 전까지는 무엇이 잘못되었는지 알 수가 없어 서식 전체가 작동하지 않는 것으로 여기고 입력을 포기할 가능성이 있다. 입력에 오류가 생겼을 때 그것을 보조 기술 사용자들이 알 수 있게 하고, 오류 내용에 대한 설명에 접근할 수 있도록 하는 것이 이 검사 항목의 목적이다.

주의할 점은 입력 값의 오류를 검사하는 데 전적으로 자바스크립트와 같은 클라이언트측 스크립트 기술에만 의존해서는 안 된다는 것이다. 클라이언트측 기술은 값이 서버에 전송되기 전에 유효성 검사를 빠르게 할 수 있는 장점이 있지만, 스크립트가 작동하지 않는 환경에서는 서버에 잘못된 값이 그대로 제출되어 데이터베이스에 오류를 야기할 가능성이 있다. 따라서 클라이언트측 오류 검사 기능은 부가적으로 사용자 편의성을 위해 제공하는 것이고, 사용자가 최종적으로 값을 제출할 때 반드시 서버측에서 오류를 검사해야 한다.

□ 적용 기술

적용 기술 65) 입력 값 또는 형식에 대한 예시 제공

입력 서식에 어떤 순서로 무슨 값을 넣어야 하는지 정확한 설명을 미리 해줌으로써 사용자들마다 다르게 해석할 수 있는 여지를 줄여 주어야 한다. 또한 HTML5에서 제공하는 placeholder 속성을 이용해 예제를 보여주거나 다양한 웹 폼을 사용하면, 복잡한 자바스크립트를 쓰지 않고 사용자의 입력 오류를 줄이는

데 도움을 줄 수 있다.

다음 예에서는 몇 가지 방법으로 입력 값에 대한 안내를 제공하고 있다. 이름 입력을 위해 예(홍길동)를 미리 보여주기도 하고, title 속성을 통해 성과 이름을 붙여 쓰도록 안내하고 있다. 둘째, 이메일 입력은 HTML5의 이메일 서식을 직접 이용하여 HTML5을 지원하는 브라우저에서는 자동으로 이메일 형식이 맞지 않을 경우, 값을 제출하기 전에 경고 메시지를 주면서 초점을 이메일 필드로 이동시킨다. 또한 HTML5의 placeholder 속성을 이용해 미리 어떤 값을 입력하는 것이 좋은지 서식 필드 안에 예제를 보여주고 있다. 마지막으로, 생년월일의 경우, 날짜 형식을 yyyy-mm-dd로 입력하라고 서식 레이블에서 명시적으로 안내하고 있다.



입력 값의 형식, 범위, 예제를 미리 안내해 주는 예

```
<form>
  <label for="name">이름 (예: 홍길동): </label><br/>
  <input type="text" id="name" name="name" title="성과 이름을 붙여 쓰세요."/><br/>
  <label for="email">이메일: </label><br/>
  <input type="email" id="email" placeholder="mailid@example.com"><br/>
  <label for="birthdate">생년월일 (형식: yyyy-mm-dd): </label><br/>
  <input type="text" id="birthdate" name="birthdate" /><br/>
  <input type="submit" value="제출">
</form>
```

적용 기술 66) 서버측 오류 검사

서식 값에 대한 적합성 검사를 순전히 클라이언트측 자바스크립트로만 운용하는 것은 위험하고, 접근 가능하지 않다. 어떤 이유이든지 사용자가 자바스크립트를 사용하지 않거나 꺼놓은 경우에도 서버측에서 하는 검사를 생략할 수 없기 때문에 `<form action="">` 값으로는 반드시 서버측 URI를 넣어주어야 한다. 즉, 다

음과 같은 두 개의 예는 잘못된 것이다.

```
<form action="javascript:validateform()">
```

```
<form action="#" onsubmit="validateform()">
```

대신에, 다음과 같이 서버측 처리 프로그램을 지정해 주되, 자바스크립트 함수도 함께 지정해 주면 된다.

```
<form action="submit.php" onsubmit="return validateform()">
```

적용 기술 67) 클라이언트측 적합성 검사와 본문에 오류 메시지 표시

클라이언트 측에서 서식의 적합성을 검사한 후 오류 안내 메시지를 어떻게 보여주어야 할까? 서식 필드가 나오기 전에 페이지 내 링크(앵커, <a href>)를 이용해 오류 메시지를 보여주고, 오류 메시지의 링크를 따라가면 해당 서식 필드로 초점을 이동시킬 수 있도록 해야 한다.

다만, 클라이언트측 서식 적합성 검사는 빠르고 효율적일 수 있지만, 반드시 서버측 서식 적합성 검사와 병행해야 한다. 사용자가 클라이언트 스크립트 기능을 꺼놓았거나 작동하지 않았을 때에도 서버에 잘못된 데이터가 들어가는 것을 방지하기 위한 서버측 서식 적합성 검사가 반드시 이루어져야 한다.

다음 예에서는 필수 입력 필드에 값이 없거나, 입력 값의 형식이 잘못되었을 때, 서식 제출 전에 에러 메시지들을 문서 상단에 보여주고, 오류 메시지 링크를 따라가면 해당 서식 필드로 이동하여 다시 입력할 수 있도록 하고 있다. (<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/SCR32>에서 인용하여 변형)

서식 적합성 검사

아래 서식은 자바스크립트가 작동할 때에는 서식 값이 제출되기 전에 검사가 이루어지고, 에서 오류가 나면, 서식이 취소되고, 서식 맨 위에 오류 목록을 표시한다.

아래에 개인 정보를 입력하세요.

첫 번째 서식

2개의 오류가 있습니다.

다음 필드를 다시 확인하세요.

1. [이름:](#)
2. [나이:](#)

개인 정보
이름: <input type="text"/>
나이: <input type="text" value="서른"/>
이메일: <input type="text" value="greg@mail.com"/>
<input type="button" value="가입"/>

입력 값의 형식이 잘못된 채로 제출하면 문서 상단에 오류 메시지와 링크를 보여준다.

두 번째 서식

1개의 오류가 있습니다.

다음 필드를 다시 확인하세요.

1. [제안 내용에 대한 평가:](#)

두 번째 서식 내용
제안 내용: <input type="text" value="Nuclear is not green!"/>
이메일 주소 (선택 사항): <input type="text" value="sss@mail.com"/>
제안 내용에 대한 평가: <input type="text"/>
기타 하고 싶은 말 (선택 사항): <input type="text" value="없어요"/>
<input type="button" value="제출"/>

필수 입력 필드를 비워두고 제출하면 문서의 상단에 오류를 안내해 주고, 해당 필드로 갈 수 있는 링크를 제공한다.

- HTML 부분

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="validate.js"></script>
    <title>서식 적합성 검사</title>
</head>

<body>
    <h1>서식 적합성 검사</h1>
    <p>다음 서식은 자바스크립트가 작동할 때에는 서식 값이 제출되기 전에 검사가 이루어지고,
    그렇지 않을 경우는 서버에서 검사가 이루어진다. 별도로 표시되지 않은 모든 필드 값을 반드시 입력해야
    한다. 제출하는 시점에서 오류가 나면, 서식이 취소되고, 서식 맨 위에 오류 목록을 표시한다.</p>
    <h2>첫 번째 서식</h2>
    <form id="personalform" method="post" action="index.php">
        <div class="validationerrors"></div>
        <fieldset>
            <legend>개인 정보</legend>
            <p>
                <label for="forename">이름: </label>
                <input type="text" size="20" name="forename" id="forename"
class="string" value="">
            </p>
            <p>
                <label for="age">나이: </label>
                <input type="text" size="20" name="age" id="age" class="number"
value="">
            </p>
            <p>
                <label for="email">이메일: </label>
                <input type="text" size="20" name="email" id="email"
class="email" value="">
            </p>
        </fieldset>
        <p>
            <input type="submit" name="signup" value="가입">
        </p>
    </form>
    <h2>두 번째 서식</h2>
    <form id="secondform" method="post" action="index.php#focuspoint">
        <div class="validationerrors"></div>
        <fieldset>
            <legend>두 번째 서식 내용</legend>
            <p>

```

```

        <label for="suggestion">제안 내용: </label>
        <input type="text" size="20" name="suggestion" id="suggestion"
          class="string" value="">
      </p>
      <p>
        <label for="optemail">이메일 주소 (선택 사항): </label>
        <input type="text" size="20" name="optemail" id="optemail"
          class="optional email" value="">
      </p>
      <p>
        <label for="rating">제안 내용에 대한 평가: </label>
        <input type="text" size="20" name="rating" id="rating"
          class="number" value="">
      </p>
      <p>
        <label for="jibberish">기타 하고 싶은 말 (선택 사항): </label>
        <input type="text" size="20" name="jibberish" id="jibberish"
value="">
      </p>
    </fieldset>
    <p>
      <input type="submit" name="submit" value="제출">
    </p>
  </form>
</body>
</html>

```

- 자바스크립트 부분 (validate.js 파일)

```

window.onload = initialise;
function initialise()
{
  var objForms = document.getElementsByTagName('form');
  var iCounter;
  // Attach an event handler for each form
  for (iCounter=0; iCounter<objForms.length; iCounter++)
  {
    objForms[iCounter].onsubmit = function(){return validateForm(this)};
  }
}

// Event handler for the form
function validateForm(objForm)
{
  var arClass = [];

```

```

var iErrors = 0;
var objField = objForm.getElementsByTagName('input');
var objLabel = objForm.getElementsByTagName('label');
var objList = document.createElement('ol');
var objError, objExisting, objNew, objTitle, objParagraph, objAnchor,
objPosition;
var strLinkID, iFieldCounter, iClassCounter, iCounter;

// Get the id or name of the form, to make a unique
// fragment identifier
if (objForm.id)
{
    strLinkID = objForm.id + 'ErrorID';
}
else
{
    strLinkID = objForm.name + 'ErrorID';
}

// Iterate through input form controls, looking for validation classes
for (iFieldCounter=0; iFieldCounter<objField.length; iFieldCounter++)
{
    // Get the class for the field, and look for the appropriate class
    arClass = objField[iFieldCounter].className.split(' ');
    for (iClassCounter=0; iClassCounter<arClass.length; iClassCounter++)
    {
        switch (arClass[iClassCounter])
        {
            case 'string':
                if (!isString(objField[iFieldCounter].value, arClass))
                {
                    if (iErrors === 0)
                    {
                        logError(objField[iFieldCounter], objLabel, objList, strLinkID);
                    }
                    else
                    {
                        logError(objField[iFieldCounter], objLabel, objList, '');
                    }
                    iErrors++;
                }
                break;
            case 'number':
                if (!isNumber(objField[iFieldCounter].value, arClass))
                {
                    if (iErrors === 0)
                    {

```

```

        logError(objField[iFieldCounter], objLabel, objList, strLinkID);
    }
    else
    {
        logError(objField[iFieldCounter], objLabel, objList, '');
    }
    iErrors++;
}
break;
case 'email' :
    if (!isEmail(objField[iFieldCounter].value, arClass))
    {
        if (iErrors === 0)
        {
            logError(objField[iFieldCounter], objLabel, objList, strLinkID);
        }
        else
        {
            logError(objField[iFieldCounter], objLabel, objList, '');
        }
        iErrors++;
    }
    break;
}
}
}

if (iErrors > 0)
{
    // If not valid, display error messages
    objError = objForm.getElementsByTagName('div');

    // Look for existing errors
    for (iCounter=0; iCounter<objError.length; iCounter++)
    {
        if (objError[iCounter].className == 'validationerrors')
        {
            objExisting = objError[iCounter];
        }
    }
    objNew = document.createElement('div');
    objTitle = document.createElement('h2');
    objParagraph = document.createElement('p');
    objAnchor = document.createElement('a');
    if (iErrors == 1)
    {
        objAnchor.appendChild(document.createTextNode('1개의 오류가 있습니다. '));
    }
}

```

```

    }
    else
    {
        objAnchor.appendChild(document.createTextNode(iErrors + '개의 오류가
있습니다. '));
    }
    objAnchor.href = '#' + strLinkID;
    objAnchor.className = 'submissionerror';

    objTitle.appendChild(objAnchor);
    objParagraph.appendChild(document.createTextNode('다음 필드를 다시
확인하세요. '));
    objNew.className = 'validationerrors';

    objNew.appendChild(objTitle);
    objNew.appendChild(objParagraph);
    objNew.appendChild(objList);

    // If there were existing error, replace them with the new lot,
    // otherwise add the new errors to the start of the form
    if (objExisting)
    {
        objExisting.parentNode.replaceChild(objNew, objExisting);
    }
    else
    {
        objPosition = objForm.firstChild;
        objForm.insertBefore(objNew, objPosition);
    }
    // Allow for latency
    setTimeout(function() { objAnchor.focus(); }, 50);

    // Don't submit the form
    objForm.submitAllowed = false;
    return false;
}
// Submit the form
return true;
}

// Function to add a link in a list item that points to problematic field control
function addError(objList, strError, strID, strErrorID)
{
    var objListItem = document.createElement('li');
    var objAnchor = document.createElement('a');

    // Fragment identifier to the form control

```

```

objAnchor.href='#' + strID;

// Make this the target for the error heading
if (strErrorID.length > 0)
{
    objAnchor.id = strErrorID;
}

// Use the label prompt for the error message
objAnchor.appendChild(document.createTextNode(strError));
// Add keyboard and mouse events to set focus to the form control
objAnchor.onclick = function(event){return focusFormField(this, event);};
objAnchor.onkeypress = function(event){return focusFormField(this, event);};
objListItem.appendChild(objAnchor);
objList.appendChild(objListItem);
}

function focusFormField(objAnchor, objEvent)
{
    var strFormField, objForm;
    // Allow keyboard navigation over links
    if (objEvent && objEvent.type == 'keypress')
    {
        if (objEvent.keyCode != 13 && objEvent.keyCode != 32)
        {
            return true;
        }
    }
    // set focus to the form control
    strFormField = objAnchor.href.match(/[#]\w*$/);
    objForm = getForm(strFormField);
    objForm[strFormField].focus();
    return false;
}

// Function to return the form element from a given form field name
function getForm(strField)
{
    var objElement = document.getElementById(strField);
    // Find the appropriate form
    do
    {
        objElement = objElement.parentNode;
    } while (!objElement.tagName.match(/form/i) && objElement.parentNode);
    return objElement;
}

```

```

// Function to log the error in a list
function logError(objField, objLabel, objList, strErrorID)
{
    var iCounter, strError;

    // Search the label for the error prompt
    for (iCounter=0; iCounter<objLabel.length; iCounter++)
    {
        if (objLabel[iCounter].htmlFor == objField.id)
        {
            strError = objLabel[iCounter].firstChild.nodeValue;
        }
    }
    addError(objList, strError, objField.id, strErrorID);
}

// Validation routines - add as required

function isString(strValue, arClass)
{
    var bvalid = (typeof strValue == 'string' && strValue.replace(/^\s*|\s*$/g, '')
        !== '' && isNaN(strValue));
    return checkOptional(bvalid, strValue, arClass);
}

function isEmail(strValue, arClass)
{
    var objRE = /^[\\w-\\.']{1,}\\@([\\da-zA-Z\\-]{1,}\\.){1,}[\\da-zA-Z\\-]{2,}$/;
    var bvalid = objRE.test(strValue);
    return checkOptional(bvalid, strValue, arClass);
}

function isNumber(strValue, arClass)
{
    var bvalid = (!isNaN(strValue) && strValue.replace(/^\s*|\s*$/g, '') !== '');
    return checkOptional(bvalid, strValue, arClass);
}

function checkOptional(bvalid, strValue, arClass)
{
    var boptional = false;
    var iCounter;

    // Check if optional
    for (iCounter=0; iCounter<arClass.length; iCounter++)
    {
        if (arClass[iCounter] == 'optional')

```

```

{
    boptional = true;
}
}

if (boptional && strValue.replace(/^\s*|\s*$/g, '') === '')
{
    return true;
}

return bvalid;
}

```

적용 기술 68) 오류 메시지를 스크립트 경고창으로 제공

서식 값을 제출한 값을 클라이언트측이나 서버측에서 검사한 후 이상이 있을 때 오류가 발생한 사실을 경고창(alert)으로 보여줄 수 있다. 중요한 것은 경고창을 확인한 이후 초점이 오류가 난 서식 필드로 이동해야 사용자가 오류를 쉽게 수정할 수 있다. 화면 낭독 프로그램 사용자들의 경우, 초점을 이동시켜 주지 않으면, 오류가 난 필드를 찾기 위해 처음부터 다시 모든 서식을 거쳐 가야 한다.

이체종류	<input checked="" type="radio"/> 즉시이체 <input type="radio"/> 예약이체
출금 계좌번호	<input type="text"/> <input type="button" value="이체가능금액"/> <input type="button" value="포인트/등록조회"/> * 수수료는 포인트로 결제됩니다. 포인트 부족 시 추후계좌에서 현금으로 출금처리됩니다.
계좌 비밀번호	<input type="password"/> * 숫자 4자리
이체금액	<input type="text"/> 원 <input type="button" value="100만"/> 영 원
출금통장 표기내용	<input type="text"/> * 최대글자수: 한글,영숫자 7자 (미입력시 출금계좌'예금주명'표기)
입금 계좌번호	<input type="text"/> 은행 <input type="text"/> <input type="button" value="장예은행"/> <input type="button" value="자주쓰는계좌"/> <input type="button" value="최근입금계좌"/> * 계좌번호 입력 시 (-)없이 입력하시기 바랍니다.
입금통장 표기내용	<input type="text"/> * 최대글자수: 한글,영숫자 7자 (미입력시 출금계좌'예금주명'표기)
<input type="button" value="확인 >"/>	

웹 페이지의 메시지

! 보내는 금액을 입력하십시오!

이체 금액을 넣지 않고 서식을 제출하니 경고창으로 보내는 금액을 입력하라고 오류 메시지가 뜬다.

이체종류	<input checked="" type="radio"/> 즉시이체 <input type="radio"/> 예약이체	
출금 계좌번호	<input type="text"/>	<input type="button" value="이체가능금액"/> <input type="button" value="포인트/등록조회"/>
<p style="color: red;">* 수수료는 포인트로 결제됩니다. 포인트 부족 시 출금계좌에서 현금으로 출금처리됩니다.</p>		
계좌 비밀번호	<input type="password"/>	* 숫자 4자리
이체금액	<input type="text"/> 원	<input type="button" value="100만"/> <input type="button" value="50만"/> <input type="button" value="10만"/> <input type="button" value="5만"/> <input type="button" value="1만"/> <input type="button" value="정정"/>
출금통장 표기내용	<input type="text"/>	* 최대글자수: 한글,영숫자7자 (미입력시 수취인'은행-예금주'표기)
입금 계좌번호	<input type="text"/> 은행	<input type="button" value="장예은행"/> <input type="button" value="자주쓰는계좌"/> <input type="button" value="최근입금계좌"/>
<p>* 계좌번호 입력 시 (-)없이 입력하시기 바랍니다.</p>		
입금통장 표기내용	<input type="text"/>	* 최대글자수: 한글,영숫자 7자 (미입력시 출금계좌'예금주명'표기)
<input type="button" value="확인 >"/>		

경고창을 확인한 다음에는 반드시 오류가 난 서식 필드로 초점을 이동해야 한다.

<참고> 클라이언트측 스크립트를 이용한 실시간 오류 검사

사용자가 서식에 값을 입력할 때 잘못된 값을 입력하지 않도록 정확히 안내해 주어야 한다. 그럼에도 불구하고, 잘못된 값을 입력했을 경우 자바스크립트 등을 이용하여 경고 창을 보여주거나 DOM을 통해 오류 메시지를 텍스트로 보여줄 수 있다.

사용자가 서식에서 허용하는 값의 범위를 벗어난 데이터를 입력했을 경우, 사용자가 필드를 벗어나자마자 경고창이 뜨고, 사용자가 즉시 데이터를 수정할 수 있게 한다. 다음 예에서는 날짜 입력 필드에 날짜를 제대로 입력하지 않고, 다음 필드로 넘어가려 할 때 자바스크립트 경고창을 띄워준다. isNaN 함수는 Date.parse를 통해 숫자로 변환된 날짜가 숫자인지 아닌지를 판단해서 숫자가 아닌 경우는 경고창을 발생시킨다.

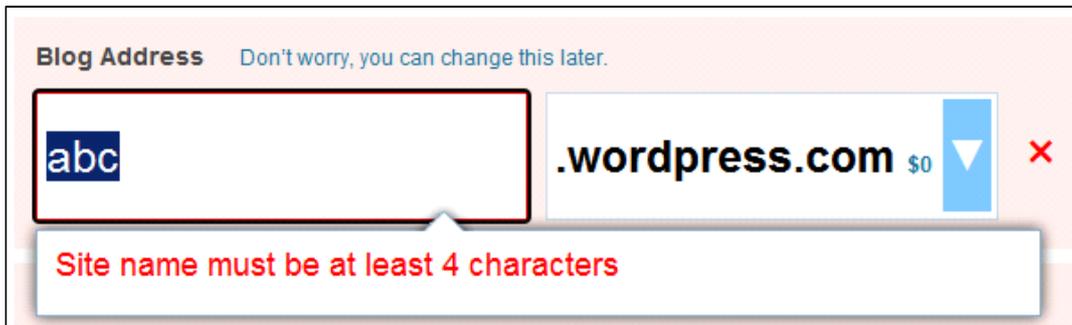
```
<label for="date">날짜:</label>
```

```
<input type="text" name="date" id="date" onchange="
```

```
if(isNaN(Date.parse(this.value)))
```

```
    alert('날짜 형식이 맞지 않습니다. 다시 입력하십시오.');" />
```

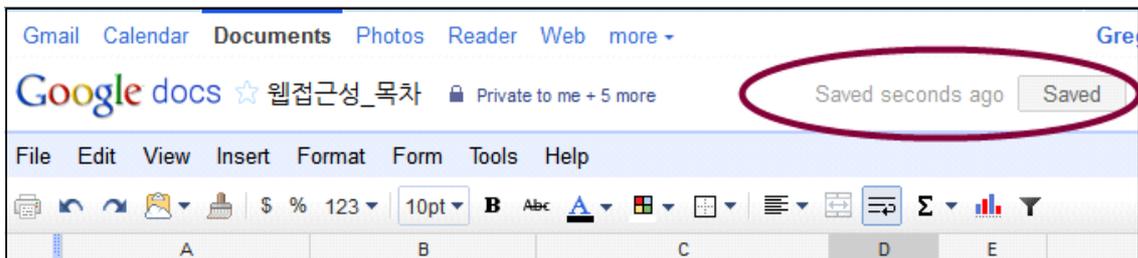
다음 그림에서는 wordpress.com에 가입할 때 블로그 주소를 입력하면서 실시간으로 주소의 요구 조건을 만족하는지 비동기식 자바스크립트(AJAX)를 이용하여 검사하고 사용자에게 피드백을 주고 있다.



<https://en.wordpress.com/signup>

<참고> 타이머를 사용하여 주기적으로 서버측 저장 기능을 불러오기

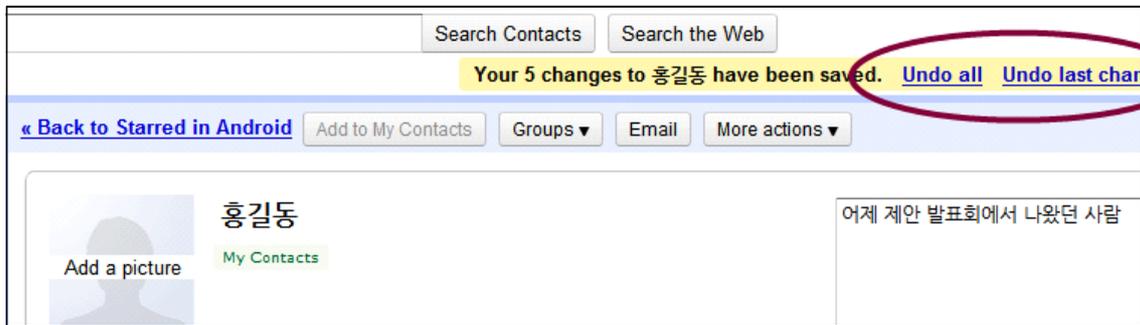
온라인에서 긴 서식을 작성하다 네트워크가 끊기거나, 서버측 사정으로 세션이 끊어지거나, 또는 사용자의 실수로 브라우저 창을 닫거나, 값을 저장하거나 제출하지 않고 다른 곳으로 이동하여 오랫동안 작성했던 내용을 처음부터 다시 작성해야 하는 경우가 종종 있다. 서버측에서 타이머를 이용해 주기적으로 서버에 값을 저장하는 방법을 사용하면 이러한 실수로 인한 데이터 손실을 줄일 수 있다.



온라인 스프레드시트 프로그램에서 수정 사항이 생기면 수시로 자동으로 서버에 저장된다.

출처 : 2011년 4월 10일 <https://spreadsheets.google.com>

역으로, 사용자가 중요한 정보를 실수로 지웠거나 덮어썼을 경우 그것을 다시 되돌릴 수 있는 방법을 제공한다. 다음 <그림 23>과 <그림 24>에서는 서식의 값을 입력하는 동안에도 실수로 수정한 내역을 되돌릴 수 있는 경우와 서식의 값의 입력이 끝난 후에도 수정 내역을 기록으로 남겨두고 다시 과거 내용으로 되돌릴 수 있는 두 가지 다른 방식의 취소(Undo) 기능의 예를 보여주고 있다.



온라인 연락처 관리 사이트에서 사용자가 수정한 내용을 되돌릴 수 있는 기능

출처 : <http://mail.google.com>



많은 위키 프로그램들은 페이지의 수정 내역을 보관하고,

잘못 수정된 것은 되돌릴 수 있는 기능

출처: http://en.wikipedia.org/w/index.php?title=Public_key_infrastructure&action=history

적용 기술 69) 중요한 서식 제출 시 재확인 절차 제공

사용자가 최종적으로 되돌릴 수 없는 거래(transaction)를 하기 전에 자신의 입력 사항이 맞는지 검토할 기회를 주어야 한다. 온라인 시험, 금융 거래, 법적인 절차에 대한 동의, 법적인 거래 등은 일반적으로 다시 취소할 수 없다. 따라서 사용자가 지금까지 입력한 여러 페이지에 걸친 내용들을 다시 왔다갔다하면서 점검할 수 있도록 캐시에 저장된 내용을 보여주거나, 지금까지 입력한 내용들을 요약해서 한 페이지에 보여주고, 마지막으로 다시 최종 확인하는 방법을 사용할 수 있

다. 최종 확인할 때 서식의 값을 제출(submit)하는 버튼 외에도 다시 사용자가 모든 값을 확인했다는 확인을 받는 체크상자(checkbox)를 함께 제공하면 실수로 서식 값이 제출되는 것을 방지할 수 있다.

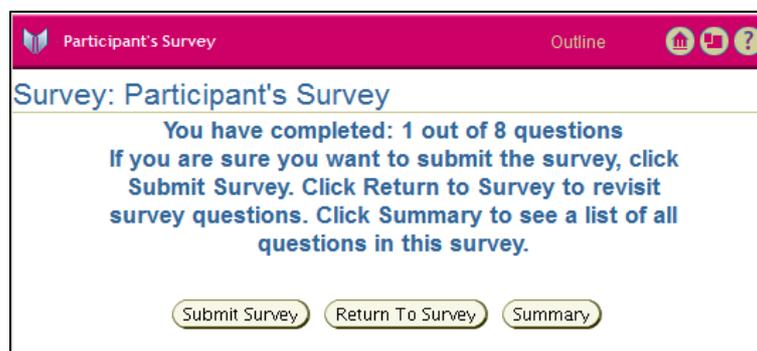
다음 예에서는 전자 상거래 사이트에서 최종 물품을 주문하거나 결제하기 전에 결제 예정 내용을 요약해서 다시 보여주고 확인을 받아 사용자가 상품을 잘못 선택했거나 마음이 바뀌었을 때 취소할 수 있는 기회를 여러 번에 걸쳐 제공하고 있다.

결제 내역 확인	
주문상품금액정보	총 1권(장), 12000원
상품 주문 총액	12000원
선물포장료	0원
배송료	0원
결제 총액	12000원
남은 결제 금액	

남은 결제금액 = (주문상품금액 - 쿠폰할인금액) +

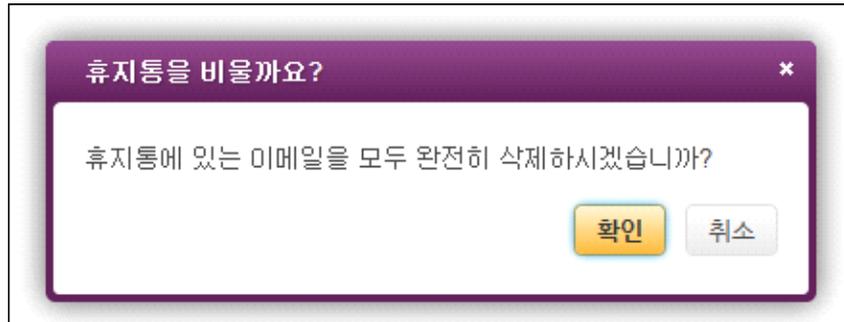
전자상거래에서 최종 결제를 하기 전에 결제 내역이 맞는지 확인하는 절차가 있다.

다음 그림에서는 여러 문항 또는 여러 페이지로 이루어진 설문(또는 온라인 시험)에서 최종 설문을 제출하기 전에 다시 설문지로 돌아가거나, 응답한 내용의 요약을 확인해 볼 수 있는 기회를 제공함으로써 실수로 잘못된 설문 내용이 제출되지 않도록 하고 있다.



설문의 응답을 최종 제출하기 전에 설문지로 돌아가거나 전체 요약은 볼 수 있도록 제공하고 있는 예
출처 : 2011년 4월 10일 Oracle iLearning 시스템의 설문 화면

다음 그림에서는 온라인 메일 시스템에서 휴지통에 있는 메일을 완전히 지워 휴지통을 비우면 다시 살릴 수 없기 때문에 다시 한 번 확인하는 경고창을 제공하고 있다.



웹 기반 이메일 시스템에서 휴지통에 있는 메일을 완전히 삭제하기 전에 확인 절차를 제공(Yahoo! 메일)

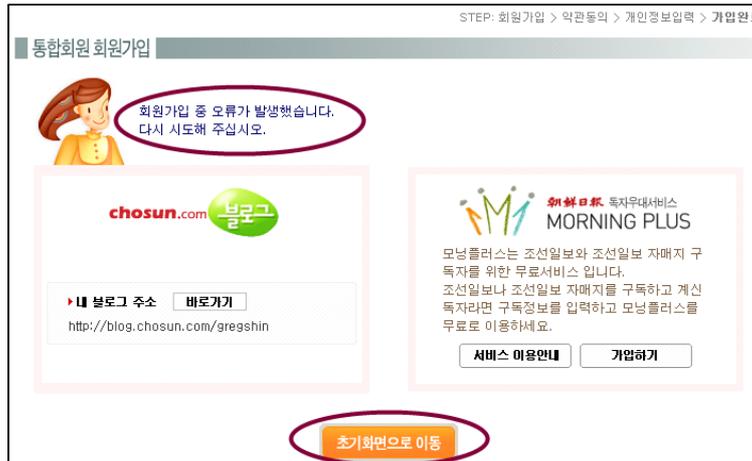
□ 잘못된 사례

잘못된 사례 54) 오류 정보를 확인한 후 오류 발생 페이지로 돌아가지 못하는 경우

서식 입력 값을 제출하고 오류가 생겼을 때, 그 오류를 표시하는 방법은, 앞서 소개했듯이, 문서 내부에 DOM(Document Object Model)을 이용해서 하는 것이 좋다. 또한 자바스크립트로 경고창(alert)을 이용해도 좋다. 다만, 오류 피드백 내용을 별도 페이지에 보여주고, 사용자가 실제 해당 항목을 수정하기 위해 뒤로 가기를 눌러야 한다면, 사용자가 모든 오류 항목들을 기억해야 한다는 부담이 생긴다. 따라서 일반적으로 원래 페이지를 다시 생성해서 DOM을 이용해 오류 메시지를 문서 상단에 보여주는 것이 좋은 방법이다.

또한, 서식 제출 후 오류가 생겼을 때 서식의 모든 입력 값이 사라지거나 서식 페이지로 돌아가는 링크가 없는 경우 사용자는 한두 가지 실수만 수정하면 될 것을 다시 서식 페이지로 찾아가 처음부터 입력을 반복해야 한다. 다음 그림에서는 사이트에 회원 가입을 하기 위해 정보를 입력한 후 서식을 제출하면, 새로운 페이지로 이동하여 오류가 났다고 안내해 준다. 문제는 어떤 오류인지에 대한 설명이 없어 무엇이 잘못 되었는지 알 수도 없고, 브라우저의 [뒤로 가기] 버튼을 눌러서 뒤로 가면 서식 자체가 사라져 버려 작동하지 않는다. 오류 안내 페이지에서도 원래의 회원 가입 페이지로 돌아가는 링크가 없이 아예 전체 사이트의 초기(홈) 화

면으로 가도록 하고 있기 때문에, 회원 가입 페이지를 다시 찾아간 다음, 모든 정보를 처음부터 다시 입력해야 한다.



회원 가입에 필요한 정보를 정확히 입력하지 않고 서식을 제출하면, 새로운 페이지에서 오류 발생 사실을 알려줌
출처 : 2011년 4월 11일

<https://membership.chosun.com/join/registUser.jsp?site=>

개선방법

서식을 입력하는 페이지 내에서 값을 최종 제출하기 전에 자바스크립트 경고창을 이용해 입력이 잘못되었다는 것을 알리고, 해당 서식 컨트롤로 이동할 수 있도록 한다. 서식을 제출하기 버튼을 누르는 순간 입력 오류를 클라이언트 측에서 먼저 검사하여 오류의 목록을 현재 문서의 최상단에 각각의 오류에 대한 링크와 함께 제공하여 오류를 수정할 수 있도록 한다. 마지막으로, 서식 값을 최종적으로 서버에 전송하기 전에 서버에서 값의 적합성 검사를 실시하면 된다.

원칙 4. 견고성(Robust): 웹 콘텐츠는 미래의 기술로도 접근할 수 있도록 견고하게 만들어야 한다.

□ 견고성

웹 콘텐츠는 견고성이 있어야 한다. 견고성이란 웹 사이트에서 제공하는 콘텐츠가 현재 사용되고 있는 브라우저나 보조기술 뿐만 아니라 앞으로 개발될 브라우저와 보조기술을 이용하여 사용이 가능해야 한다. 따라서 견고성이 있는 웹 콘텐츠는 관련 기술이 새로 개발되더라도 아무런 문제없이 사용될 수 있다.

견고성이 있는 웹 콘텐츠를 제공하려면, 콘텐츠를 구현하는 과정에서 사용하는 기술과 관련한 표준을 준수해야 한다. 예를 들어, 웹 콘텐츠를 구성하는 일부 콘텐츠를 어떤 프로그래밍 언어를 이용하여 구현하려면, 반드시 해당 언어가 정의하고 있는 문법을 준수해야 한다.

최근에는 다양한 기능과 형태의 서비스를 웹을 통하여 제공하고 있으며, 이들 서비스는 웹 애플리케이션 방식으로 구현된다. 웹 애플리케이션은 관련 콘텐츠와 함께 사용자 컴퓨터 또는 장치에 다운로드 되며, 사용자는 브라우저 등을 통해 접근이 가능하다. 따라서 웹 애플리케이션을 포함하고 있는 웹 콘텐츠를 모든 사용자가 접근할 수 있으려면 웹 콘텐츠의 접근성뿐만 아니라 웹 애플리케이션의 접근성도 보장되어야 한다.

□ 관련 지침

<표 9> 견고성 관련 지침 및 검사항목

지침(2개)	검사항목(2개)
4.1(문법 준수) 웹 콘텐츠는 마크업 언어의 문법을 준수해야 한다.	4.1.1(마크업 오류 방지) 마크업 언어의 요소는 열고 닫음, 중첩 관계 및 속성 선언에 오류가 없어야 한다.
4.2(웹 애플리케이션 접근성) 웹 애플리케이션은 접근성이 있어야 한다.	4.2.1(웹 애플리케이션 접근성 준수) 콘텐츠에 포함된 웹 애플리케이션은 접근성이 있어야 한다.

검사항목 4.1.1 (마크업 오류 방지) 마크업 언어의 요소는 열고 닫음, 중첩 관계 및 속성 선언에 오류가 없어야 한다.

□ 개요

콘텐츠를 구성하고 있는 소스 코드에 문법적인 오류가 있거나 모호한 부분이 있다면 사용하는 브라우저에 따라 오류를 해석하는 방법이나 모호한 부분을 처리하는 과정이 달라 사용자에게 전달되는 정보에 차이가 발생할 수 있다. 이러한 차이로 인하여 사용자는 웹 콘텐츠의 인식, 이해 및 사용에 크고 작은 영향을 주게 된다. 따라서 콘텐츠를 개발하는 과정에서 마크업 언어의 표준을 준수하고 가능한 문법적 오류나 모호성이 없도록 구현되어야 한다.

□ 적용 기술

적용 기술 70) 여는 태그와 닫는 태그의 정확한 사용

마크업 언어를 이용하여 마크업 문서를 구현할 때 반드시 태그의 열림과 닫힘이 일치해야 한다.

다음 HTML 코드는 올바르게 구현된 소스다.

```
<p>요소를 여는 태그의 닫음 표시가 불명확한 문단입니다.</p> (O)
```

그러나 다음 HTML 소스는 문단을 p 태그의 속성으로 제공하여 문법적으로 잘못된 구현한 예다.

```
<p 요소를 여는 태그의 닫음 표시가 불명확한 문단입니다.</p> (X)
```

다음 HTML 소스는 닫는 태그 대신 여는 태그를 중복 사용한 경우로, 문법적인 오류라기보다는 논리적인 오류에 해당하는 경우다. 이 경우, 브라우저가 p 태그를 두 번 선언한 것으로 간주한다.

<p>요소를 닫는 태그에 슬래시(slash)가 빠진 문단입니다.<p> (X)

<참고> HTML과 XHTML의 차이점

XHTML은 HTML을 XML의 형식으로 구성한 XML 문서의 한 종류다. HTML과는 달리, XHTML은 문법이 엄격하여 표준 XML 해석기(parser)로 해석이 가능하다. XHTML 1.0과 XHTML 1.1이 현재 W3C의 권고안이며, 현재 HTML5의 한 부분으로서 XHTML5가 개발 중이다. HTML과 XHTML의 주요 차이점은 다음과 같다.

1. 태그의 열고 닫음이 명확해야 한다.
2. 단독으로 사용되는 태그도 닫는 표기를 해야 한다.
3. 속성 값은 따옴표로 묶어야 한다.
4. 대소문자를 엄격하게 구분한다.
5. 축약형 속성을 사용하지 않는다.
6. 스크립트와 스타일은 CDATA로 표기한다.

<참고> 닫는 태그의 생략이 가능한 경우

다음 HTML 태그는 닫는 태그의 생략이 가능하다.

html, head, body, p, li, dt, dd, thead, tbody, tfoot, colgroup, tr, th, td, option

보다 자세한 HTML과 그밖에 마크업언어에 대한 정보는 W3C 웹사이트에서 확인할 수 있다.

HTML 4.01 Specification: <http://www.w3.org/TR/html401/>

적용 기술 71) 속성 이름과 속성 값의 정확한 사용

따옴표로 속성 값을 구분하여 제공할 경우 따옴표를 누락하게 되면 일부 정보가 무시될 수 있다. 다음 HTML 소스는 name 다음에 따옴표를 누락하여 type=... 부분도 title 속성 값으로 인식하게 된다.

```
<input title="name type="text"> (X)
```

따라서 다음과 같이 따옴표의 시작과 끝을 정확히 표시해야 type 속성이 정확하게 인식이 된다.

```
<input title="name" type="text"> (O)
```

다음 예는 속성과 속성 사이에 공백이 누락된 경우다.

```
<input title="name" type="text"> (X)
```

이 경우도 속성 구분이 모호하므로 반드시 따옴표 사이를 빈칸(space)으로 구분해야 한다.

다음 HTML 소스는 속성 값을 따옴표로 구분하여 제공하지 않는 예다. 속성 값이 빈칸을 포함하는 경우 반드시 따옴표를 이용하여 표시해야 한다.

```
<input title=이름을 입력하세요. type=text> (X)  
<input title="이름을 입력하세요." type="text"> (O)
```

하나의 요소에서 동일한 속성을 두 번 이상 선언할 경우 중복 선언한 속성 중 하나만 적용되므로 중복적으로 선언하지 않아야 한다.

```
<p style="color: #000" class="notice" style="width: 50em"> (X)  
<p class="notice" style="color: #000; width: 50em"> (O)
```

속성 값에 따옴표를 사용해야 할 경우, 따옴표를 그냥 사용하면 속성 값의 종결 문자로 인식하기 때문에 값으로 사용할 때 "로 사용해야 한다.

```
<input type="text" value=""가나다""> (X)  
<input type="text" value="&quot;가나다&quot;"> (O)
```

적용 기술 72) 태그의 정확한 중첩관계

태그를 사용할 경우 중첩관계가 엇갈리지 않도록 해야 한다. 특히 이를 과싱하는 브라우저에 따라서 오류를 처리하는 방법이 서로 달라 과싱 결과가 다른 경우도 종종 발생한다. 무엇보다도 태그를 사용하는 과정에서 중첩관계에 오류가 없도록 다음 예와 같이 태그의 열고 닫는 순서를 잘 지켜야 한다.

```
<p>중첩관계가 <strong>명확해야 한다.</p></strong> (X)  
<p><strong>명확해야 한다.</strong></p> (O)
```

적용 기술 73) 표준에 부합하는 태그와 속성 이름, 값 사용

브라우저에 따라서는 표준에 명시되지 않는 태그나 속성을 허용하는 경우가 있다. 그러나 마크업 문서를 작성할 경우 반드시 표준에 명시된 태그와 속성을 사용해야 이 문서를 어떠한 브라우저와 보조기술을 이용하여 접근할 때에도 정상적인 접근이 가능하게 된다.

일부 마크업 언어는 사용자가 속성을 정의하여 사용할 수 있다. 예를 들어, HTML5 언어는 'data-' 접두어를 사용하여 속성 이름을 정의하면 문서 작성 시 사용자가 정의한 속성을 사용할 수 있다.

마크업 문서의 오류 여부를 수동으로 검사하는 것은 매우 어려운 일이다. 따라서 이들 마크업 문서의 문법 오류를 자동으로 검사할 수 있는 다양한 도구가 개발되어 사용되고 있다.

<참고> 마크업 언어의 검사

W3C에서는 여러 가지 마크업 문서의 오류여부를 검증할 수 있는 마크업 유효성 검사 도구 (<http://validator.w3.org/>)를 제공하고 있다. 현재 지원하고 있는 마크업 언어는 HTML, XHTML, XML, SMIL, MathML 등이다.

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. This validator is part of [Unicorn](#), W3C's unified validator service. See the [list of checks](#) performed by Unicorn and learn about [other tools](#)

The W3C validators are hosted on server technology donated by HP, and supported by community donations. [Donate](#) and help us build better tools for a better web.

[Home](#) [About...](#) [News](#) [Docs](#) [Help & FAQ](#) [Feedback](#) [Contribute](#)



This service runs the W3C Markup Validator, [v1.1](#).
COPYRIGHT © 1994-2010 W3C® (MIT, ERCIM, KEIO), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.



CSS나 자바스크립트 문법의 오류는 기능에 직접적인 영향을 주기 때문에 마크업 언어와 함께 오류가 발생되지 않도록 주의해야 한다.

□ 잘못된 사례 및 개선방법

잘못된 사례 55) 아이디를 중복 선언한 경우

한 페이지를 구성하는 문서 내에서는 동일한 아이디를 중복하여 사용할 수 없다. 'content'와 같이 여러 용도로 사용될 수 있는 경우도 다음 예와 같이 한 문서에서 두 번 사용한다면 예기치 못한 오류가 발생할 수 있다. 다음 소스는 텍스트 영역(textarea) 요소의 값으로 'content'를 참조하려 하였으나 이미 레이아웃 영역에서 content라는 아이디를 사용하고 있어 의도와 달리 다른 요소를 참조하게 되는 사례다.

```
<div id="header"> ... </div>
<div id="content"> ... </div>
...
<form ...>
  ...
  <textarea id="content" name="content"></textarea>
  ...
</form>
...
</div>
<script type="text/javascript">
function validateForm() {
  ...
  var e1 = document.getElementById('content');
  if (e1.value) {
    ...
  }
  ...
}
</script>
```

개선방법

아이디를 중복하여 선언하지 않는다. 특히 여러 개발자가 협업하여 웹 콘텐츠를 제작할 경우 아이디의 중복 선언과 같은 일이 종종 발생한다. 따라서 아이디나 속성 값들이 충돌하지 않도록 사전 협의와 문서화가 필수적이다.

content영역을 나타내는 div 요소의 id를 충돌이 발생하지 않도록 'content'를

'content-area'로 변경한다.

```
<div id="header"> ... </div>
<div id="content-area"> ... </div>
  ...
  <form ...>
    ...
    <textarea id="content" name="content"></textarea>
    ...
  </form>
  ...
</div>
<script type="text/javascript">
function validateForm() {
  ...
  var e1 = document.getElementById('content');
  if (e1.value) {
    ...
  }
  ...
}
</script>
```

잘못된 사례 56) 태그의 중첩관계에 오류가 있는 경우

다음 코드는 공지사항을 링크 목록으로 구현하는 과정에서 각 링크 항목의 닫는 태그()를 누락한 사례다. 이 경우, 공지사항뿐만 아니라 뒤따르는 콘텐츠에도 링크가 제공된다. 다음 코드에서는 "사용자 게시판"에도 공지사항의 5번째 항목의 링크가 제공된다.

```
<div class="notice">
  <h3></h3>
  <ul class="arw">
    <li><a href="notice_list?id=899">시스템 정기 점검 안내</li>
    <li><a href="notice_list?id=650">댓글을 평가해주세요!</li>
    <li><a href="notice_list?id=183">친구관계를 등록할 수 있습니다.</li>
    <li><a href="notice_list?id=407">구독 버튼 오류 정정 안내</li>
    <li><a href="notice_list?id=748">내 글 목록 페이지 개선</li>
  </ul>
</div>
<div>
  <h3></h3>
```

개선방법

링크 항목별로 반드시 닫는 태그를 제공해야 한다.

```
<div class="notice">
  <h3></h3>
  <ul class="arw">
    <li><a href="notice_list?id=899">시스템 정기 점검 안내</a></li>
    <li><a href="notice_list?id=650">댓글을 평가해주세요!</a></li>
    <li><a href="notice_list?id=183">친구관계를 등록할 수 있습니다.</a></li>
    <li><a href="notice_list?id=407">구독 버튼 오류 정정 안내</a></li>
    <li><a href="notice_list?id=748">내 글 목록 페이지 개선</a></li>
  </ul>
</div>
<div>
  <h3></h3>
```

검사항목 4.2.1 (웹 애플리케이션 접근성 준수) 콘텐츠에 포함된 웹 애플리케이션은 접근성이 있어야 한다.

□ 개요

자바스크립트를 이용하여 구현한 웹 콘텐츠가 접근성을 지원하지 않는다면 모든 사용자가 사용할 수 없게 된다. 마찬가지로, 플러그인이나 웹 애플리케이션을 제공하는 경우도 이들 플러그인이나 웹 애플리케이션은 반드시 사용자의 접근성을 지원하도록 구현해야 한다.

플러그인이나 웹 애플리케이션을 구현할 때 우선 현재 사용 가능한 보조기술이 지원하는 플랫폼이나 기술을 사용해야 한다. 그렇지 않으면 이들 플러그인이나 웹 애플리케이션을 접근성 있게 구현하더라도 현재의 브라우저와 보조기술로는 접근이 어렵기 때문이다.

만일 현재의 보조기술을 이용하면 구현이 불가능한 플러그인이나 웹 애플리케이션은 현재의 보조기술의 적용을 예외로 하되 향후 개발되는 보조기술이 지원할 수 있도록 접근성을 지원하여 개발해야 한다. 또한, 이들 플러그인이나 웹 애플리케이션은 반드시 보조기술 사용자가 접근할 수 있는 대체 수단을 함께 제공해야 한다.

□ 적용 기술

적용 기술 74) 플러그인 플랫폼이 제공하는 접근성 API 활용

플러그인을 이용하여 콘텐츠를 개발할 경우 플러그인 플랫폼이 제공하는 접근성 API를 활용하여 구현한다.

플래시 콘텐츠는 Adobe에서 제공하는 접근할 수 있는 플래시(Best Practices for Accessible Flash Design;

http://www.adobe.com/resources/accessibility/best_practices/best_practices_acc_flash.pdf) 문서에서 제시하는 방법을 활용하여 콘텐츠를 제작한다. 다음

목록은 웹 애플리케이션을 구현하는 도구에 대한 접근성 제공 방법에 관한 정보다.

- Best Practices for Accessible Flash Design
(http://www.adobe.com/resources/accessibility/best_practices/best_practices_acc_flash.pdf)
- Flex accessibility (<http://www.adobe.com/accessibility/products/flex/>)
- UI Automation and Microsoft Active Accessibility
(<http://msdn.microsoft.com/en-us/library/ms788733.aspx>)
- Silverlight Accessibility Overview
([http://msdn.microsoft.com/en-us/library/cc707824\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/cc707824(VS.95).aspx))
- Java Accessibility Helper Early Access v0.8
(<http://java.sun.com/developer/earlyAccess/jaccesshelper/>)
- Accessible Explorer
(<http://msdn.microsoft.com/en-us/library/ms696082.aspx>)

적용 기술 75) 웹 애플리케이션의 대체 콘텐츠 제공

플러그인을 사용하는 웹 애플리케이션을 제공할 때 웹 애플리케이션과 동등한 기능을 수행할 수 있는 대체 콘텐츠를 함께 제공한다.

다음 그림은 플래시 플러그인을 사용할 때 플러그인 사용이 가능한 환경(좌)과 플러그인 사용이 가능하지 않은 환경(우)에서 각각 웹 애플리케이션이 어떻게 제공되고 있는지를 보여주는 예다. 플러그인 사용이 가능하지 않은 환경에서는 웹 애플리케이션 기능과 동등한 기능을 하는 이미지로 되어 있는 콘텐츠를 제공하고 있다.

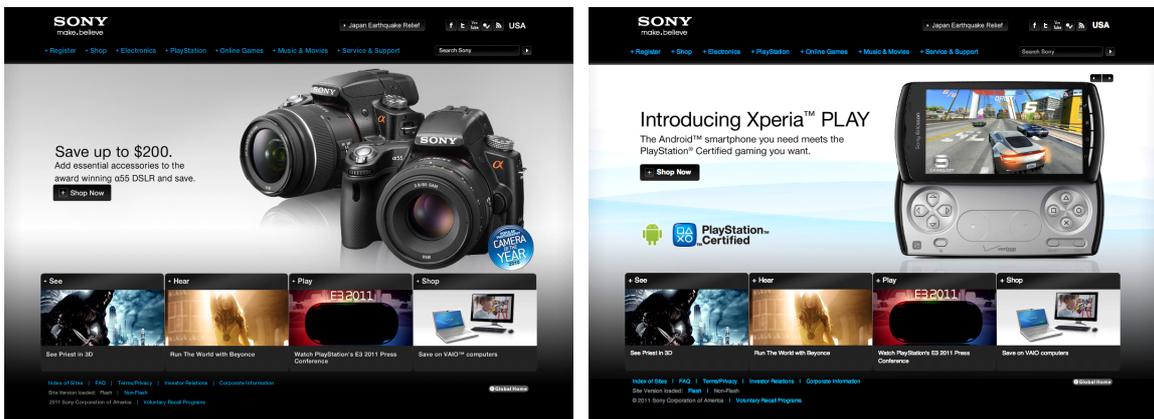


다음 소스는 HTML object 구문을 사용하여 플래시 콘텐츠에 대한 대체 콘텐츠를 제공하기 위한 경우다. object 요소는 포함하고 있는 콘텐츠를 대체 콘텐츠로 인식하여 플러그인을 구동할 수 없을 때 하위에 포함하고 있는 콘텐츠를 대신 출력하게 된다.

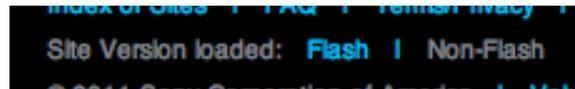
```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0" width="550" height="400">
  <param name="movie" value="flash_movie.swf">
  <!--[if !IE]> <-->
  <object type="application/x-shockwave-flash" data="flash_movie.swf"
width="550" height="400">
    <p></p>
  </object>
  <!--> <![endif]-->
</object>
```

대체 콘텐츠는 플러그인 기술의 사용 가능 여부에 따라 자동으로 선택되게 할 수도 있지만 사용자에게 사용 여부를 선택하게 할 수도 있다. 사용자에게 플러그인 기술의 사용여부를 선택할 수 있게 하면 사용자는 자신의 환경과 선호도에 따라 적합한 웹 애플리케이션을 사용할 수 있게 된다.

다음은 사용자의 선택에 따라 플래시 버전과 플래시가 아닌 버전을 제공하는 사이트의 예다.



각 버전의 사이트 푸터에 플래시 버전이나 플래시가 아닌 버전을 선택하는 링크를 제공하고 있다.



대체 콘텐츠를 제공하고 있더라도 플러그인 플랫폼의 접근성 기능을 잘 활용하여 콘텐츠를 제공하는 것이 중요하다. 또한 대체 콘텐츠로 제공된 웹 애플리케이션도 접근성 있게 제작할 필요가 있다.

나온 지 얼마 되지 않는 새로운 기술의 경우 현재 수준의 보조기기에서 아직 접근성 기능을 충분히 지원하지 않을 수도 있다. 또한 새로운 기술의 기술적인 완성도가 완벽하지 않아서 접근성 API 제공이 미흡할 수 있다. 이러한 상황에서는 접근성이 낮다는 이유로 신기술의 사용을 제한하기보다는 접근성 있는 대체 콘텐츠를 제공하여 접근성을 보장하는 것도 좋은 해결책이 될 수 있다.

<참고> WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications)

ARIA는 W3C의 WAI에서 권고하는 접근성 향상 방법이다. 웹 페이지나 웹 애플리케이션의 구성 요소들에 역할(role), 속성(property), 상태(state) 정보를 기술할 수 있게 하여 다른 웹 애플리케이션이나 브라우저, 보조기술, 접근성 평가 도구 등으로 하여금 접근이 용이하게 한다.

WAI-ARIA는 기본적으로 HTML에 적용되도록 고안되었지만 HTML 뿐만 아니라 SVG같은 마크업 기반의 언어들에 넓게 적용 가능하다.

적용 기술 76) 용도, 목적지, 종류, 사용법에 맞는 사용자 인터페이스 요소 사용

사용자 인터페이스 요소는 사용자에게 그 용도, 목적지, 종류, 사용법 등에 관한 올바른 정보를 제공할 수 있어야 한다. 일반적으로, 플랫폼에서 제공하는 인터페이스 요소를 사용했을 때 플랫폼에서 이러한 요소에 대한 접근성 정보를 이미 적절하게 제공하고 있기 때문에 별다른 조치를 취하지 않아도 사용자가 이에 대한 명확한 정보를 취할 수 있다.

만약 HTML에서 링크를 a 태그를 이용하여 구현하지 않고 div나 span 요소에 자바스크립트 클릭 이벤트를 사용하여 구현하면 키보드 초점을 받지 못하거나 이를 링크로 인식하지 못하게 된다. 또한 서식 버튼 요소를 구현하는 데 input 요소를 사용하지 않고 img 요소나 a 요소 등을 사용하여 구현하여도 비슷한 문제가 발생할 수 있다.

플랫폼에서 제공하는 인터페이스 요소가 아닌 다른 인터페이스 요소를 만들어서 사용할 경우 접근성 API를 이용하여 이름과 용도가 노출될 수 있도록 하여야 한다. 예를 들어, 플래시를 이용해서 인터페이스 요소를 제작할 때 서식 요소에 레이블 속성을 지정하거나 이미지 버튼에는 이름을 지정하여야 한다.

□ 잘못된 사례 및 개선방법

잘못된 사례 57) 이미지 링크를 자바스크립트로 잘못 구현한 경우

다음 소스 코드는 이미지를 클릭하면 자바스크립트를 이용하여 popup.html 페이지를 열도록 구현한 사례다. 이 코드의 문제점은 두 가지다. 첫 번째는 키보드로 접근이 불가능하다는 점이며, 두 번째는 이미지가 의미하는 사용자 인터페이스의 종류가 링크임을 알려주지 못한다는 점이다.

```

```

개선방법

HTML의 a 태그를 이용하여 키보드 초점을 받을 수 있도록 하고 이미지가 링크로 사용됨을 알려주도록 수정해야 한다. 다음 코드는 a 태그를 이용하여 수정한 결과다.

```
<a href="popup.html" target="_blank"
  onclick="window.open(this.href, '', 'width=300,height=200'); return false;">
  
</a>
```

여기에서 a 태그의 href 속성으로 목적지 URL을 제공하고 onclick 이벤트 핸들러의 URL을 this.href로 변경하였다. onclick 이벤트 핸들러를 img 요소가 아닌 a 태그에 사용하여 this.href로 URL을 참조할 수 있게 하였다. 이벤트 핸들러에 'return false;' 부분을 생략하면 마우스 클릭과 키보드에 의해서 새창열기와 함께 원래 창의 콘텐츠가 'popup.html'로 로드된다. 따라서 'return false;' 부분을 반드시 추가해야 한다.

잘못된 사례 58) 잘못된 이미지 버튼 구현 사례

다음 코드는 이미지(login.gif)를 버튼으로 이용하도록 자바스크립트로 onclick 이벤트를 제공하는 사례다.

```

<script type="text/javascript">
function submitForm() {
    loginForm.submit();
}
</script>
<form id="loginForm" name="loginForm" action="">
    아이디 <input type="text" name="loginId">
    비밀번호 <input type="password" name="loginPassword"><br>
    
</form>

```

이 코드의 문제점은 첫째, 키보드 초점이 버튼 이미지로 이동하지 않아 키보드로 실행시킬 수 없다는 점이며, 둘째, 이미지가 의미하는 사용자 인터페이스 요소의 종류가 버튼임을 알 수 없다는 점이고, 셋째, 이미지 버튼에 대체 텍스트가 누락되어 그 용도를 알 수 없는 점이다.

개선방법

위와 같이 이미지를 버튼으로 사용하고자 할 경우 반드시 키보드를 이용할 수 있고 그 용도와 종류를 알 수 있도록 구현해야 한다.

서식 전송을 의미하는 버튼 컨트롤은 form 요소와 함께 <button type="submit">, <input type="submit">, <input type="image"> 요소 중 하나를 이용하여 구현해야 한다.

다음 코드는 서식 전송 버튼 기능을 HTML의 form 태그를 이용해서 구현하고 자바스크립트를 이용한 유효성 검증은 onsubmit 이벤트 핸들러를 이용하여 수행하도록 수정한 것이다. 또한 버튼 이미지에는 대체 텍스트(로그인)를 제공하였다.

```

<script type="text/javascript">
function submitForm(form) {
    if (form.loginId.value && form.loginPassword.value) {
        return true;
    } else {
        return false;
    }
}
</script>
<form id="loginForm" name="loginForm" action="login.php" onsubmit="return
submitForm(this)">

```

```
<label>아이디 <input type="text" name="loginId"></label>
<label>비밀번호 <input type="password" name="loginPassword"></label><br>
<input type="image" src="login.gif" alt="로그인">
</form>
```