CHAPTER 2

모바일 웹 개발 환경 구축하기

이 장에서는 모바일 웹 개발 환경에 대해서 배울 것이다.

모바일 웹 개발을 위해서 필요한 도구들은 전통적인 웹 개발 도구와 다르지 않다. 모바일 웹 개발 환경에 필요한 도구로는 대표적으로 서버 측 스크립트 언어, 스크립트와 마크업을 지원하는 견고한 IDE, 환경에 맞게 설정할 수 있는 웹 서버, 그리고 웹 페이지를 테스트하고 표시 방식을 확인할 수 있는 웹 브라우저 등이 있다. 또한 프로젝트 진행 중 변경 내용을 관리하고 추적할 수 있도록 파일을 비교하고 소스 코드를 제어하는 도구를 사용할 것을 권장한다.

모바일 웹 개발자로서 여러분은 개발 환경에 맞춰 새로운 도구 사용법을 배우고, 웹 서버나 웹 브라우저 설정을 수정하는 데 익숙해져야 한다. IDE를 사용해서 모바일 웹 개발 프로젝트를 시작하면 문법 요소마다 구별된 색을 제공하기 때문에 구문을 한눈에 파악할 수 있고, 모바일 마크업 언어 자동 완성 기능을 사용해서 개발 효율성을 높일 수 있다. 하지만 웹 서버가 모바일 MIME 유형을 지원하게 하려면 개발자가 직접 웹 서버 설정을 확장해야만 한다. 오픈소스 브라우저인 파이어폭스는 매우 유연한 브라우저여서 설정을 변경하면 모바일 디바이스로 위장해 모바일 웹을 데스크톱에서 열람할 수 있다. 또한 개발자들은 특정 모바일 브라우저나 디바이스에 특화된 모바일 웹 화면을 좀 더 정확히 보기 위해 모바일 폰 에뮬레이터를 사용할 수 있다. 그리고 콘텐

츠와 실제 디바이스 사이의 호환성을 테스트하기 위해 모바일 특화된 테스트 도구를 사용해야 한다.

여러분은 다음 리스트에서 소개하는 세 가지 도구 중 하나를 사용해 다양한 모바일 상황에서 웹 페이지를 볼 수 있다. 리스트는 테스트 검증 신뢰도가 낮은 도구부터 높은 도구 순서로 나열했다.

1. 모바일 부가 기능add-on을 사용한 **파이어폭스**는 브라우저를 모바일 디바이스들로 위장해 모바일 웹 문서를 확인할 수 있다. 파이어폭스는 모바일 마크업을 테스트하는 편리한 개발자 도구지만 실제 모바일 디바이스의 표현 방식을 묘사하기에는 부족하다. 그래서 이 방법은 개발자 수준의 테스트로만 적합하다.
2. **모바일 브라우저 에뮬레이터**는 실제 모바일 브라우저 코드를 실행시켜 문서 표현 방식을 포함한 모든 브라우저 기능을 시뮬레이션한다. 이것은 실제 모바일 브라우저의 행동 방식과 거의 비슷하다. 하지만 모든 유형의 모바일 디바이스에 대해 해당 디바이스의 에뮬레이터를 사용할 수 있는 것은 아니다.
3. 말할 것도 없이 **실제 모바일 디바이스의 모바일 브라우저**를 사용하는 것이 모바일 웹 페이지의 행동을 테스트하기에는 가장 훌륭하다. 이 테스트 방법은 모바일 사용자와 웹 페이지 간의 상호작용을 확인할 수 있는 가장 정확한 방법이다. 모바일 웹 페이지 테스트에 관한 좀 더 자세한 정보를 얻고 싶다면 11장을 참고하라.

추천 IDE

IDE는 웹 애플리케이션 개발의 전 과정에서 사용되는 개발자 도구로 애플리케이션을 설계하고, 프로그래밍하고, 실행하고, 그리고 디버깅하는 소프트웨어이다.

모바일 웹 개발을 위한 IDE가 따로 정해져 있는 것은 아니다. IDE가 마크업 언어를 사용하는 웹 개발을 지원한다면 모바일 환경에서 사용하는 것이 가능하다. 여러분의 모바일 웹 프로젝트 상황에 맞게 IDE를 선택할 수 있다. 선택은 여러분의 몫이다. 웹

개발에서 요구하는 언어에는 웹 문서를 만들 때 사용하는 마크업 언어와 서버 측 런타임 언어뿐만 아니라 웹 문서의 스타일을 정의하는 CSS와 클라이언트 측과 상호작용 기능을 구현하는 자바스크립트 파일도 있다.

모든 IDE는 문법 요소에 따라 다른 색상을 제공하는 기능, 그리고 구문 자동 완성 기능을 제공해야 한다. 또한 마크업과 스크립트 언어의 구문 에러를 표시해야 한다. CSS에 대한 구문 에러 표시를 일부분 제공하고는 있지만 완전하지는 않다. IDE 중에는 다른 IDE에서 설정한 웹 개발 프로젝트 환경을 그대로 옮겨오는 기능을 제공하는 것도 있다. IDE를 선택할 때는 개발 생산성을 고려해야 하며, 만약 기존 IDE의 사용성에 만족하지 못한다면 IDE를 바꾸는 데 주저하지 말아야 한다. 여러분은 IDE에서 모바일 웹 개발 프로젝트를 만드는 데 많은 시간을 쓸 것이다. 그렇기 때문에 효율적인 웹 개발을 지원하는 IDE를 선택하는 것은 매우 중요하다.

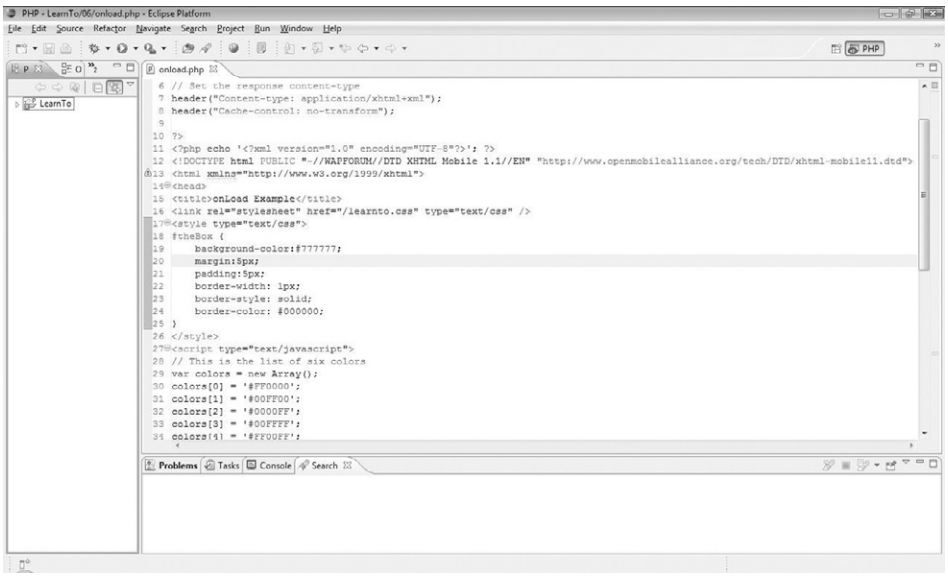
웹 개발자들은 보통 IDE를 선택할 때 서버 측 런타임 언어의 지원 여부를 따진다. 마크업 언어 지원은 안정화된 모든 IDE에서 가능하다. 표 2-1은 인지도가 높은 웹 개발 IDE에 대한 개발 플랫폼과 지원하는 런타임 언어를 정리한 것이다. 표에서 소개한 IDE는 모두 HTML, XHTML을 지원한다. 그리고 대부분의 모바일 마크업 언어들에 대한 최소한의 지원을 보장한다.

Adobe Dreamweaver와 같이 웹 디자인에 특화된 환경은 모바일 웹 페이지의 레이아웃을 만드는 데는 도움이 되지만 레이아웃을 모바일 웹 애플리케이션으로 구현하는 런타임 언어에 대한 지원은 부족하다.

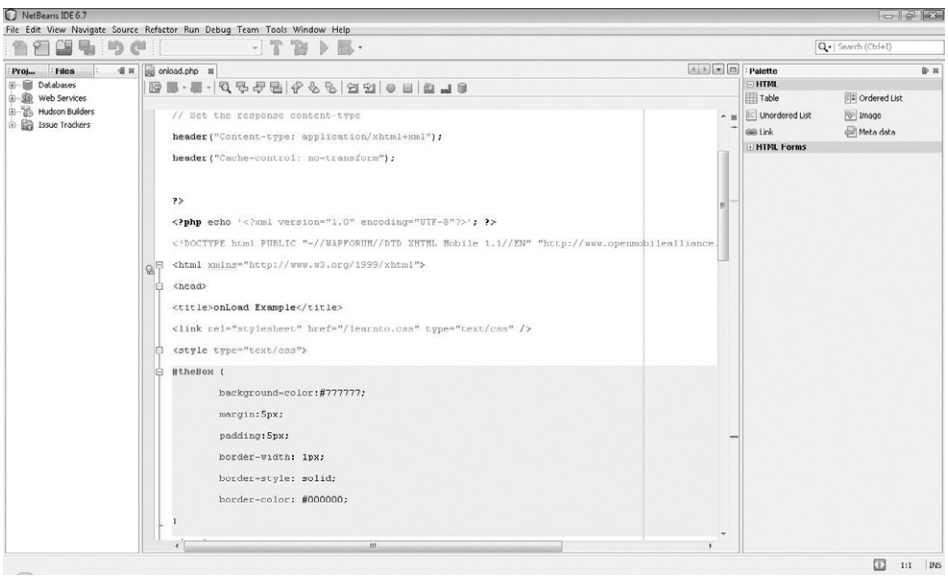
그림 2-1은 이클립스 PHP 개발 도구에서 본 XHTML-MP 웹 페이지를 구현한 PHP 템플릿의 스크린샷을 보여준다. 그림 2-2는 같은 PHP 템플릿을 NetBeans에서 본 모습이다.

:: **표 2-1** 유명한 웹 개발 IDE에 대한 플랫폼과 런타임 언어

IDE	개발 플랫폼	웹 런타임 언어	라이선스	URL
Eclipse	Windows Linux Mac OS	C/C++ Java PHP Python	무료 오픈소스	http://eclipse.org
Microsoft Visual Studio	Windows	.NET C# C/C++ Python (requires add-on) Ruby (requires add-on)	상용	http://www.microsoft.com/visualstudio/
Komodo	Windows Linux Mac OS	Perl PHP Python Ruby	상용	http://www.activestate.com/komodo/
NetBeans	Windows Linux Mac OS Solaris	C/C++ Groovy Java PHP Python Ruby	무료 오픈소스	http://netbeans.org
NuSphere PhpED	Windows	PHP	상용	http://www.nusphere.com/products/phped.htm
Aptana Studio	Windows Linux Mac	PHP Python Ruby on Rails	이중 라이선스, 무료 오픈소스 프로페셔널 버전을 위한 상용	http://aptana.com/studio
Zend Studio	Windows Linux Mac	PHP	상용	http://www.zend.com/en/products/studio



:: **그림 2-1** 이클립스 PHP 개발 도구 3.3에서 본 XHTML-MP 1.1 마크업 문서



:: **그림 2-2** NetBeans PHP 6.7에서 본 XHTML-MP 1.1 마크업 문서

모바일 MIME 유형

모바일 MIME 유형(콘텐츠 유형)은 모바일 웹 콘텐츠 형식을 알아보는 데 사용한다. 모바일 콘텐츠 종류는 매우 다양하다. 모바일 마크업을 포함하는 텍스트 문서 또는 볼 수 있고 플레이할 수 있는 콘텐츠를 포함하는, 예를 들어 휴대폰 벨소리, 바탕화면, 비디오 같은 바이너리 파일들, 그리고 실행 가능한 바이너리 모바일 애플리케이션 등으로 나눌 수 있다. HTTP 트랜잭션 중에 웹 서버와 브라우저 클라이언트는 MIME 유형을 사용해 콘텐츠 형식을 구별한다.

모바일 웹 브라우저와 웹 서버 사이에서 HTTP 트랜잭션을 실행하는 동안 몇 가지 방법으로 MIME 유형을 사용한다(코드 변환기 또는 프럭시에 의해 몇 가지 방법으로 사용된다).

- **모바일 브라우저**Mobile Browser: 모바일 브라우저는 `Accept` HTTP 요청 헤더 값을 지원되는 MIME 형식 리스트로 설정하여 보낸다. `Accept` 요청 헤더는 해당 모바일 디바이스에서 지원되는 콘텐츠 유형을 알려준다. 불행하게도 어떤 모바일 디바이스가 보내는 요청 헤더는 불확실한 정보를 포함하는 것으로 알려져 있다. 웹 서버는 이 헤더 값과 모바일 디바이스 특성 데이터베이스를 고려하여 HTTP 응답에 실어 보낼 가장 적합한 콘텐츠를 결정한다.
- **웹 서버**Web Server: 웹 서버 설정 정보를 수정하여 모바일 MIME 유형과 모바일 콘텐츠의 파일 확장자를 연관시킬 수 있다(웹 서버들은 보통 모바일 MIME 유형을 지원하는 설정이 미리 되어 있다. 그러나 웹 마스터는 반드시 수동으로 추가적인 모바일 MIME 유형을 추가해야 한다). 웹 문서와 연관된 MIME 유형은 HTTP 응답에서 `Content-Type` 헤더의 값으로 사용된다. 웹 서버가 모바일 브라우저로 올바른 모바일 MIME 유형을 헤더에 넣어 파일과 함께 전송했다면 모바일 브라우저는 그 파일을 어떻게 해석해야 되는지 알 수 있다. 파일은 웹 페이지, 모바일 애플리케이션, 바탕화면, 휴대폰 벨소리, 비디오 등이 될 수 있다(브라우저가 자체적으로 파일을 표시할 수 없는 경우에는 사용자에게 파일을 저장하거나 설치할 수 있는 창을 띄울 것이다).
- **코드 변환기와 게이트웨이**Transcoder and Gateway: 모바일 디바이스와 웹 서버 사이에는 종종 코드를 변환시키는 HTTP 프럭시 서버가 존재한다. 이 프럭시 서버들

은 디바이스를 대신해서 원본 콘텐츠를 조작하는 방법을 결정하기 위해 Accept와 Content-Type 헤더를 조사할 수 있다. 예를 들어, PNG 이미지는 GIF 이미지로 코드가 변경될 수 있다. 또는 HTMP 문서는 XHTML로 변환될 수 있다.

- **서버 측 런타임 언어**Server-Side Runtime language: 문서와 연관된 MIME 유형을 서버 측 런타임 언어를 이용해 오버라이드할 수 있다. 리스트 2-1은 XHTML-MP 문서를 포함한 HTTP 응답의 MIME 유형을 설정하기 위해 스크립트의 첫 부분에서 내장 함수를 사용하는 간단한 PHP 코드를 보여준다.

리스트 2-1 PHP를 사용하여 HTTP 응답의 MIME 유형 설정

```
<?php
header('Content-Type: application/vnd.wap.xhtml+xml');
?>
```

웹 서버의 모바일 MIME 유형 설정은 웹 콘텐츠에 대한 모바일 접근성 측면에서 중요하다. 모바일 브라우저는 HTTP 응답의 MIME 유형을 사용해 웹 문서가 해당 브라우저에서 보일 수 있는지, 혹은 모바일 OS 컴포넌트나 네이티브 모바일 애플리케이션을 실행시켜 보여줄 수 있는지를 판단한다.

표 2-2는 모바일 웹에서 일반적으로 사용되는 몇 가지 파일 유형에 대한 MIME 유형 리스트이다.

:: **표 2-2** 일반적으로 사용하는 모바일 파일 종류와 관련된 MIME 유형

MIME 유형	파일 확장자	콘텐츠	사용
application/vnd.wap.xhtml+xml application/xhtml+xml	xhtml	XHTML-MP markup	모바일 웹 페이지
text/html	html(윈도우 서버에서는 htm)	HTML markup	HTML을 지원하는 스마트폰과 모바일 디바이스용 모바일 웹 페이지
text/css	css	CSS1, CSS2, 그리고 무선 CSS	모바일 웹 문서에서 캐스케이딩 스타일시트

application/javascript text/javascript	js	JavaScript	HTML, XHTL-MP 1.1, 그리고 1.2와 함께 사용되는 스크립트 언어
multipart/mixed	–	MIME 멀티파트 인코딩 문서	단일 HTTP 응답으로 마크업과 이와 관련된 자원들(이미지, CSS, 스크립트 등)을 다운로드하는 것을 허용함
text/vnd.wap.wml	wml	WML markup	성능이 낮은 구식 모바일 디바이스를 위한 가벼운 모바일 웹 페이지
text/vnd.wap.wmlscript	wmls	WML Script	WML과 함께 사용되는 스크립트 언어
audio/mp3 audio/mpeg	mp3	MP3 audio	벨소리와 음악 전체
audio/x-midi	midi	MIDI audio	벨소리
image/gif	gif	GIF image	바탕화면
image/jpg image/jpeg	jpg jpeg	JPG image	바탕화면
image/png	png	PNG image	바탕화면
video/3gpp	3gp	3GP video	모바일 비디오
video/mp4	mp4	MPEG4 video	모바일 비디오

웹 서버 설정

모바일 마크업과 모바일 문서 파일 확장자는 MIME 유형과 연관된다. 모바일 웹 콘텐츠를 호스팅하는 웹 서버는 설정된 내용을 바탕으로 모바일 파일 확장자와 연관된 MIME 유형을 정확히 결정한다. 웹 서버에 새로운 MIME 유형을 추가하는 설정은 웹 서버 모델마다 다르다.

아파치

아파치Apache 웹 서버는 mode_mime 설정에서 AddType 지시어를 사용해서 새로운 MIME 유형을 mime.types, httpd.conf, 또는 .htaccess 설정 파일에 추가한다.

아파치의 mime.types과 httpd.conf 설정 파일은 전역 웹 서버 설정 파일이다. 이 파일들은 아파치 웹 서버의 전체 행동을 제어한다.

아파치는 .htaccess 파일을 로컬 또는 디렉토리에 특화된 설정을 위해 사용한다. .htaccess 파일의 설정 내용은 그것이 위치한 디렉토리(모든 하위 디렉토리를 포함하여) 범위에 영향을 준다. 아파치 프로젝트(http://httpd.apache.org)는 전역 아파치 설정 파일에 접근이 제한됐을 때만 .htaccess 파일을 사용할 것을 권장한다(예를 들어, 사용자가 자신의 웹 문서에 대한 제어 권한만을 가지고 있고 전체 아파치 서버에 대한 권한이 없는 공유 웹 호스팅 환경에서 .htaccess를 사용할 것을 추천한다). 하지만 .htaccess 파일 사용을 남용하면 서버 성능에 영향을 줄 수 있다.

AddType 설정 지시어는 다음과 같은 구문을 사용해 새로운 모바일 MIME 유형과 연관된 파일 확장자 리스트를 명시한다.

```
AddType <MIME type> <file extension> [<file extension>] ...
```

리스트 2-2는 새로운 모바일 MIME 유형 지원을 추가하는 .htaccess 설정 파일을 보여준다. 이 파일은 .xhtml과 .xhtm 파일 확장자를 XHTML-MP 마크업에 대한 MIME 유형과 연관시킨다. 비슷하게, .wml 파일 확장자는 WML 마크업에 대한 MIME 유형과 연결됐다. 반면, .wmls 파일 확장자는 WML 스크립트에 대한 MIME 유형과 짝지어졌다.

리스트 2-2 모바일 MIME 유형에 대한 아파치 .htaccess 설정

```
# XHTML-MP, WML, 그리고 WML 스크립트 파일 확장자를 위한 모바일 MIME 유형 추가
AddType application/vnd.wap.xhtml+xml .xhtml .xhtm
AddType text/vnd.wap.wml .wml
AddType text/vnd.wap.wmlscript .wmls
```

Microsoft IIS

Microsoft IIS Internet Information Services 웹 서버는 파일 확장자와 MIME 유형 연결을 관리할 수 있도록 사용자 인터페이스, 커맨드라인, 그리고 프로그래밍적 메소드를 제공한다.

MIME 유형 설정을 위해 IIS 관리 애플리케이션을 사용하는 것과 더불어 IIS 7은 MIME 유형 관리를 위한 커맨드라인 구문을 소개한다. 리스트 2-3은 XHTML-MP 마크업을 포함하는 .xhtml 파일을 "application/vnd.wap.xhtml+xml" MIME 유형에 연관시키는 명령어를 보여준다.

리스트 2-3 커맨드라인에서 Microsoft IIS 7로 MIME 유형 추가

```
appcmd set config /section:staticContent ^
/+"[fileExtension='.xhtml',mimeType='application/vnd.wap.xhtml+xml']"
```

다음 링크는 Microsoft TechNet에 올라온 글이다. IIS 관리 애플리케이션 또는 스크립트를 이용한 MIME 유형 관리에 대한 좀 더 자세한 정보를 제공한다.

- IIS 4와 5: http://technet.microsoft.com/en-us/library/bb742440.aspx
- IIS 6: www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/cd72c0dc-c5b8-42e4-96c2-b3c656f99ead.mspx?mfr=true
- IIS 7: http://technet.microsoft.com/en-us/library/cc725608(WS.10).aspx

Nginx

Nginx(http://nginx.net/)는 정적 파일을 제공하는 데 매우 적합할 뿐 아니라 가벼운 웹 서버이다. 사용자가 많은 사이트 환경에서는 일반적으로 정적 페이지를 처리하는 nginx 뒤에서 동적 페이지 생성 작업을 위임 받아 지원하는 아파치 또는 IIS 서버를 사용한다. Nginx는 이미지, 스타일시트, 그리고 리소스 요구가 없는 정적 파일들을 다룬다.

여러분은 types 지시어를 사용해 nginx.conf에서 MIME 유형을 설정할 수 있다. 리스트 2-4는 .xhtml 파일 확장자를 XHTML-MP 마크업 MIME 유형과 연관시키고, .wml 파일 확장자를 WML 마크업 MIME 유형과 연결하는 명령어 예제를 보여준다.

리스트 2-4 Nginx에 MIME 유형 추가하기

```
types {
  application/vnd.wap.xhtml+xml    xhtml;
  text/vnd.wap.wml    wml;
}
```

데스크톱에서 모바일 웹 브라우저

모바일 웹 개발 환경은 모바일 환경에서 웹 페이지를 열람하는 것으로 위장하는 도구를 갖춰야 완성된다. 모바일 웹 페이지를 보기 위한 가장 편리한 개발자 도구는 파이어폭스이다. 파이어폭스는 특정 모바일 디바이스로 위장하는 설정 방법이 있고, 모바일 디바이스에서 웹 콘텐츠가 표현되는 방식과 비슷하게 제공해 준다. 모바일 디바이스 에뮬레이터는 모바일 디바이스와 브라우저를 대신할 수 있는 데스크톱 소프트웨어이다. 여러분은 실제 모바일 디바이스 행동 방식에 좀 더 가까이 다가갈 수 있다. 에뮬레이터는 모바일 디바이스에서 사용하는 것과 같은 브라우저 엔진을 사용한다. 그러나 같은 실행 환경에서 수행되는 것이 아니기 때문에 클라이언트 측 성능 문제를 추적할 때는 이 접근방식을 사용할 수 없다. 모바일 환경에서 웹 페이지를 보는 가장 정확한 방법은 실제 모바일 디바이스를 사용하는 것이다. 실제 모바일을 사용해야 네트워크 지연과 프럭시 효과를 볼 수 있기 때문이다.

Note 만약 여러분이 아이폰, 안드로이드 또는 노키아 Series 60 3rd Edition 또는 그 이후 버전 모바일 디바이스에 우선순위를 둔 모바일 웹 콘텐츠를 개발해야 한다면, 여러분은 애플 사파리 (http://apple.com/safari/) 또는 구글 크롬(http://google.com/chrome)과 같은 WebKit 브라우저를 사용한 테스트가 필요하다. 이 모바일 디바이스는 WebKit 기반 브라우저를 사용한다. 사파리와 크롬이 이에 속하는 브라우저이다. WebKit 기반 브라우저는 user-agent 수정을 허용하고 코드 점검 도구를 제공한다.

파이어폭스와 모바일 부가 기능

파이어폭스는 모질라 협회의 표준 준수, 오픈소스 웹 브라우저이다. 특히 파이어폭스는 사용자가 쉽게 부가 기능을 사용해서 환경에 맞게 브라우저 기능을 확장할 수 있다. 파이어폭스 부가 기능은 XUL(XML User Interface Language)(https://developer.mozilla.org/en/XUL) 프로그래밍 언어로 개발된 확장 모듈이다. 모질라는 부가 기능 디렉토리를 https://addons.mozilla.org/에서 호스팅하고 있다.

> Note 여러분 중에 궁금해할 사람들을 위해서 짧게 언급하자면, XUL 용어는 zool로 발음된다. 사실 XUL은 고전 괴짜 영화, 『Ghostbusters』에서 등장하는 신인 Zuul에 빗대어 묘사된다. 그 영화에서 시고니 위버(Signourney Weaver)가 맡은 역할은 Zuul에게 영혼을 잡아먹힌 후 아주 유명한 대사를 한다. "Dana는 이제 없다. 오직 Zuul이 있을 뿐이다"

프로그래밍 언어 XUL은 애플리케이션 로직을 구현하는 사용자 인터페이스와 자바스크립트를 정의하기 위해 XML을 사용했다. XUL의 구호는 "data는 이제 없다. 오직 XUL이 있을 뿐이다"이다.

이 이야기로 더 관심이 생겼다면, XUL 문서에서 사용한 XML 네임스페이스를 리스트 2-5에서 확인해 보자. 개발자들의 유머감각도 꽤 쓸 만한 수준인 것 같다.

리스트 2-5 파이어폭스 확장 모듈에서 사용한 XML 기반 XUL 프로그래밍 언어에서 네임스페이스 URI

```
http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul
```

다음 단계는 파이어폭스가 모바일 디바이스로 위장해서 모바일 마크업 언어로 작성된 웹 문서를 표시하는 데 필요한 몇 가지 부가 기능들을 다운받아서 설치하는 것이다. http://www.mozilla.com/en-US/products/firefox/에서 최신 파이어폭스 버전을 다운받는 것으로 시작하자. 다음에 이어지는 절에서는 부가 기능을 어떻게 설치하고 사용하는지 설명할 것이다. 모든 부가 기능을 설치한 후 실행시키기 위해서는 파이어폭스를 재실행한다.

확장 모듈을 설치하고 파이어폭스를 재실행한 후에 파이어폭스에서 Tools → Add-ons 대화 상자를 연다. 그림 2-3은 몇 가지 확장 모듈이 설치된 부가 기능 대화 상자를 보여준다.

:: **그림 2-3** 모바일 부가 기능이 설치된 파이어폭스 3.0.11에서 Add-ons 대화 상자

XHTML Mobile Profile

XHTML Mobile Profile 부가 기능은 `application/vnd.wap.xhtml+xml` MIME 유형 지원을 돕는다. 이 부가 기능으로 파이어폭스는 XHTML-MP 모바일 웹 문서를 브라우저 창에서 표시할 수 있다. 이 확장 모듈이 없는 파이어폭스는 XHTML-MP 문서를 파일로 저장하도록 사용자에게 창을 띄운다.

XHTML Mobile Profile 부가 기능 버전 0.5.3은 `multipart/mixed` MIME 유형에 대한 지원도 추가했다. 멀티파트 인코딩은 모바일 마크업에 최적화된 마크업 문서와 의존

리소스(이미지, CSS 스크립트 등등)를 단일 HTTP 응답 하나로 묶어 전송할 수 있게 한다. 상당수의 모바일 브라우저와 사업자들이 모바일 웹 콘텐츠를 위해 멀티파트multipart 인코딩을 지원하다. 이 부가 기능은 멀티파트 인코딩된 HTML 컴포넌트를 브라우저가 표시할 수 있게 한다. 멀티파트 인코딩에 대한 좀 더 자세한 정보는 9장을 참고하라.

파이어폭스에서 https://addons.mozilla.org/en-US/firefox/addon/1345 사이트를 열람한 후 Add to Firefox 버튼을 클릭하면 XHTML Mobile Profile을 설치할 수 있다. 이 부가 기능은 옵션이나 설정이 따로 필요 없다. 기능을 활성화시키면 브라우저에서 XHTML-MP와 멀티파트 인코딩된 모바일 마크업을 볼 수 있다.

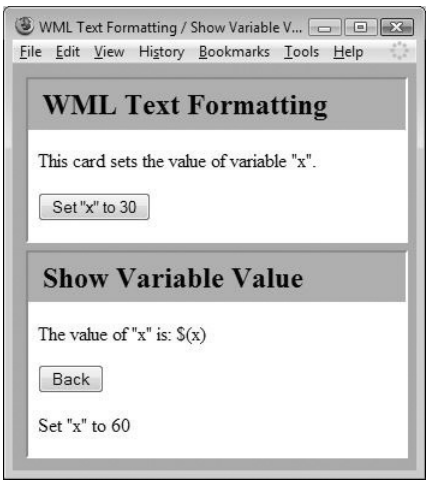
wmlbrowser

wmlbrowser 부가 기능은 파이어폭스가 WML 문서를 브라우저에서 표시하는 것을 허용한다. 이 부가 기능의 WML 렌더링은 WML 코드로 웹 문서를 개발하고 디버깅하는 데 적합하고, 실제 모바일 디바이스 수준으로 WML 문서의 레이아웃을 비슷하게 볼 수 있다. 하지만 여전히 에뮬레이터나 실제 모바일 디바이스에서 테스트해 볼 필요가 있다. WML에 대해서는 3장에서 더 자세히 배울 것이다.

이 확장 모듈이 없는 파이어폭스는 사용자에게 WML 문서를 파일로 저장하도록 창을 띄울 것이다.

파이어폭스에서 https://addons.mozilla.org/en-US/firefox/addon/62를 통해 부가 기능을 설치할 수 있다.

이 부가 기능은 추가적인 옵션이나 설정이 필요하지 않다. 이 기능을 활성화시키면 브라우저 창에서 WML 모바일 마크업을 볼 수 있다. 여러분은 wmlbrowser를 사용해서 샘플 WML 문서 스크린 샷을 http://learnto.mobi/02/wml-var.wml에서 볼 수 있다(그림 2-4).



:: **그림 2-4** wmlbrowser 부가 기능을 설치한 파이어폭스 3.0.11에서 본 카드 두 장을 구현한 WML 문서

User Agent Switcher

User Agent Switcher 부가 기능은 파이어폭스가 user-agent 값을 변경하는 것을 가능하게 한다. user-agent 값은 HTTP 트랜잭션 중에 웹 서버가 브라우저를 식별하는 데 사용된다. 이 부가 기능은 사용자가 미리 정의한 user-agent 중 하나로 `User-Agent` HTTP 요청 헤더의 값을 설정한다. 모바일 웹 개발자들이 User Agent Switcher를 이용해 모바일 디바이스를 나타내는 user-agent 값으로 변경하게 된다. 웹사이트는 `User-Agent` 요청 헤더를 사용해 모바일 디바이스를 식별한 후 디바이스에 따라 맞춤 콘텐츠를 제공한다. 파이어폭스에서 모바일 디바이스로 위장하는 첫 번째 단계는 웹 요청을 보낼 때 특정 모바일로 인식되는 user-agent를 전송하는 것이다.

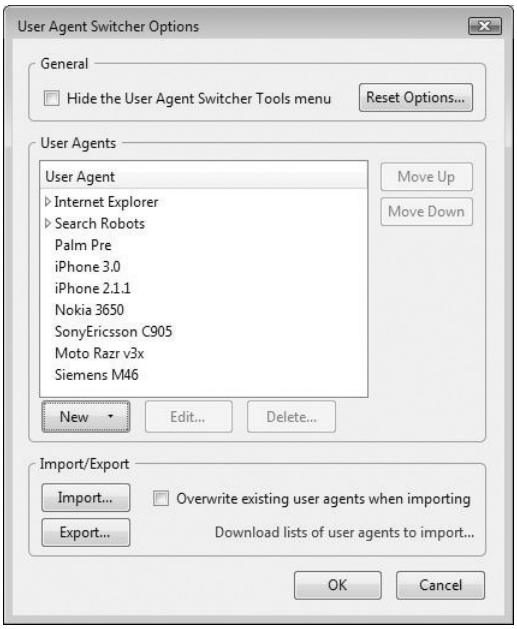
이 확장 모듈이 없는 파이어폭스는 항상 HTTP 요청에 디폴트 user-agent를 보낸다.

파이어폭스에서 `https://addons.mozilla.org/en-US/firefox/addon/526`를 열람하면 User Agent Switcher를 설치할 수 있다.

이 부가 기능은 사용자가 많은 모바일 디바이스에 대한 user-agent를 저장할 수 있도록 했다. 사용자는 부가 기능을 활성화한 후 Tools → Default User Agent 메뉴에서

쉽게 User-Agent HTTP 요청 헤더 값을 변경할 수 있다.

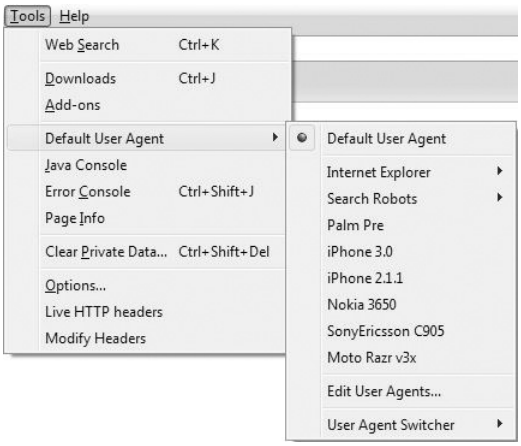
그림 2-5는 User Agent Switcher의 옵션 대화 상자 스크린 샷이다.



:: **그림 2-5** User Agent Switcher의 옵션 대화 상자

옵션 대화 상자는 Tools → Default User-Agent → Edit User-Agent 메뉴 항목을 선택하면 볼 수 있다. 이 대화 상자는 설치된 user-agent 리스트를 수정하고 볼 수 있도록 한다. 모바일 디바이스들에 대한 샘플 user-agent에 대해서는 부록 A를 참고하자. 또는 모바일 개발자 커뮤니티 mobiFoarge(http://mobiforge.com)에서 User Agent Switcher를 위해 모바일 user-agent를 모아 패키지 집합으로 제공한다. 블로그 포스트 http://mobiforge.com/ developing/blog/user-agent-switcher-config-file에서 첨부파일로 받아볼 수 있다.

그림 2-6은 User Agent Switcher에 설치한 user-agent 리스트에서 하나를 선택한 모습이다.



:: **그림 2-6** 파이어폭스 3.0.11의 Tools 메뉴를 사용해 user-agent 값 선택

User-Agent 요청 헤더 값을 추가된 값으로 변경하려면 Tools 메뉴를 사용하라.

Modify Headers

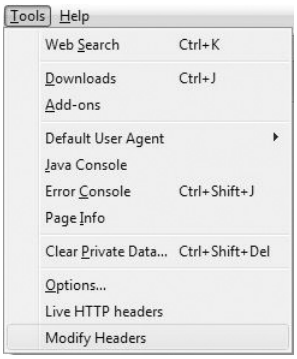
Modify Headers 부가 기능은 파이어폭스에서 보낸 HTTP 요청 헤더를 미리 정의한 규칙을 사용해 수정하는 것을 허용한다. 모바일 웹 개발자들은 이 확장 모듈을 사용해 파이어폭스의 요청 헤더를 모바일 디바이스가 보내는 것과 정확하게 일치하도록 수정하는 데 사용한다. 파이어폭스에서 모바일 디바이스로 위장하기 위한 두 번째 단계는 웹 요청을 보낼 때 모바일이 보내는 HTTP 요청 헤더 집합과 일치하게 보내는 것이다.

Modify Headers를 사용해 User-Agent 요청 헤더까지 수정하는 것도 가능하다. 하지만, User-Agent 헤더를 수정할 때는 좀 더 견고하고 안정된 지원이 가능한 User Agent Switcher 부가 기능을 사용할 것을 권장한다.

이 확장 모듈이 없는 파이어폭스는 디폴트 HTTP 요청 헤더 집합을 웹 서버로 전송할 것이다.

Modify Headers는 파이어폭스에서 https://addons.mozilla.org/en-US/firefox/addon/967을 통해 설치할 수 있다.

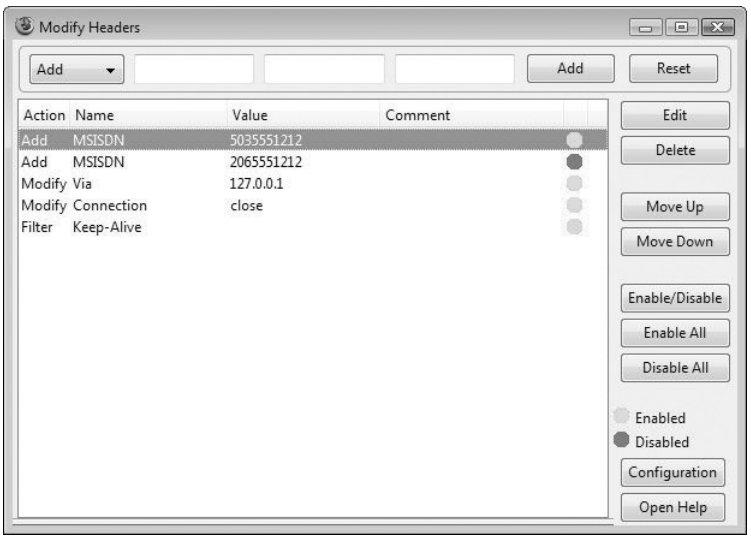
그림 2-7은 파이어폭스 도구 메뉴에서 Modify Headers 메뉴 항목을 보여준다.



:: **그림 2-7** Modify Headers 대화 상자는 파이어폭스 3.0.11의 Tools 메뉴에서 사용할 수 있다

Modify Headers 부가 기능은 파이어폭스에서 보내는 HTTP 요청 헤더를 변경하는 간단한 규칙을 정의하기 위해 대화 상자를 이용한다. 각 규칙은 각각 하나의 요청 헤더를 더하고 수정하고 또는 숨기는 작업을 한다. 규칙들은 순차적으로 실행되고, 개별적으로 활성화시키거나 숨겨질 수 있다.

그림 2-8의 예제 규칙들은 파이어폭스의 요청 헤더 몇 개를 변경시킨다.



:: **그림 2-8** 샘플 규칙 정의를 위한 Modify Headers 대화 상자

- MSISDN: 5035551212을 추가한다.
- Via 헤더를 Via: 127.0.0.1로 수정한다.
- Connection 헤더를 Connection: close로 수정한다.
- Keep-Alive 헤더를 제거한다.

규칙 중 두 번째 MSISDN 규칙은 비활성 처리됐기 때문에 수행되지 않는다.

추가적인 조작이 필요 없이 Modify Headers 대화 상자가 열려 있기만 하면 Modify Headers는 파이어폭스 요청 헤더를 변경한다. Configuration 버튼을 클릭하면 설정을 변경할 수 있다.

파이어폭스의 요청 헤더를 직접 보면서 모바일 디바이스로 위장하기 위한 헤더 수정 규칙을 작성하는 것이 도움이 될 수 있다. Live HTTP Headers 부가 기능은 HTTP 요청/응답 헤더를 파이어폭스 창에서 보여준다. 또는 파이어폭스에서 http://learnto.mobi/view.php를 열람하여 요청 헤더를 볼 수 있다.

모바일 디바이스의 샘플 HTTP 요청 헤더에 대해서는 부록 B를 참고하도록 하자.

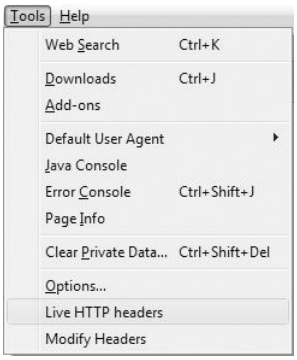
Live HTTP Headers

Live HTTP Headers 부가 기능을 사용해 사용자는 요청/응답 헤더를 볼 수 있다. 모바일 웹 개발자들은 HTTP 헤더를 다시 검사해 파이어폭스가 모바일 디바이스로 완전히 위장했는지 검사할 수 있다. 또한 이 과정을 통해 HTTP 트랜잭션을 더 깊이 이해할 수 있다.

이 확장 모듈이 없는 파이어폭스는 Tools → Page Info 대화 상자에서 요약된 헤더의 정보만을 확인할 수 있다.

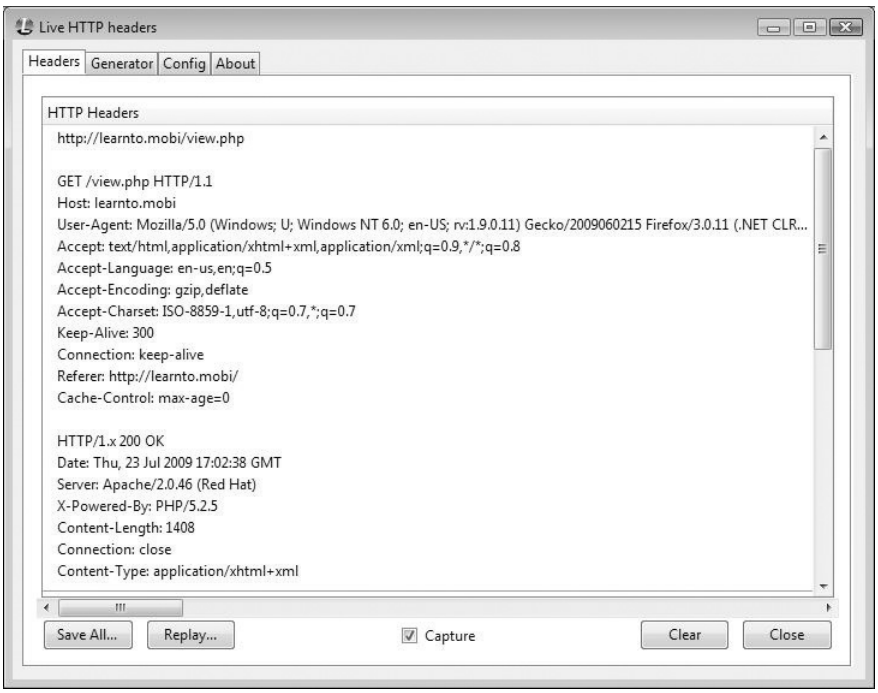
Live HTTP는 https://addons.mozilla.org/en-US/firefox/addon/3829에서 설치할 수 있다.

그림 2-9는 파이어폭스 도구 메뉴의 Live HTTP Headers 메뉴 항목을 보여준다.



:: **그림 2-9** 파이어폭스의 Tools 메뉴에서 볼 수 있는 Live HTTP Headers 대화 상자

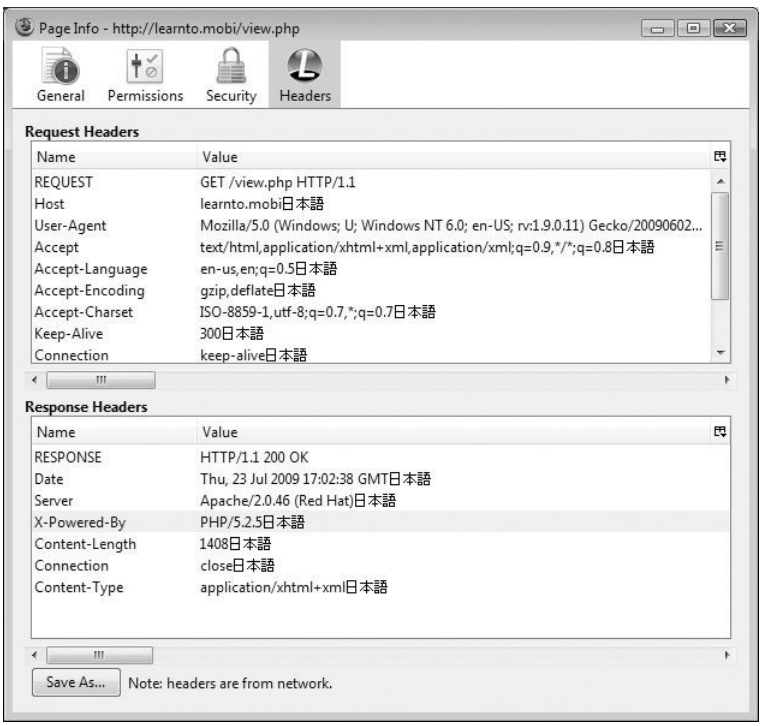
그리고 대화 상자를 열기 위해서 Tools → Live HTTP Headers 메뉴 항목을 선택하라 (그림 2-10).



:: **그림 2-10** 헤더를 보여주고 부가 기능 설정을 위한 Live HTTP Headers 대화 상자

Live HTTP Headers 대화 상자는 실시간으로 HTTP 요청/응답 스트림을 보여준다. 헤더 수집을 멈추기 위해서는 Capture 체크박스를 해제하면 된다. 설정 내용을 변경하기 위해서는 Config 탭을 사용하자. 환경 설정으로 개발자들은 어떤 헤더를 보여주고 저장할 것인지에 대해 웹 콘텐츠를 필터filter할 수 있다.

이 부가 기능은 파이어폭스의 Page Info 대화 상자에 웹 문서의 요청/응답 헤더를 보여주는 화면 한 장pane을 추가했다.



:: **그림 2-11** 파이어폭스의 Page Info 대화 상자에 헤더 정보를 보여주는 Live HTTP Headers

Small Screen Renderer

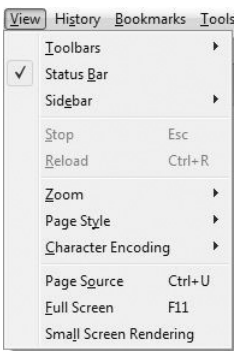
Small Screen Renderer 부가 기능은 모바일 디바이스 화면처럼 작은 크기의 웹 페이지를 표시할 화면을 만든다. 이 부가 기능은 웹 페이지가 모바일 디바이스의 작은 화면

에서 어떻게 표현될지 미리 확인하기 위해 설계됐다. 모바일 웹 개발자는 이 부가 기능을 사용해서 그들의 페이지 레이아웃이 작은 화면에 적합한지 어느 정도 확신을 가질 수 있다. 하지만 에뮬레이터나 실제 모바일 디바이스에서 테스트하는 것이 훨씬 정확히 페이지 표시 결과를 얻을 수 있는 방법이다.

이 확장 모듈이 없는 파이어폭스는 웹 페이지를 브라우저 창 전체에 맞춰 보여준다. Small Screen Renderer는 `https://addons.mozilla.org/en-US/firefox/addon/526`에서 설치할 수 있다.

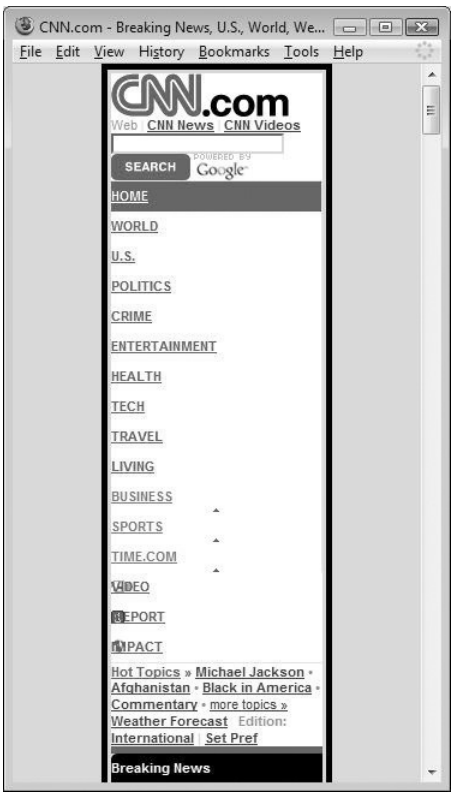
이 부가 기능은 추가적으로 필요한 옵션이나 설정할 내용이 없다. 기능이 활성화되면 파이어폭스의 View 메뉴에서 작은 화면을 껐다 켰다 할 수 있다.

파이어폭스에서 View → Small Screen Rendering을 선택해서 웹 페이지를 작은 화면에 표시하거나 아니면 원래 화면에 표시할 수 있다(그림 2-12).



:: **그림 2-12** 파이어폭스의 View 메뉴에서 Small Screen Rendering 토글링

Small Screen Rendering 부가 기능을 사용해 웹 페이지를 모바일 디바이스에서 어떻게 보이는지 확인할 수 있다. 예를 들어, 그림 2-13은 CNN의 데스크톱 웹사이트를 파이어폭스의 작은 화면에서 본 것이다.



:: **그림 2-13** 파이어폭스에서 Small Screen Rendering으로 본 www.cnn.com

Firebug

Firebug 부가 기능은 어떤 웹 페이지에서든지 HTML, CSS, 그리고 자바스크립트를 편집하고 디버깅할 수 있게 해준다. Firebug는 웹사이트 성능 향상을 위한 검사와 DOM 검사를 지원한다. 모바일 웹 개발자들은 마크업과 스타일 구문에 대한 재검사, 에러 검출, 클라이언트 측 스크립트 디버그, 성능 개선을 위해 Firebug를 사용한다.

이 확장 모듈이 없는 파이어폭스는 브라우저 창에서 보이는 웹 콘텐츠를 편집하지 못한다. 다른 확장 모듈은 자바스크립트 디버깅, DOM 검사, 그리고 성능 지표 등 각각의 기능을 개별적으로 제공하긴 하지만 Firebug처럼 다양한 기능을 통합해서 제공하지는 않는다. Firebug는 여러 웹 개발 도구들이 가지고 있는 중요한 기능들을 단일 확

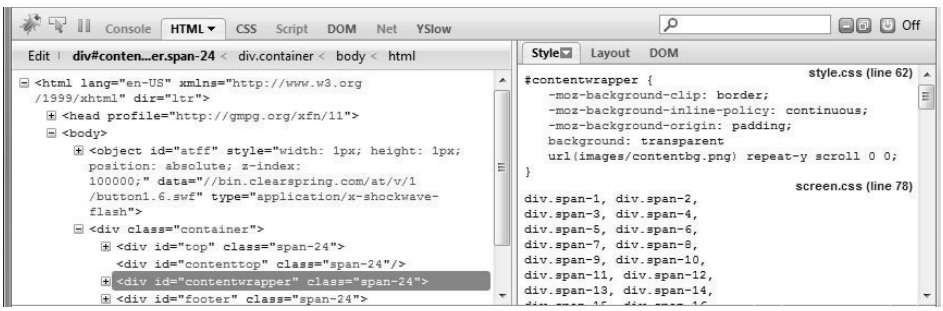
장 모듈로 통합해서 제공할 뿐 아니라 파이어폭스 브라우저에 깨끗하게 결합할 수 있어 사용이 용이하다.

Firebug는 파이어폭스의 https://addons.mozilla.org/en-US/firefox/addon/1843에서 설치할 수 있다. 또는 Firebug 웹사이트 http://getfirebug.com/을 방문해서 설치할 수 있다.

이 부가 기능은 검사할 웹 콘텐츠의 유형과 웹 페이지를 제어하는 옵션을 가지고 있다. 그림 2-14는 파이어폭스의 Tools → Firebug 메뉴를 보여준다.

:: **그림 2-14** 파이어폭스의 Tools → Firebug 메뉴

Firebug를 사용하려면 F12를 누르거나, Tools → Firebug → Open Firebug 메뉴 항목을 사용해라. 그림 2-15는 데스크톱 웹 페이지의 HTML과 CSS를 보여주는 Firebug 모습이다. 그 화면에서 페이지 마크업의 편집까지 가능하다.



:: **그림 2-15** HTML과 CSS의 구문을 확인하고 편집할 수 있는 Firebug 인터페이스

모바일 브라우저 에뮬레이터

모바일 디바이스 OEM과 소프트웨어 사업자들은 모바일 디바이스와 브라우저에 대한 에뮬레이터 제품을 제공한다. 개발자들은 이 에뮬레이터를 사용해 모바일 애플리케이션을 실행하거나 모바일 웹을 열람하도록 시뮬레이션한다. 모바일 웹 개발자는 모바일 에뮬레이터를 가지고 모바일 디바이스에서 웹 페이지가 어떻게 표시되는지 대체로 정확하게 파악할 수 있고, 그 정보를 바탕으로 디버깅이 가능하다. 하지만 역시 가장 훌륭한 디버깅 방법은 실제 모바일 디바이스에서 모바일 웹 애플리케이션을 테스트하는 것이다.

모바일 웹 개발 과정에서 사용되는 세 가지 유형의 모바일 에뮬레이터가 있다.

- **모바일 디바이스 에뮬레이터**Mobile device emulator는 모바일 폰 OS와 모바일 애플리케이션을 실행했을 때 모바일 디바이스가 어떻게 동작하는지 시뮬레이션한다. 개발자들은 네이티브 기능에 접근하고, 모바일 브라우저를 사용하고, 또한 외부 애플리케이션을 실행하고 디버깅할 수 있다. 모바일 디바이스 OEM 측은 디바이스 에뮬레이터를 모바일 개발자 커뮤니티에 무료로 제공하고 있다.
- **모바일 브라우저 에뮬레이터**Mobile browser emulator는 오직 모바일 브라우저 애플리케이션만을 시뮬레이션 한다. 이 에뮬레이터는 모바일 디바이스에서 동작하는 것과 같은 애플리케이션을 사용해 사용자에게 웹 페이지 표시 결과를 보여준다. 브라우저 사업자들은 브라우저 에뮬레이터를 모바일 개발자 커뮤니티에 무료로 제공하고 있다.
- **모바일 기반 시설 에뮬레이터**Mobile infrastructure emulator는 모바일 디바이스와 모바일 에코시스템에서 의존관계에 있는 서비스를 시뮬레이션한다. RIM이 생산한 블랙베리Blackberry 스마트폰은 이메일과 인터넷 접근을 관리하기 위해 기업 서버와 통신한다. RIM은 모바일 개발자가 블랙베리 사용 환경을 처음부터 끝까지 테스트할 수 있도록 블랙베리 디바이스와 기업 서비스를 시뮬레이션하는 에뮬레이터를 제공한다.

모바일 웹 개발자들은 그들의 웹 페이지가 다양한 종류의 모바일 디바이스와 브라우저에서 각각 어떻게 표시되는지 분석하기 위해 여러 가지 모바일 에뮬레이터를 사용하게 될 것이다. 그림 2-16~22는 CNN 모바일 웹사이트가 각각의 모바일 에뮬레이터에서 표시되는 모습이다.

:: **그림 2-16** 아이폰 에뮬레이터에서 본 CNN 모바일

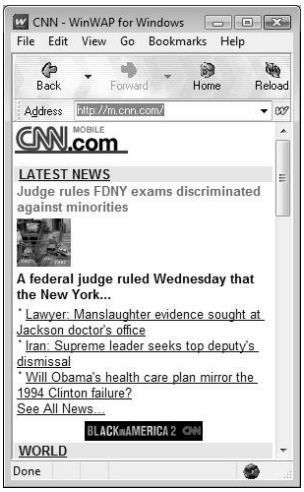
:: **그림 2-17** 안드로이드 1.6 에뮬레이터에서 본 CNN 모바일

:: **그림 2-18** 팜 프리 에뮬레이터에서 본 CNN 모바일

:: **그림 2-19** 오페라 미니 에뮬레이터에서 본 CNN 모바일

:: **그림 2-20** Openwave v7 에뮬레이터에서 본 CNN 모바일

:: **그림 2-21** dotMobi 에뮬레이터에서 본 CNN 모바일(Nokia N70 스킨 사용)



:: **그림 2-22** WinWAP 에뮬레이터에서 본 CNN 모바일

표 2-3은 모바일 웹 개발에서 사용되는 모바일 에뮬레이터 리스트이다. 여러분은 모바일 에뮬레이터의 가장 최신 리스트를 http://learnthemobileweb.com/mobile-web-emulators/를 방문에 얻을 수 있다.

:: **표 2-3** 모바일 디바이스, 브라우저, 그리고 기반 구조(인프라) 에뮬레이터

에뮬레이터 이름	에뮬레이터 유형	URL	설명
3	3	http://developer.palm.com/	SDK는 팜프리 에뮬레이터를 포함한다.
iPhone SDK	디바이스	http://developer.apple.com/iPhone/program/	아이폰 시뮬레이터를 포함한다. Mac OS X에서만 구동한다.
Android SDK	디바이스	http://developer.android.com/sdk/	SDK는 안드로이드 에뮬레이터를 포함한다.
Windows Mobile 6 SDK	디바이스	http://msdn.microsoft.com/enus/windowsmobile/bb264327.aspx	SDK는 윈도우 모바일 디바이스의 스킨과 에뮬레이터를 포함한다.
Nokia Mobile Browser Simulator 4.0	브라우저	www.forum.nokia.com/info/sw.nokia.com/id/db2c69a2-4066-46ff-81c4-caac8872a7c5/NMB40_install.zip.html	노키아 디바이스에서 표시하는 XHTML, XHTML-MP, 그리고 WML을 보여준다.

:: **표 2-3** (계속)

에뮬레이터 이름	에뮬레이터 유형	URL	설명
RIM Blackberry Simulators	디바이스 기반 시설	www.blackberry.com/developers/downloads/simulators/index.shtml	블랙베리와 블랙베리 엔터프라이즈 서버에 대한 시뮬레이터이다.
Opera Mini Simulator	브라우저	www.opera.com/mini/demo/	J2ME 또는 BREW 런타임 환경을 지원하는 일반폰에서 고급 모바일 브라우징을 위해 Opera Mini와 프럭시 솔루션을 시뮬레이션한다.
YoSpace SmartPhone Emulator	디바이스	www.yospace.com/index.php/spedemo.html	상용 디바이스 에뮬레이터 데모
Openwave Phone Simulator V7	브라우저	http://developer.openwave.com/dvl/tools_and_sdk/phone simulator/	일반폰에서 설치된 점유율이 높은 모바일 브라우저 에뮬레이터. XHTML과 WML을 표현한다.
WinWAP Simulator	브라우저	www.winwap.com/desktop_applications/browser_emulator	윈도우 모바일 디바이스의 WML 모바일 브라우저를 시뮬레이션한다. 트라이얼(trial)을 30일 동안 무료로 사용할 수 있다.

실제 모바일 디바이스

다시 말하지만, 모바일 웹 페이지를 테스트하고 확인하는 가장 훌륭하고 확실한 방법은 실제 모바일 디바이스를 사용하는 것이다. 모바일 웹 개발자들은 개발 기간 동안 디바이스 종류에 의존하는 기능 테스트를 위해 다양한 폰을 곁에 두게 된다. 데스크톱 소프트웨어 대신에 실제 배터리를 사용하는 모바일 디바이스를 사용하기 때문에 디바이스 종류별로 테스트를 하면서 파이어폭스와 에뮬레이터 테스트에서 찾을 수 없었던 모바일 브라우저의 렌더링과 성능 문제를 찾을 수 있다.

만약 모바일 폰 서비스와 디바이스에 투자하는 것이 어렵다면 DeviceAnywhere (http://deviceanywhere.com) 같은 상용 대체 서비스를 사용해 네트워크로 모바일 디바이스에 접근할 수 있다. 모바일 디바이스 제조사와 네트워크 사업자들은 자주 개발자

와 파트너 프로그램 회원들에게 디바이스 값을 빌려주거나 디바이스를 대여해 준다. 모바일 디바이스에서 모바일 웹 콘텐츠를 테스트하는 방법에 대해 좀 더 많은 정보는 11장에서 얻을 수 있다.

다른 개발 도구들

파일 비교와 소스 코드 제어 유틸리티는 모바일 웹 개발에 빠뜨릴 수 없는 유용한 도구이다. 이 도구는 IDE에 통합돼서 사용하거나 분리된 단일 소프트웨어 제품으로도 사용 가능하다.

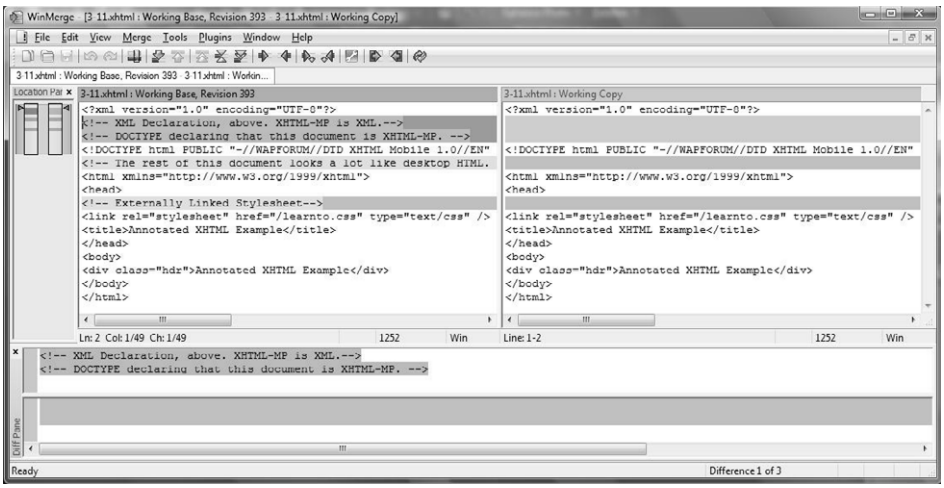
파일 비교

파일 비교 유틸리티는 파일을 한 줄 단위로 검사하고 텍스트 파일에 차이를 병합하는 기능을 제공한다. 이 유틸리티는 프로젝트 기간 동안 계속 변경되는 소스 코드 파일들의 변화를 추적하는 데 편리하다. IDE에 결합해서 사용하는 소스 코드 관리, 파일 비교 도구는 여러분이 웹 개발 프로젝트에서 소스 코드 파일의 변경 사항을 관리하고 추적하는 데 도움을 줄 것이다.

여러분은 몇 가지 윈도우용 오픈소스 파일 비교 유틸리티를 마이크로소프트에서 찾을 수 있다: WinMerge(`http://winmerge.org/`), ExamDiff(`http://www.prestosoft.com/edp_examdiff.asp`), 그리고 WinDiff from Microsoft(`http://support.microsoft.com/kb/159214`).

리눅스와 유닉스 운영체제에서는 내장된 `diff` 명령어를 사용할 수 있다. 애플 역시 Mac OS X를 위해 애플 개발자 도구의 한 부분으로 무료 FileMerge 유틸리티를 제공한다.

뿐만 아니라 많이 사용되고 있는 CVS와 SVN 버전 관리 클라이언트(그림 2–23)는 파일 비교와 병합 유틸리티를 함께 묶어서 제공하는 도구이다. 소스 관리 저장소에 변경된 코드를 제출하기 전에 차이점을 비교해 보고, 각 파일의 수정 내용을 다시 살펴볼 수 있도록 파일 비교 도구를 사용하는 것은 소프트웨어 개발 과정에 꼭 필요한 훌륭한 방법이다.



:: **그림 2-23** 주석이 제거된 XHTML-MP 문서 비교(Tortoise SVN client)

Note 텍스트 파일 비교를 괴짜 속어로 doffing이라고 한다. diff는 초기 유닉스 파일 비교 프로그램의 이름이다.

소스 코드 관리

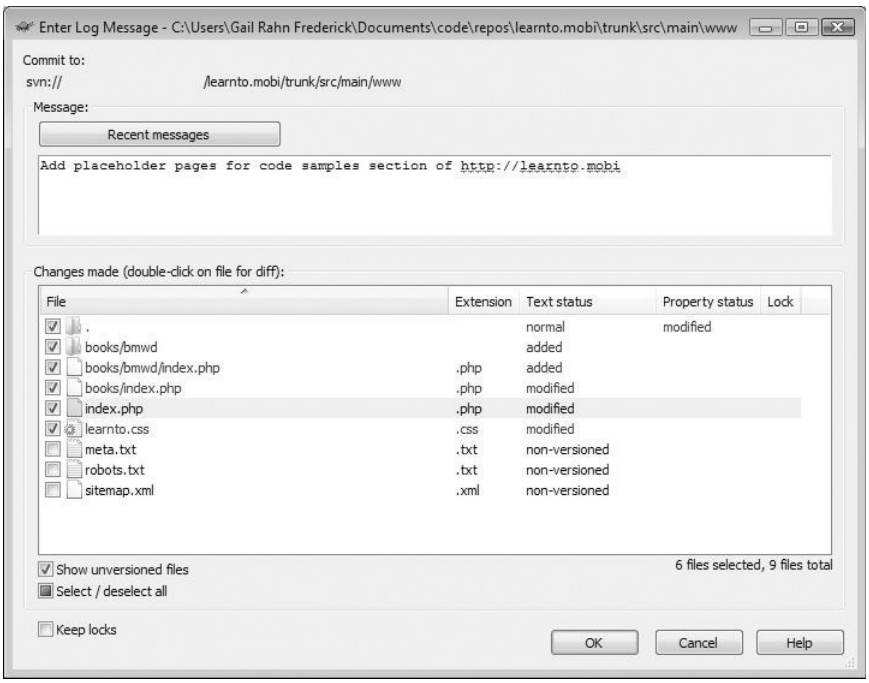
소스 코드 관리는 프로젝트 개발 과정에서 텍스트 파일에 대한 수정 내용을 관리하고 저장하는 것이다. 소스 코드 관리는 팀원들이 같은 파일들을 동시에 작업하고 하나의 복사본으로 변경 내용을 병합해 관리할 수 있도록 한다. 파일에 대한 각각의 수정 내용은 수정 당사자가 소유한다. 변경 내용이 제출되는 저장 장소는 중앙 또는 분산 소스 코드 저장소를 선택할 수 있다. 다수의 사용자가 하나의 파일을 수정하고 제출해서 충돌이 발생하면, 소스 관리 시스템은 자동으로 여러 수정 내용을 단일 변경 파일에 병합하도록 시도한다. 만약 자동 병합이 수행될 수 없다면, 시스템은 파일 비교 기능을 사용해 사용자가 직접 코드 변경 내용들에 대해 수동 병합을 할 수 있도록 가이드한다.

소스 코드 관리는 개발 기간 동안 소프트웨어 프로젝트를 제어하고 안전한 상태로 유지시킨다. 특히 개발 팀이 분산돼 있을 때 이 시스템을 사용하면 효과가 크다. 이 시스템의 클라이언트 서버 구조는 클라이언트가 물리적으로 어디에 있든지 어떤 운영체제를 사용하든지에 상관없이 소스 코드 저장소와 상호작용할 수 있다. 중앙 소스 관리 서버는 소스 코드 파일과 주석화된 수정 내용 히스토리를 저장하고 사용자 권한을 관리한다. 개발자들은 GUI 또는 커맨드라인 클라이언트 애플리케이션을 사용해 로컬 소스 파일들을 최신 버전 저장소로 업데이트하고, 편집을 위해 파일을 따로 관리하고, 변경된 파일을 다시 저장소로 제출할 수 있다. 개발자들은 저장소에 소스 코드를 제출할 때 변경 내용을 설명할 수 있도록 명확하면서 의미를 제대로 전달할 수 있는 주석을 포함해야 한다.

현재 많이 사용하면서 신뢰할 수 있는 두 가지 무료 소스 버전 관리 시스템에는 CVS와 SVN이 있다. 최근에는 git과 mercurial 같은 시스템이 새로운 대안으로 떠오르고 있다.

CVS[Concurrent Version System](http://www.nongnu.org/cvs/)는 두 시스템 중 좀 더 오래된 것이다. 설치가 쉽고 널리 사용되고 있지만, 최근에 들어서는 높은 브랜치 태그 작성 비용과 UNIX 환경에서의 서버 지원 제한, 그리고 고급 파일 관리 시나리오를 다루는 기술에서 문서화된 이슈들이 알려지면서 진부한 도구라는 평을 받고 있다.

Subversion[SVN](http://subversion.tigris.org/)은 CVS를 대체할 수 있는 도구이다. 그리고 많은 CVS의 단점을 보안한다. SVN은 서버와 클라이언트 사이에서 플랫폼 독립적으로 사용된다. 그것은 소스 코드 관리 트랜잭션을 위해 커맨드라인 클라이언트와 Tortoise SVN(http://tortoisesvn.tigris.org/), 그리고 쉘[shell]과 통합된 강력하고 훌륭한 기능의 윈도우 GUI 클라이언트를 제공한다(그림 2-24).



:: **그림 2-24** 추가되고, 수정되고, 그리고 관리되지 않는 파일들을 보여주는 Tortoise SVN 제출 대화 상자

Note 여러 종류의 OS 사이에서 일관성 없이 사용되는 라인 브레이크line break 문자는 플랫폼 독립적인 소프트웨어 개발에 난적이다. 예를 들어, 윈도우 개발자가 소스 코드를 UNIX 스타일 라인 브레이크를 갖도록 변경하고 제출한다면, 그 파일은 다시 윈도우 스타일 라인 브레이크를 포함하도록 업데이트될 수 있다. 이것은 소스 관리 시스템이 소스 코드가 변경될 때 코드와 라인 브레이크가 업데이트됐다는 것을 인식한 후, 그것이 파일 변경 내용들 사이의 차이를 다시 살펴볼 때 원치 않는 복잡함을 유발할 것이라는 것을 판단할 수 있기 때문이다. 만약 여러분의 프로젝트 팀원 중 다중 OS를 사용하는 개발자가 있다면 소스 코드 파일에서 라인 브레이크 형식 표준을 설정할 필요가 있다. IDE와 테스트 편집기는 라인 브레이크 형식을 선택할 수 있는 옵션을 가지고 있으니 그것을 사용하면 된다.

CVS와 SVN은 모두 서버를 설치하고 저장소를 생성하기 위해서 관리자 권한을 가진 시스템 관리자 또는 OS 전문가를 요구한다. CVS 또는 SVN 클라이언트는 오픈소스 프로젝트로서 누구나 자유롭게 어떤 운영체제에서도 쉽게 설치할 수 있다. 또한 플랫폼 독립적으로 동작한다. 이것은 클라이언트 애플리케이션과 서버가 서로 어떤 OS에서 동작하는지에 상관없이 사용할 수 있다는 것을 의미한다.

연습문제 2 파이어폭스에서 모바일 디바이스 위장하기

이 연습문제를 통해 여러분은 이 장에서 배운 확장 모듈을 이용해 파이어폭스로 모바일 디바이스를 위장해서 모바일 웹을 열람할 것이다. 다음 방법을 사용해 모바일 디바이스로 위장해 보자.

- 모바일 MIME 유형 지원
- 모바일 마크업 형식 렌더링
- 모바일 디바이스와 같은 HTTP 요청 헤더 전송

파이어폭스는 모바일 브라우저 또는 모바일 디바이스의 HTTP 요청 특성을 흉내 낸다. 하지만 모바일 브라우징 행동을 복제하는 시도는 하지 않는다.

파이어폭스로 모바일 디바이스를 위장하기 위해 부록 A를 참고해서 user-agent를 사용해 보고, 부록 B를 참고해 HTTP 요청 헤더를 수정해 보자.

- user-agent를 모바일 디바이스로 설정하기 위해 User-Agent Switcher 부가 기능을 사용하라.
- 파이어폭스 요청 헤더를 모바일 디바이스로 맞추기 위해서 Modify Headers 부가 기능을 사용하라.
- 요청 헤더가 부록 B의 실제 모바일 디바이스의 헤더와 일치하는지 확인하기 위해 Live HTTP Headers 부가 기능을 사용하라.

다음은 모바일 웹에서 XHTMP-MP와 WML 문서를 열람해 보자. 서로 다른 다양한 특성을 가진 모바일 디바이스로 위장해 모바일 웹을 열람해 보자(아이폰, 팜 프리, 소니에릭슨 C905, 그리고 LG VX9100 디바이스 등).

앞의 단계를 실행한 후 다음 질문에 답하라.

1. 여러분은 같은 도메인에서 다양한 모바일 웹 경험을 할 수 있었나? 예를 들어, http://cnn.com은 XHTML-MP와 WebKit에 최적화된 XHTML 모바일 웹사이트를 모두 제공한다.
2. 모바일 상황에서 콘텐츠는 데스크톱 웹사이트와 어떤 관련이 있었나? 데스크톱 웹과 모바일 웹 사이의 사이트의 테마는 일관성이 있었나?

3. 모바일 사용자는 모바일 웹사이트를 사용할 때 목적 지향적인 패턴을 보인다. 그래서 한 사이트에서 3분 이상 머무르는 경우가 거의 없다. 여러분은 이 시간 안에 모바일 웹사이트에서 일반적인 작업을 끝낼 수 있었는가? (예를 들어, 날씨 웹사이트의 경우에 여러분이 사는 도시의 날씨 예보를 찾는 게 쉬었는가? 여행 웹사이트의 경우에 비행기 도착 정보를 얼마나 빨리 찾을 수 있었나?)
4. 어떤 user-agents에 대해서 웹사이트는 모바일 최적화된 마크업으로 리다이렉트했는가? 어떤 user-agents가 데스크톱 마크업을 받았는가? 이유는 무엇인가?

이제 이 연습문제를 모바일 에뮬레이터에서 다시 수행해 보자. 파이어폭스와 에뮬레이터 사이에 모바일 웹 페이지를 표시하는 방식에서 어떤 차이가 있었는가?

요약

이 장에서 여러분은 모바일 웹 개발에서 사용하는 도구를 설치하고 설정하는 방법을 배웠다. 마크업 사용 지원과 서버 측 언어를 고려해 IDE를 선택했으며, 모바일 웹에서 흔히 사용하는 MIME 유형에 대해서 배웠다. 추가로 MIME 유형을 지원하도록 웹 서버 설정 방법도 함께 배웠다. 또한 모바일 웹 페이지가 어떻게 표현되는지 확인하고 디버그하는 세 가지 방법을 배웠다. 파이어폭스에서 부가 기능을 사용하는 방법, 모바일 브라우저 에뮬레이터를 사용하는 방법, 그리고 실제 모바일 디바이스를 사용하는 방법이 있었다. 모바일 웹 소스 코드 관리를 위해 사용하는 파일 비교 및 소스 버전 관리 유틸리티도 배웠다. 연습문제에서는 파이어폭스와 모바일 에뮬레이터를 사용해서 모바일 디바이스로 위장하고 모바일 웹을 열람하는 방법을 다시 되짚어 봤다.

다음 장에서는 모바일 마크업 언어와 스타일시트의 구문을 실제 산업 기관과 표준 단체가 인정한 모범 사례를 토대로 살펴볼 것이다.