

# 모바일 웹 앱을 위한 HTML5 및 프레임워크 동향

HTML5 and Mobile Framework Trends for Mobile Web Applications

채원석 (W.S. Chae) 콘텐츠서비스연구팀 선임연구원  
 박찬우 (C.W. Park) 콘텐츠서비스연구팀 기술원  
 최완 (W. Choi) 콘텐츠서비스연구팀 기술원  
 안세영 (S.Y. Ahn) 콘텐츠서비스연구팀 기술원  
 노병석 (B.S. Roh) 콘텐츠서비스연구팀 연구원  
 이준우 (J.W. Lee) 콘텐츠서비스연구팀 팀장

애플과 구글이 주도한 모바일 앱 장터에 새로운 변화가 찾아올 것으로 예상된다. HTML5(HyperText Mark-up Language 5)로 무장한 모바일 웹 앱의 등장은 모바일 인터넷 환경의 비약적인 발전과 고사양의 스마트기기 보급 확대에 힘입어 모바일 앱의 생태계를 재구성하고 있다. 본고에서는 애플리케이션 플랫폼의 발달 과정과 모바일 웹 환경에서의 HTML5의 역할에 대해서 알아보고 모바일 웹 프레임워크에 대해 정리하였다.

2012  
 Electronics and  
 Telecommunications  
 Trends

사용자 중심  
 차세대콘텐츠기술 특집

- I. 머리말
- II. 모바일 웹 애플리케이션
- III. 모바일 웹에서의 HTML5
- IV. 모바일 웹 프레임워크
- V. 맺음말

## I. 머리말

최근 수년째 세계에서 가장 혁신적이고 경쟁력 있는 기업으로 평가받고 있는 애플의 성공은 아이폰이나 아이패드와 같은 하드웨어 그 자체보다는 엄청난 규모의 애플리케이션 개수와 이들을 간편하게 유통시킬 수 있는 온라인 장터에 그 요인이 있다. 애플의 앱스토어뿐만 아니라 이어서 등장한 구글의 안드로이드마켓과 그 외 다양한 오픈 마켓에 의해 형성된 모바일 앱 생태계에도 이제 변화가 찾아오고 있다.

파이낸셜타임스는 다양한 플랫폼의 스마트폰 및 태블릿에서 이용할 수 있는 서비스를 HTML5(Hyper-Text Mark-up Language 5)가 적용된 웹 앱으로 출시하여 기존 앱스토어에 도전장을 내밀었다. 아이패드 웹 앱이 곧 앱이고 웹 사이트인 셈이다. 웹 앱은 매번 업데이트할 필요 없이 자동으로 업데이트된 콘텐츠를 제공받을 수 있다. 기존 앱과 구별하기 힘든 웹 앱의 등장은 새로운 시대를 예고하고 있다.

본고에서는 우선 애플리케이션 플랫폼의 발달 과정과 모바일 환경의 애플리케이션 유형 및 특징을 소개한 후, HTML5라는 웹 표준 기술이 어떠한 것이고 그것으로 인해 모바일 앱 시장에 미친 영향과 모바일 웹 환경에서 HTML5의 역할에 대해서 살펴본다. 마지막으로 모바일 웹 프레임워크란 무엇이며 어떤 역할을 하는지와 대표적인 세 가지 모바일 웹 프레임워크에 대해 어떠한 특징들이 있는지 간략히 알아본다.

## II. 모바일 웹 애플리케이션

불과 십 년 전만 하더라도 소프트웨어를 개발한다고 하면 우선 어떤 OS 기반의 플랫폼이 대상인지가 중요했다. 플랫폼이 서로 다르기 때문에 윈도우용 프

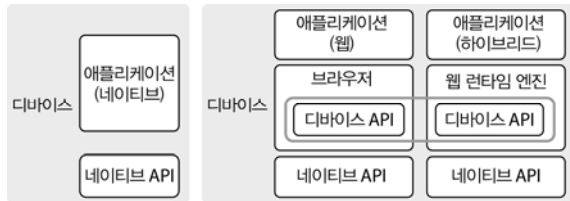
로그램은 리눅스에서 실행되지 않고 그 반대도 마찬가지이다. 여러 플랫폼에서 같은 프로그램을 실행시키기 위해서는 플랫폼별로 버전이 따로 제공되어야만 했다.

2000년대 중반 웹 2.0 시대에 들어서면서 많은 부분에 변화가 찾아왔다. 팀 버너스 리(Tim Berners-Lee)에 의해 시작된 월드와이드웹(WWW)은 정적인 HTML 문서로 구성되어 하이퍼링크를 통해 단순히 자신이 읽을 문서 페이지로 이동하는 정도의 상호작용만 있는 한계점이 있었다. 그러나 웹 2.0은 단순한 웹사이트의 집합체가 아닌 웹 애플리케이션을 제공하는 하나의 완전한 플랫폼으로의 발전을 이끌었다.

웹을 플랫폼화한 가장 좋은 예는 ‘구글 문서도구’에서 찾을 수 있다. 워드프로세서에서부터 스프레드시트까지 다양한 오피스 애플리케이션 작업을 OS 기반의 프로그램이 아닌 웹 기반의 환경에서 처리할 수가 있다. 인터넷만 접속되는 환경이라면 어떤 OS나 브라우저를 선택하더라도 프로그램을 설치할 필요도 없이 웹 서비스로 이용이 가능하게 된 것이다.

이러한 현상은 모바일 인터넷 환경의 변화와 고사양의 스마트기기 확대와 더불어 모바일 플랫폼의 진화 과정에서도 더 빠르게 나타나고 있다. 모바일 환경이 급격하게 대두된 것은 아이폰의 혁신과 뒤이은 구글의 안드로이드 스마트폰이 시장을 재편하면서부터이다. 아이폰이나 안드로이드용 앱을 개발하면서 플랫폼별로 앱을 다르게 개발해야 하기 때문에 ‘플랫폼 파편화’로 인한 개발자들의 고충은 날로 커지고 있다. HTML5 기술을 활용한 모바일 웹 플랫폼은 모바일 플랫폼 파편화를 해결하는 대안으로 떠오르고 있으며, 아이폰 OS(iOS)와 안드로이드가 동일한 ‘웹킷(Webkit)’ 웹 브라우저 엔진을 사용하기 때문에 이를 지원하는 다양한 솔루션이 발표되고 있다[1].

모바일 애플리케이션을 유형별로 분류하면 모바일



(그림 1) 모바일 애플리케이션 종류에 따른 개념도[2]

네이티브 애플리케이션, 모바일 웹 애플리케이션, 모바일 하이브리드 애플리케이션의 세 가지로 나눌 수 있다[3]. 애플리케이션을 줄임말로 ‘앱’이라는 용어를 사용하기도 한다. 네이티브 앱은 각각의 스마트폰 환경에서 별도로 제공하는 프로그래밍 언어와 SDK (Software Development Kit)를 이용해 제작한다. 즉, 자바(Java), 오브젝티브-C(Objective-C), C# 등의 프로그래밍 언어를 별도로 배워서 해당 OS 환경에서 개발해야 한다. 반면에 웹 앱은 HTML, CSS(Cascading Style Sheets), 자바스크립트(JavaScript) 등 웹 표준 기술을 이용하여 개발된다. 하이브리드 앱은 웹 표준 기술을 사용하되 서로 다른 플랫폼에서 실행될 수 있도록 크로스 프레임워크를 이용하여 네이티브 앱으로 변환시켜 배포되는 형식이다.

네이티브 앱은 (그림 1)에서와 같이 디바이스의 네이티브 API(Application Programming Interface)를 사용해 개발된다. 이것은 빠른 속도와 디바이스의 다양한 기능을 효과적으로 활용할 수 있게 해주지만 플랫폼이 달라지면 개발 모듈의 재활용이나 애플리케이션의 업데이트 측면에서 불편함을 초래한다.

디바이스 API는 서로 다른 플랫폼의 네이티브 API를 이용할 수 있도록 표준화한 것으로 보통 디바이스에 설치된 브라우저에 내장되게 된다. 웹 앱은 이를 통하여 디바이스의 다양한 기능을 활용할 수 있게 되고, 웹 개발자들이 네이티브 API를 알지 못하더라도 웹 표준 기술을 활용하여 프로그램이 가능하다. 또한 오픈 API 등을 활용해 인터넷을 통하여 다양한 서비스와

〈표 1〉 웹 앱과 네이티브 앱의 비교[4]

기능	웹 앱	네이티브 앱
개발 환경	기존에 사용하던 웹 개발 환경, 모든 운영체제 가능	Mac 장비(iOS) 또는 멀티 플랫폼(Android)
사용 언어	HTML, CSS, 자바스크립트	Objective-C, Java 등
앱 개발자 비용	무료	연 99달러(iOS) 또는 무료(Android)
배포	브라우저가 설치된 환경	앱스토어/안드로이드 마켓
결제 시스템	독자 결제 구축 또는 광고 수익	앱스토어 판매 수익
하드웨어 지원	제한적	모든 기능 활용
업데이트	즉시 반영	검수 필수
UI	제한적	풍부한 UI 가능

〈자료〉: 김응석, 쉽게 배우는 웹앱&하이브리드앱, 이지스퍼블리싱, 2012, 3, p. 31.

와의 융합이 가능하다. 〈표 1〉에서와 같이 웹 앱은 하드웨어 지원이나 UI 측면에서 네이티브 앱에 비해 제한된 기능만 활용할 수 있지만 다양한 플랫폼에서 크로스 브라우징이 가능한 것을 최대의 장점으로 꼽을 수 있다.

한편 하이브리드 앱은 디바이스 API를 내장하고 있는 웹 런타임 엔진을 이용해 네이티브 UI 형태로 웹 앱을 사용할 수 있도록 하는 방식이다. 웹 앱은 구현 기술 자체가 일반 웹 기술이라서 일반 사용자가 웹 앱에 접근하려면 브라우저를 통해 URL로만 접근해야 한다. 하이브리드 앱은 웹 앱과 네이티브 앱의 특징을 조합한 형태로 웹 표준 기술을 사용하여 개발을 하지만 애플의 앱스토어나 안드로이드 마켓과 같은 네이티브 앱 장터에 등록하여 배포가 가능하다. 따라서 웹 앱을 네이티브 앱으로 감싸주는 장치가 필요한데 어도비(Adobe)사의 폰갭(PhoneGap)이나 앱셀러레이터(Appcelerator)사의 티타늄(Titanium)과 같은 제품이 이와 같은 역할을 하게 된다.

웹 기반의 애플리케이션은 하나의 소스코드로 여러

플랫폼을 지원할 수 있다. 모바일 웹 앱은 일반 웹 사이트와 달리 모바일 친화적인 인터페이스와 향상된 스타일로 네이티브의 사용성을 따라 잡고 있다. 또한 웹만의 장점인 즉각적인 업데이트가 가능하며, 최신 웹 기술인 HTML5를 접목하면 오프라인 지원, 로컬 저장소 활용, 2D 그래픽의 처리, 미디어 재생, 소켓 통신, 멀티 스레드 프로그래밍 등의 구현이 가능해져 기존 웹의 한계를 훌쩍 뛰어 넘을 수 있게 되었다.

### III. 모바일 웹에서의 HTML5

‘웹의 아버지’라 불리는 팀 버너스 리에 의해 HTML의 시안이 발표되고, 웹 표준과 가이드라인을 마련하기 위한 국제 표준화 기구인 W3C(World Wide Web Consortium)가 결성된 이후 HTML4.01이 발표되기까지 HTML 표준은 거듭 발전해 왔다. 그러나 XML(eXtensible Mark-up Language) 형식으로 재정의한 XHTML(eXtensible HTML)로의 표준화 추진 방향은 기존의 HTML과 호환성이 부족하여 확산에 실패하게 된다[5]. 때마침 웹 2.0의 열풍이 불면서 ‘플랫폼으로서의 웹’은 웹 그 자체를 소프트웨어로 보는 웹 애플리케이션 시대를 열었다.

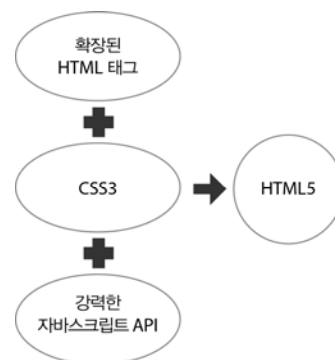
웹 기반의 애플리케이션이 데스크톱 애플리케이션과 대등한 사용자 경험을 주는 것을 목표로 하는 ‘리치 인터넷 애플리케이션(RIA: Rich Internet Application)’ 기술이 이러한 시기에 유행하기 시작했다. RIA는 웹 브라우저에서 풍부한 인터페이스를 지원하고 데이터를 인터넷으로 제공받는 웹 애플리케이션 형태를 말한다. 현재 RIA를 구현하는 기술로는 어도비사의 플렉스(FLEX), 마이크로소프트사의 실버라이트(Silverlight), W3C의 HTML5, 이 세 가지로 크게 나눌 수 있다[6]. 여기서 HTML5를 제외한 나머지 기술은 웹 브라우저가 아닌 별도의 실행 환경을 요구하

는 플러그인 기술이라는 한계가 있다.

HTML5는 웹 브라우저만으로 실행 가능한 환경과 전 세계 표준적인 기술 플랫폼이라는 점을 강조하면서 강력한 멀티미디어 기능을 탑재한 HTML의 차기 버전으로 제안되었다. 2014년 2분기가 되어야 HTML5 표준이 확정될 예정이지만, 이미 최신 브라우저에서는 이를 적극적으로 지원하고 있으며 최신 모바일 기기에 탑재된 브라우저는 더욱 지원을 강화해 가고 있다. 애플과 구글뿐만 아니라 마이크로소프트, 모질라, 오페라 등 대부분의 웹 브라우저 벤더들이 개발에 참여하여 점진적으로 HTML5의 지원을 늘려가고 있다.

모바일 웹 앱을 HTML5 기반으로 개발하는 이유는 다음과 같이 세 가지를 꼽을 수 있다[7]. 첫째, 스마트 열풍과 더불어 출시되는 스마트폰을 포함한 대부분의 스마트기기에는 HTML5를 지원하는 브라우저가 탑재되어 있어서 바로 이용이 가능하다. 둘째, 웹 앱은 네이티브 앱과 달리 네이티브 개발환경을 설치할 필요가 없으며, 네이티브 앱 장터를 통하여 심사를 받을 필요도 없이 배포가 가능하다. 셋째, 표준 규격인 HTML5는 아이폰 환경이든 안드로이드 환경이든 플랫폼과 상관없이 크로스 브라우징이 가능하다.

HTML5가 단순히 기존의 HTML 표준에서 태그를 확장한 것이라고 생각할 수도 있지만, (그림 2)에서



(그림 2) HTML5의 구성 요소[8]

보는 바와 같이 CSS와 자바스크립트를 포함하고 있는 개념으로 이해하는 것이 맞다. HTML이 문장의 논리 구조를 지정한다면, CSS는 표시 방법을 지정한다고 할 수 있다. 이전 버전의 HTML에 논리 구조에 하면 표시와 관련된 태그까지 추가하게 되자 복잡하고 부자연스럽게 되어 버린 문제를 해결하기 위해 HTML5에서는 화면 표시와 관련된 부분은 CSS가 전담하도록 권고하고 있다.

CSS3는 HTML5에서 권장되는 스타일 표준으로 웹 기술의 한 요소이다. CSS2와 CSS3의 가장 큰 차이점은 후자가 모듈 기반이라는 점이다. 즉, 모바일 웹 브라우저가 필요에 따라 원하는 CSS 모듈만 탑재하거나 수시로 업데이트하는 것이 가능해졌다. 게다가 화려하고 역동적인 면모를 갖추기 위해 상자의 크기에 따른 말줄임 표시, 투명한 배경, 그림자 효과, 둥근 모서리, 애니메이션 효과 등의 속성을 추가하여 자바스크립트나 서버 기술에 의존하지 않아도 되어서 웹 사이트의 성능 향상에까지 기여하게 되었다[5].

모바일 환경에서 HTML5가 특별히 다른 태그를 활용하는 것은 아니지만, 몇 가지 특징이 모바일 환경에서 활용하면 더욱 효과적이다. 첫째, 문서의 구조를 강화하여 ‘시멘틱 웹(semantic web)’을 추구함으로써 접근성을 높이고 검색 엔진의 정확도가 높아져 음성 지원 브라우저나 텍스트 전용 브라우저 등 특수한 브라우저에서도 정보 전달이 용이하다. 둘째, 새로 추가된 <canvas>, <audio>, <video> 태그는 별도의 플러그인을 설치할 필요 없이 그래픽 및 멀티미디어 재생 기능을 강화할 수 있다. 마지막으로, 로컬 저장소나 앱 캐시, 지오로케이션(geolocation) 등의 자바스크립트로 이용할 수 있는 API의 추가는 네트워크 연결이 끊기더라도 오프라인 모드로 동작이 가능하게 하고 다양한 위치 기반 서비스가 실현될 수 있다.

<표 2>에 나타낸 것처럼 HTML5에는 보다 강력한

<표 2> HTML5 자바스크립트의 여러 가지 기능들[4]

분류	항목	설명
오프라인/저장 기능	Local Storage API	클라이언트 쪽에서 상태값 등의 정보를 저장하고 읽기
	Web SQL DB API	DB 생성 및 테이블을 이용하여 SQL을 사용
	Indexed DB API	key와 value를 단위로 인덱싱된 DB를 사용하여 빠른 검색 가능
	Offline Web APPs API	네트워크 접속이 끊기더라도 모바일 웹을 이용할 수 있는 캐시 지원
실시간 통신	Web Socket	네트워크를 통한 소켓 통신이 용이
	Web Worker	백그라운드 모드로 실행되는 복잡한 처리가 동시에 가능
하드웨어 사용	Geolocation API	지도와 GPS의 위치 정보를 연동하여 처리 가능
	Drag and Drop API	끌어다 놓기의 UI 처리가 용이
	File API	로컬 컴퓨터에 파일을 저장하고 읽기
	Device Orientation	자이로스코프 센서를 인식하여 방향을 제어

<자료>: 김용석, 쉽게 배우는 웹애플리케이션, 이지스퍼블리싱, 2012, 3, p. 371.

자바스크립트 API가 추가되어 독립적인 애플리케이션 플랫폼으로서의 면모가 갖추어지고 있다. 지금도 표준화가 진행 중이어서 새로운 API의 추가도 예상되지만 현재까지 알려진 중요한 API에 대해서 간단히 소개한다.

## 1. 애플리케이션 캐시

오프라인 상태에서도 웹 프로그램을 작동시킬 수 있는 기능으로 ‘매니페스트(manifest)’ 파일에 등록하면 인터넷에 접속하였을 때와 동일하게 동작한다.

## 2. 크로스 도큐먼트 메시징

여러 개의 창 또는 프레임 간에 메시지를 주고 받을

수 있게 해주는 기술로 특정 페이지에 했던 조작을 나머지에 대해서도 동일하게 적용시킬 수 있다.

### 3. 웹 스토리지

사용자가 입력한 정보나 기타 데이터를 로컬 영역에 저장하여 오프라인으로 동작하는 상태에서도 이 정보를 사용하여 웹 프로그램이 동작할 수 있다.

### 4. 웹 소켓

웹 서버와 클라이언트가 양방향 통신을 가능하게 해주며, 연결 상태를 유지할 수 있어 실시간으로 메시지를 주고받거나 서버 푸시(push) 처리가 가능하다.

### 5. 웹 워커

자바스크립트를 브라우저에 부담을 주지 않고 백그라운드에서 실행할 수 있게 해주어 사용자의 조작을 방해하지 않고 작업의 효율을 높일 수 있다.

### 6. 지오로케이션

사용자의 지리적 정보에 대해 현재의 위치뿐만 아니라 이동하는 위치를 지속적으로 추적할 수 있어 다양한 위치 기반 애플리케이션을 구현할 수 있다.

## IV. 모바일 웹 프레임워크

모바일 웹 앱은 HTML5, CSS3, 자바스크립트와 같은 웹 표준 기술을 활용하여 개발된다고는 하지만 사용자 인터페이스가 모바일 친화적이어야만 비로소 일반적인 웹 기반의 애플리케이션과 차별화될 수 있다. 즉, 터치 상호작용에 있어서 네이티브 앱과 동일한 경

험을 줄 수 있어야 하며 단순히 전통적인 새로 고침 형태의 페이지 이동 방식이 아닌 전환 애니메이션 효과를 느낄 수 있는 자연스러운 화면 구성이 필수적이다.

모바일 환경에 최적화된 UI와 터치 처리, 애니메이션을 구현하려면 모바일 웹 앱을 개발하는 과정에서 최신 웹 표준을 준수하면서도 정교한 자바스크립트를 작성하여 각종 동작과 이벤트에 반응해야 한다. 특히, 네트워크 연결이 차단되거나 불안정한 상황에서도 정상적으로 작동하려면 HTML5의 로컬 스토리지와 애플리케이션 캐시 기능을 활용하기 위해 세심한 노력이 필요하게 된다. 즉, 페이지 이동에 따른 내비게이션(navigation) 관리와 비동기 서버 통신을 직접 처리해 주어야 한다. 그 외에 나머지 세세한 부분까지 모바일 최적화를 위해 섬세한 구현이 필요하기 때문에 개발자가 이 모든 것을 직접 구현하려면 상당한 노력이 필요하게 된다. 무엇보다도 개발 시간과 안정성이 문제가 된다[6],[9].

모바일 웹 프레임워크는 이러한 모바일 환경에 최적화된 구현의 상당 부분을 활용할 수 있게 해준다. 프레임워크에 정교하게 녹아있는 HTML5와 CSS3의 최신 웹 표준 기술 요소와 강력하게 진화된 자바스크립트 기능을 활용하면 UI/UX(User Interface/User eXperience)와 화면 구성을 네이티브 앱 수준으로 끌어올릴 수 있다. 따라서 모바일 웹 프레임워크는 모바일 UI 프레임워크라고도 불리며, 아이폰 개발 환경의 인터페이스 빌더(interface builder)나 안드로이드 개발 환경의 레이아웃 에디터(layout editor)와 같은 역할을 한다고 볼 수 있다.

근래 관심이 집중되고 있는 모바일 웹 프레임워크에 대한 차이를 비교하기 쉽게 <표 3>에 나타내었으며, 각각의 프레임워크에 대해 좀 더 자세히 소개하고자 한다.

〈표 3〉 주요 모바일 웹 개발 프레임워크 비교[9]

	jQTouch(version1, beta4)	jQuery Mobile(1.1.0)	Sencha Touch(2.0)
개발 난이도	하	중	상
표현 자유도(기능의 다양함)	하	중	상
개발 방식	마크업 기반	마크업 기반	스크립트 기반
테마 특징	일반 CSS	테마롤러 도구 지원	Sass 기반
기반 스크립트	jQuery 기반	jQuery 기반	ExtJS 기반
라이브러리 용량(최소화 버전 파일 기준)	소	중	대
최적화 대상	스마트폰	스마트폰 및 태블릿	스마트폰 및 태블릿
지원 기기	한정적 아이폰 및 안드로이드 계열 의 모바일 기기	다양함 아이폰, 안드로이드, 블랙베리, 팜 웹 OS, 오페라 모바일, 파이어폭스 모바일 등	중간 아이폰, 안드로이드 계열의 모바일 기기, 블랙베리 OS 6.0 이상

〈자료〉: 박종명, 시작하세요! 모바일 웹 개발, 위키북스, 2011. 6, p. 30.

## 1. jQTouch

jQTouch는 클라이언트 자바스크립트 라이브러리 중 가장 많이 사용되는 jQuery 기반의 플러그인 형태로 개발되었으며 〈표 3〉에서 비교하고 있는 세 가지 중에 가장 먼저 알려졌다. 단순하고 활용하기 쉽다는 특징이 있지만 상대적으로 기능이 적어서 프레임워크라기보다는 라이브러리로 보는 것이 맞다. 그러나 마크업 기반의 개발 스타일을 채택하였기 때문에 웹 개발자들이 쉽게 적응할 수 있으며 최신 CSS3 기능을 활용해 향상된 스타일과 애니메이션을 구현할 수 있다.

지금은 안드로이드 플랫폼까지 지원이 가능하지만 원래는 아이폰용 UI 라이브러리로 개발이 시작되었으며, 태블릿 환경보다는 스마트폰과 같이 비교적 작은 화면에 적합하도록 개발 방향이 맞추어져 있다. MIT 라이선스 정책을 따르기 때문에 개발과 배포가 자유롭지만, jQTouch가 센차 터치(Sencha Touch)에 인수되어 개발이 중단된 상태이며 jQTouch 개발자는 현재 센차 터치 개발에 참여하고 있다.

## 2. jQuery Mobile

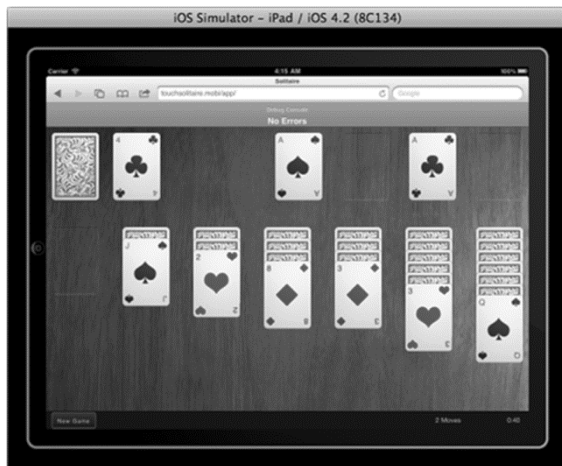
jQuery Mobile은 jQuery 진영에서 직접 개발한 모

바일 웹 프레임워크로 본 장에서 소개하고 있는 세 가지 중에 가장 최근에 개발되기 시작했다. 네이티브 수준의 모바일 웹 앱을 개발하기 위한 각종 UI 컨트롤과 이벤트 처리, 애니메이션 효과 및 자동 내비게이션, Ajax(Asynchronous JavaScript and XML) 통신 등 모바일에 최적화된 기능을 제공한다[9]. 다른 프레임워크와 비교해 가장 많은 기기를 지원하며 마크업 기반의 개발 스타일을 채택하여 HTML5와 CSS3 기능을 적극 활용하고 있다.

웹 브라우저의 종류가 다양해지고 버전별로도 지원하는 기능이 다르기 때문에 모든 브라우저에서 동일한 경험을 주는 것은 불가능하다. 이 프레임워크는 기본 태그만으로도 최소한의 동작을 보장하며 브라우저 각각의 기능 지원 여부에 따라 향상된 기술을 점진적으로 적용함으로써 구형 브라우저든 신형 브라우저든 서비스를 이용하는 데에는 큰 지장을 주지 않도록 하는 원칙을 준수하고 있다. 이러한 이유로 다양한 기기와 브라우저를 지원할 수 있게 되었다.

## 3. Sencha Touch

센차 터치는 jQuery와 경쟁관계에 있는 자바스크립



〈자료〉: w3labs.kr

(그림 3) 센차 터치로 개발한 카드 게임

트 라이브러리인 ExtJS 진영에서 개발한 프레임워크이며 상업적으로 판매될 정도로 완성도가 높고 현존하는 프레임워크 중에 가장 뛰어난 성능과 방대한 규모가 장점이다[10]. jQTouch는 화면이 작은 기기에 적합한 반면 이 프레임워크는 아이패드와 같은 태블릿 기기도 주요 지원 대상으로 삼기 때문에 센차 터치에는 다양한 스마트기기를 구분할 수 있는 섬세한 장치 인식 API를 제공하고 있다.

다른 두 프레임워크가 마크업에 기반을 두고 있는 것과는 달리 스크립트 기반의 개발 스타일을 채택하여 객체와 메소드, 상속과 확장 등 객체지향 기법과 고급 스크립트 개념이 API에 녹아 있는 것이 특징이다. 따라서 다른 프레임워크에 비해 프로그램적인 사고방식을 더 많이 요구한다. (그림 3)은 센차 터치로 개발된 카드 게임 예제 스크린샷이다. 넓은 화면을 위한 복잡한 화면과 자유도가 높은 애플리케이션의 구성을 원하면서도 최적화된 터치 이벤트 반응을 원한다면 센차 터치가 좋은 선택이 될 것이다.

## V. 맺음말

모바일 인터넷 환경의 비약적인 발전과 고성능 스

마트기기의 보급 확대로 모바일 시장에서 웹 앱은 기존의 네이티브 앱을 넘어서 모바일 디바이스 고유의 영역까지 차지해 가고 있다. 이제 웹 앱은 HTML5, CSS3, 자바스크립트라는 세 가지 요소가 결합되어 하나의 웹 브라우저 기반 프레임워크로 자리매김하였다. 모바일 환경에 최적화되어 네이티브 수준의 UX를 가능하게 해주는 모바일 웹 프레임워크의 등장은 짧은 시간에 세련된 결과물을 얻게 해준다.

모바일 앱 시장에 뿌려진 HTML5는 가뭄에 단비처럼 개발과 동시에 모든 플랫폼에 배포가 가능하다는 점에서 대단히 매력적이다. 하지만 보안에 취약하며 브라우저에 종속적이어서 서로 다른 플랫폼에서 동일한 성능을 낼 수 있는지 검증하기 어렵고 오프라인 모드 동작 후 서버와 동기화 문제 등이 지적되고 있다. 이러한 문제점들은 HTML5에 국한된 것이라기 보다는 웹 기술의 기본적인 문제일 수 있고 아직 표준화가 진행 중이기 때문에 다양한 혁신을 통하여 서서히 해결될 것으로 예상된다.

모바일 웹 기술이 네이티브 앱과 같은 수준을 쫓아가고는 있지만 반대쪽도 지속적인 발전과 혁신을 거듭하고 있어서 HTML5가 네이티브 진영을 넘어서리라 단정할 수는 없다. 또한, 현재까지 웹 앱 스토어의 성공적인 수익 모델을 찾기 어려워서 네이티브 앱 장터와 같은 성공을 보장하기는 어렵다. 모바일 웹 앱과 네이티브 앱은 당분간 양측에서 상호 보완적인 관계로 동반 성장해 나갈 것으로 전망된다.

### 용어해설

**자바스크립트** HTML 문서 안에서 사용할 수 있는 스크립트 언어로 사용자에게 인터랙티브한 경험을 부여하기 위해 사용

**시멘틱 웹** 웹에 존재하는 수많은 페이지들에 메타데이터를 부여하여, 웹의 세계를 '의미'와 '관련성'을 가지는 거대한 데이터 베이스로 구축하고자 하는 발상

## 약어 정리

Ajax Asynchronous JavaScript and XML



API	Application Programming Interface
APPs	APPLications
CSS	Cascading Style Sheets
DB	DataBase
HTML5	HyperText Mark-up Language 5
OS	Operating System
RIA	Rich Internet Application
SDK	Software Development Kit
SQL	Structured Query Language
UI/UX	User Interface/User eXperience
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WWW	World Wide Web
XHTML	eXtensible HTML
XML	eXtensible Mark-up Language

## 참고문헌

- [1] 박민우, “모바일 크로스 플랫폼 기술동향에 대한 이해와 향후 전망,” KT경제경영연구소, DIGIECO보고서 Issue&Trend, 2011. 10. 18, p. 4.
- [2] 이강찬, “모바일 웹 플랫폼과 Device API 표준,” 한국정보통신기술협회, *TTA J.*, no. 128, 2010. 3, pp. 44-45.
- [3] 남기효 외, “모바일 애플리케이션 동향 및 전망,” 주간기술동향, 정보통신산업진흥원, vol. 1480, 2011, pp. 13-20.
- [4] 김응석, 쉽게 배우는 웹앱&하이브리드앱, 이지스퍼블리싱, 2012, p. 31, p. 371.
- [5] 윤석찬 외, “실전 HTML5 가이드,” 한국 웹 표준 커뮤니티, 2010, p. 10, p. 64.
- [6] 신용권, 모바일 웹앱, 스마트미디어, 2012, p. 50.
- [7] 쿠지라 히코즈쿠에, 스마트폰 앱 개발을 위한 HTML5, 정보문화사, 2011, p. 19.
- [8] 이두진, HTML5와 폰갭으로 웹앱 나도 만든다, PC BOOK, 2012, p. 23.
- [9] 박종명, 시작하세요! 모바일 웹 개발, 위키북스, 2011, p. 30. pp. 156-162.
- [10] 이병옥, 최성민, 센자터치 모바일 프로그래밍, 에이콘, 2011, pp. 23-33.
- [1] 박민우, “모바일 크로스 플랫폼 기술동향에 대한 이해와