

27. HTML5 정리 및 하이브리드 앱

웹프로그래밍

2020년 2학기

충남대학교 컴퓨터공학과

“본 서비스는 교수/학생이 원격수업 목적으로 이용하고 있는 서비스입니다.”

HTML5 정리

HTML5의 구성요소

- HTML5로 통칭되는 요소는 HTML5 뿐 아니라 CSS3, 웹 애플리케이션을 위한 JavaScript API 확장을 포함한 것

통칭되는 "HTML5"

HTML5

(Hypertext Markup Language 5.0)
콘텐츠 내용과 형식을 표현

- 문서구조의 상세화
- 멀티미디어
- 폼과 이벤트 등

CSS3

(Cascading Style Sheet 3.0)
콘텐츠 표현 방법을 정의

- 표현 기능 모듈화
- 웹 폰트

JavaScript

각종 API를 통해 기능을 표현

- Web Storage
- Web Worker
- Web Socket
- Geolocation API ...

HTML5의 특징

HTML5 기술의 주요 특징	시사점
Semantics: 보다 구조화되고 다양한 기능의 HTML 태그를 제공	보다 지능화되고 다양한 형태의 풍부한 웹 문서 표현 가능
Multimedia: 비디오, 오디오 지원 기능의 자체 지원을 통한 강력한 멀티미디어 기능 제공	액티브X와 플래시 같은 별도 외부 플러그 필요성 제거
Offline & Storage: 네트워크가 지원되지 않는 환경에서도 웹 이용을 가능케 하는 오프라인 처리 기능과 로컬 스토리지, File 액세스 처리 가능	웹의 한계로 여겨졌던 네트워크 단절 시 처리 방법과 데이터 저장 기능 문제 해결
3D, Graphics & Effects: SVG, 캔버스, WebGL 등을 통한 다양한 2차원/3차원 그래픽 기능의 제공	외부 플러그인 없이 다양한 2D/3D 그래픽 처리 가능
Device Access: GPS, 카메라, 동작센서 등 디바이스의 하드웨어 기능을 웹에서 직접 제어할 수 있도록 하는 기능	웹 기반 디바이스 제어 기능을 통해 본격적인 웹 애플리케이션 개발 가능
Performance & Integration: 비동기 통신, 다중 스레드 기능 등을 통한 웹에서의 처리 성능 향상	웹의 가장 큰 문제 중 하나였던 성능 문제를 대폭 개선
Connectivity: 클라이언트와 서버간의 효율적인 통신 기능 제공을 통한 웹 기반 커뮤니케이션 효율 대폭 강화	웹에서 다양한 통신(메시징, 응용간 통신 등) 제공을 통한 응용 개발 범위 확대
CSS Styling Effect: 기존 웹 문서의 변경과 성능저하 없이 웹 애플리케이션의 UI(스타일과 효과 등) 기능을 대폭 강화	UI 측면에서 N-스크린 서비스 제공 가능

HTML5를 바라보는 5가지 관점

- ❖ 차세대 웹 기술의 총합의 관점으로 HTML5
- ❖ 단말, 플랫폼, 스크린의 경계가 없도록 하는 HTML5
- ❖ 새로운 앱과 SW 환경으로서의 HTML5
- ❖ 모든 정보, 서비스와 사물을 묶는 관점으로 HTML5
- ❖ 인프라와 플랫폼으로서의 HTML5

하이브리드 앱

모바일 앱 개발 방식

- ▣ 네이티브 앱
- ▣ 웹 앱
- ▣ 하이브리드 앱

네이티브 앱

- ❖ 모바일 기기에 직접 다운로드하여 로컬(기기)에 저장되는 바이너리 실행 파일
- ❖ 모바일 OS에서 제공하는 모든 API에 액세스할 수 있으며, 대부분 해당 모바일 OS의 고유 기능을 활용

	애플 iOS	안드로이드	블랙베리 OS	윈도우 폰
언어	Objective-C, C, C++	Java(일부 C, C++)	Java	C#, VB.NET 등
툴	Xcode	Android SDK	BB Java Eclipse 플러그인	Visual Studio, 윈도우 폰 개발 툴
패키징 형식	.app	.apk	.cod	.xap
앱스토어	Apple App Store	Google Play	Blackberry App World	Windows Phone Marketplace

웹 앱

- 모바일 디바이스의 브라우저에서 실행되고 모바일 브라우저에서 지원하는 JavaScript, CSS 및 HTML5 기능을 활용

특성	Pure 모바일 웹 앱	Pure 모바일 웹 사이트
도구 및 지식	HTML, CSS 및 JavaScript로 작성	HTML, CSS 및 JavaScript로 작성
실행	바로가기 설치 가능. 네이티브 앱처럼 실행	URL(Uniform Resource Locator)을 통해 웹 사이트를 탐색하여 액세스
사용자 환경	터치식의 상호 작용 UI	고정 데이터 기반 페이지 간 이동을 내비게이션 방식으로 하는 UI
성능	UI 로직이 로컬에 상주하므로 앱 응답이 빠르고 오프라인 액세스 가능	모든 코드가 서버에서 실행되므로 네트워크 접속 환경에 따라 성능 좌우됨

'웹 앱'과 '네이티브 앱'의 비교

기능	웹앱	네이티브 앱
개발환경	기존에 사용하던 웹 개발환경, 모든 운영체제 가능	아이폰(매킨토시 필요) 안드로이드폰(멀티 플랫폼)
사용 언어	HTML,CSS,자바스크립트	아이폰(Object-C) 안드로이드폰(자바)
앱 개발자 비용	무료	년 \$99(아이폰), \$35(안드로이드)
배포	브라우저가 설치된 환경	앱 스토어/안드로이드 마켓
결제 시스템	독자 결제 구축 또는 광고수익	앱 장터 판매 수익
하드웨어 지원	제한적	모든 기능 활용
업데이트	즉시 반영	검수 받아야 함
UI	상대적으로 제한적	풍부한 UI 가능

하이브리드 앱

- ❖ '웹 앱' 이면서 '네이티브 앱'의 장점을 결합한 것은 '하이브리드 앱'
- ❖ 개발자가 모바일 플랫폼과 상관없이 웹 기술로 애플리케이션을 개발할 수 있으며, 필요 시 네이티브 API에 직접 액세스 할 수 있음
- ❖ 앱 개발자는 애플리케이션 네이티브에 관련된 기능을 직접 코딩하거나 PhoneGap 과 같은 기존 솔루션을 활용

하이브리드 앱의 장/단점

장점

- 웹 표준을 지원하는 플랫폼에서 큰 수정없이 실행 가능함
- 패키징을 통해 다양한 기기를 위한 앱을 작성할 수 있음
- 네이티브 앱과 유사한 UX/UI를 제공할 수 있음

단점

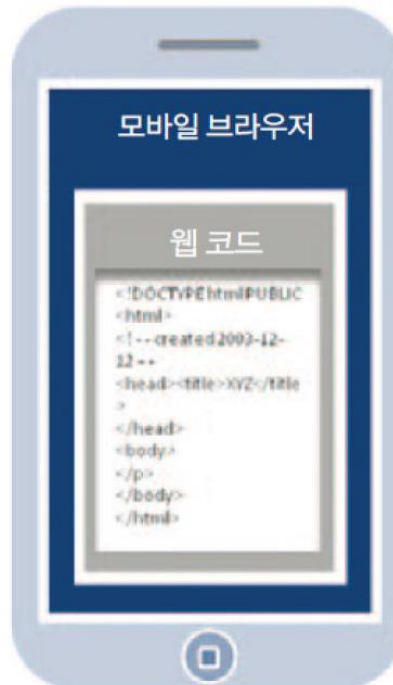
- 플랫폼이 가진 모든 기능을 사용하지 못함
- 네이티브 앱에 비해 낮은 실행 성능을 가짐
- 모든 기기에서 동일한 수준의 앱 표준을 지원하지 않음

3가지 방식 비교

네이티브 앱



웹 앱



하이브리드 앱



3가지 방식 비교 (계속)

특성	네이티브 앱	하이브리드 앱	웹 앱
개발 언어	네이티브 전용	네이티브 및 웹 또는 웹 전용	웹 전용
코드 이식성 및 최적화	없음	높음	높음
디바이스 고유의 기능 액세스	높음	중간	낮음
기존 지식 활용도	낮음	높음	높음
유려한 UI	높음	중간	중간
업그레이드 유연성	낮음(항상 앱스토어 이용)	중간(일반적으로 앱스토어 이용)	높음
설치 경험 수준	높음(앱스토어에서)	높음(앱스토어에서)	중간(모바일 브라우저 이용)

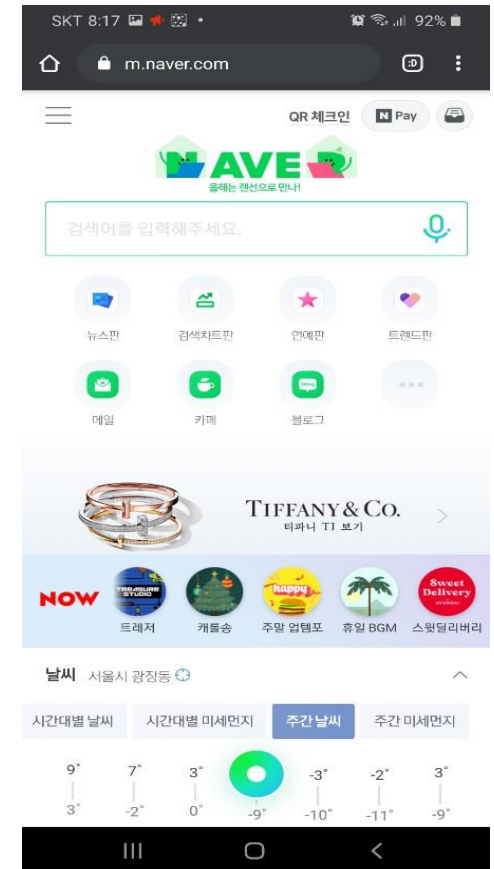
네이티브 앱, 웹 앱, 하이브리드 앱의 예



네이티브 앱 : 유튜브



하이브리드 앱 : 네이버



웹 앱 : 네이버

하이브리드 프레임워크 : 개발 프레임워크

PhoneGap <https://phonegap.com/>

- 웹 콘텐츠를 보여 주는 웹뷰(WebView)에서 스마트폰 디바이스에 접근하여, 카메라나 마이크 제어 같은 웹 브라우저 고유 기능 이외의 기능을 실행할 수 있는 인터페이스를 제공하는 대표적인 하이브리드 앱 개발 솔루션
- 캐나다에 있는 [Nitobi](#)라는 회사에서 처음 제작했으며 2011년 10월에 Adobe 사가 PhoneGap을 인수
- 가장 많은 스마트폰 플랫폼(iOS, Android, Windows Phone, Blackberry 등 7개 운영체제)을 지원
- 폰갭으로 개발된 앱 [Try it!](#)



하이브리드 프레임워크 : 개발 프레임워크 (계속)

❖ Titanium <https://www.appcelerator.com/Titanium/>

- Web App을 Native App으로 변환시켜주는 툴
- 개발은 HTML, CSS, JavaScript를 사용하지만 결과물은 네이티브 코드로 변환
- 디바이스 기능을 자체적으로 제공하고 Titanium의 자바스크립트를 사용
- 네이티브 코드로 변환하기 위해서는 Titanium에서 제공한 API 만 사용해야 하는 단점

하이브리드 프레임워크 : UI 프레임워크

❖ jQuery Mobile www.jquerymobile.com

- HTML 작성처럼 Line by Line 으로 마크업하며 작성
- 기존 Web 개발과 차이 적어서 익숙한 개발방법
- 많은 브라우저(기기)를 지원
- 기존 Web의 손쉬운 Mobile 전환
- 손쉬운 사용법

❖ Sencha Touch <http://www.sencha.com/products/touch>

- Java의 AWT나 Swing 처럼 Panel 기반으로 화면을 구성
- JavaScript와 Ext.JS 을 배우는 비용이 추가로 필요
- 네이티브 앱과 매우 유사한 UI를 제공

참고 자료

- ❖ 모바일 앱 개발 방식 비교 : 네이티브, 웹, 하이브리드, IBM 소프트웨어 Thought Leadership 백서
- ❖ NAVER D2, PhoneGap을 이용한 앱 개발,
<https://d2.naver.com/helloworld/8180>
- ❖ 하이브리드 앱,
<https://m.blog.naver.com/lool2389/220908482525>