

모바일 환경에 대하여

모바일(Mobile)은 “이동하는” 또는 “움직임이 자유로운” 이라는 뜻으로, 휴대성, 이동성의 특징을 지닌 기기를 말합니다.

즉, 현재 많이 사용하고 있는 스마트폰, 태블릿PC 가 모바일이라는 범주에 속하는 기기로, 모바일 웹이라는 것은 이 모바일 범주에 속하는 기기에 최적화된 최적화된 웹을 지칭한다고 할 수 있습니다.



모바일 웹과 모바일 앱

모바일 앱도 모바일 범주에 속하는 기기에 최적화된 앱을 말합니다.

여기서의 **앱(App)** 이란 운영체제 혹은 플랫폼에서 제공하는 **SDK(Software Development Kit)**를 기반으로 개발된 애플리케이션을 지칭하는 용어이며 일반적으로 **네이티브 앱(Native App)**을 말합니다. 그리고 네이티브 앱은 사용자가 아이폰 앱스토어 또는 구글 안드로이드 마켓에서 다운로드하여 사용하는 앱을 네이티브 앱이라고 합니다.

모바일 웹과 모바일 앱의 특징과 차이점

모바일 웹은 HTML 기반의 웹 브라우저로 동작하는데요, PC 웹하고의 가장 큰 차이점은 화면 사이즈가 작고, 터치 스크린의 인터페이스를 가지며, 'Click-to-Call'과 같은 모바일에 특화된 기능을 제공한다는 점입니다.

모바일 앱은 디바이스에 직접 설치(다운로드) 후 사용하는 방식입니다.

플랫폼(iOS, Android)에 따라 이용 가능한 서비스가 구분되며, 컨텐츠는 웹 콘텐츠를 활용(하이브리드 앱)거나 인터넷 연결 없이도 이용 가능합니다.

모바일 웹과 모바일 앱은 구동방식이 다르고, 운영체제 및 플랫폼에 따라 다르기 때문에 각각의 일장일단이 있습니다.

다시 정리하자면, 모바일 웹과 모바일 앱은 서비스나 콘텐츠를 제공한다는 역할을 같지만 동작하는 플랫폼과 구현 방식에 따라 구분됩니다.

이러한 기술적인 차이로 서비스나 콘텐츠의 용도에 따라 모바일 웹으로 개발할 것인지 모바일 앱으로 개발할 것인지를 나뉘게 됩니다.

모바일 웹

- 대체로 새로운 정보를 지속적으로 제공하는 것이 목적인 제품 리뷰, 블로그, 소식, 뉴스등과 같은 업데이트가 지속적으로 발생하는 콘텐츠를 제공하고자 하는 경우에 모바일 웹을 사용
- 모바일 웹은 웹 브라우저에서 동작하는 서비스라서 사용자는 URL 정보만 알면 사용 가능하다.
- 웹 표준을 따르면 다양한 플랫폼에서 동작하기 때문에 개발 생산성이 높고 유지보수가 쉽다.

모바일 앱

- 게임, 음악과 같이 하드웨어 성능이 높아야 하거나 모바일 기기의 하드웨어를 제어해야 하는 영역에서 모바일 앱을 주로 사용한다
- 모바일 앱은 운영체제 또는 플랫폼에 따라 각기 다르게 개발해야 하기 때문에 개발 생산성이 낮고 유지보수하기가 어렵다.
- 플랫폼에 따라 각기 다른 앱 스토어에서 다운로드해야하기 때문에 배포, 설치가 쉽지 않다.
- 모바일 웹에 비해 뛰어난 성능과 모바일 기기의 하드웨어를 제어할 수 있다는 점은 모바일 앱의 가장 큰 장점이다.

모바일 웹의 한계 극복

모바일 웹은 모바일 앱에 비해 실행 속도가 느리고 모바일 기기의 하드웨어를 제어할 수 없다는 모바일 웹의 단점을 극복하려는 노력이 계속되고 있으며 브라우저, 렌더링 엔진, 자바스크립트 엔진(V8)의 성능향상, 하드웨어 가속 기술에 대한 향상이 이루어지고 있습니다.

모바일 앱 개발 방식에 대하여

모바일 앱을 제작하는 경우에도 서비스의 목적, 여러 환경에 따라 어떤 방법으로 개발할 것인지로 나뉘게 됩니다.

네이티브 앱(Native App)

위에 서 언급했듯이 네이티브 앱은 SDK를 기반으로 개발된 애플리케이션 즉, 모바일 플랫폼 API를 이용해 개발하는 앱을 말합니다.

그리고 모바일 기기에 직접 다운로드하여 로컬(기기)에 저장되는 실행 파일로 사용됩니다.

모바일 기기에 최적화된 언어로 개발되며, 안드로이드 SDK를 이용해 java 언어로 만드는 안드로이드 앱과 iOS SDK를 이용해 Objective-C 언어로 개발된 아이폰 앱들이 여기에 속하게 됩니다.

그리고 다음과 같은 특징이 있습니다.

- 고성능의 그래픽 처리가 가능해서 2D 및 3D 게임이나 증강현실과 같은 앱을 개발할 수 있다.
- 특정 플랫폼에서만 동작하여 앱스토어를 통해 업데이트가 가능하기 때문에 업데이트가 느린 단점이 있다.
- 해당 운영체제나 플랫폼이 다르면 많은 시간과 비용을 감수하여 새롭게 개발해야하는 단점이 있다.

네이티브 앱의 웹뷰(WebView)

웹뷰는 네이티브앱 내에서 웹 기술을 사용할 수 있는 환경, 즉 네이티브 앱에 내장된 브라우저를 말합니다.

웹 앱(Web App)

웹 앱은 모바일 웹과 네이티브 앱을 결합(웹 + 앱)한 것으로 모바일웹의 특징을 가지면서 네이티브앱의 장점도 가지고 있습니다.

다시말해, 웹 기술로 구현하는 앱을 말합니다. 모바일웹 보다는 조금 더 모바일에 최적화된 앱이라고 할수 있습니다.

웹 앱의 일반적인 정의는 '브라우저를 이용한 HTML, CSS, 자바스크립트를 사용해 만들어진 애플리케이션' 이기 때문에 사용자가 직접 모바일 브라우저의 주소 창에 웹 주소를 입력해 사용하는 인터넷 서비스도 웹앱의 범주 안에 있습니다.

사용자가 웹사이트의 주소를 입력하게 하지 않고 스마트폰의 홈에 아이콘을 위치시켜 마치 네이티브앱처럼 아이콘을 클릭할 때 앱이 실행되도록 할 수도 있고 태그로 브라우저 주소 입력창 등을 숨길 수도 있어 겉모습과 동작 과정을 네이티브앱처럼 보이게 할 수도 있습니다.

웹앱은 다음과 같은 특징이 있습니다.

- 모바일웹처럼 HTML, CSS, Javascript, JSP, PHP, ASP, ASP NET 등 일반적인 웹 기술로 개발
- 웹 개발 기술을 사용하되 모든 UI와 UX를 앱과 유사하게 제작
- 모바일 브라우저에서 실행
- 모바일 웹보다 실행 속도가 빠름
- 네이티브 앱처럼 설치형이 아니기 때문에 웹앱을 실행하기 위해서 사용자가 브라우저를 열고 URL로 접근
- 첫 페이지는 웹으로부터 풀 브라우징으로 받고, 그 이후부터 AJAX 통신으로 모든 것 처리, 화면 전환시 애니-메이션 효과, 사용자 터치 이벤트 처리, AJAX 통신, HTML5 로컬 저장소 이용하여 네이티브 앱과 유사한 실행 환경 제공

그리고 웹앱과 모바일 웹에 다음과 같은 공통점이 있습니다.

- HTML, CSS, 자바스크립트 기반의 웹 기술을 똑같이 사용한다.
- 모바일 기기의 하드웨어에서 제공하는 센서, 카메라 등을 사용할 수 없다.
- 스마트폰 운영체제 혹은 플랫폼에서 제공하는 API 를 사용할 수 없다.
- 운영체제 혹은 플랫폼, 그리고 브라우저에서 사용할 수 있도록 제공하는 API 만을 사용한다

하이브리드 앱(Hybrid App)

위의 웹앱의 단점을 보완한 것이 바로 하이브리드앱입니다.

다시 말해, 하이브리드 앱은 단어 그대로 네이티브 앱과 웹 앱 기반의 기술을 함께 사용한다는 것을 의미합니다.

네이티브 앱을 개발할 경우 아이폰이든 안드로이드든 각 운영체제 혹은 플랫폼에서 제공하는 프로그래밍 언어로 제작해야 합니다.

이는 해당 운영체제나 모바일 기기에 최적화된 기능을 제공하는 점에서 이점이 있지만 운영체제나 플랫폼이 다르면 많은 시간과 비용을 감수하면서 새롭게 개발해야 하는 단점도 지니고 있습니다.

하지만 하이브리드 형태의 앱을 만들면 이는 달라집니다.

앱의 기반이 되는 콘텐츠 영역은 HTML 기반의 웹 앱으로 제작을 하고 최종 앱 배포에 필요한 패키징 처리만 아이폰, 블랙베리, 안드로이드의 플랫폼 안에서 하는 것입니다.

이는 프로그래밍 언어가 다르다는 사실과 관계없이 편리하게 다양한 앱을 제작할 수 있는 환경을 제공하며 시간과 비용도 절감할 수 있습니다.

이렇듯, 웹 퍼블리셔도 스스로 앱을 디자인하고 개발해서 앱스토어, 안드로이드 마켓등에 배포할 수 있습니다.

요약하자면, 하이브리드앱은 웹앱을 네이티브앱으로 포장하는 것을 말하는데 웹기술로 개발되었지만 모바일에 최적화된 언어로 만드는 네이티브앱처럼 보이게 하는 것으로 하이브리드앱은 네이티브앱의 형태를 띠고 있어 앱스토어나 마켓에서 다운받아 모바일에 설치할 수 있으며, 모바일의 고유정보를 이용하고 하드웨어를 제어할 수도 있습니다.

현재 즐겨쓰는 네이버앱 다음앱이 하이브리드앱 이라고 할 수 있습니다.

그리고 하이브리드 앱을 제작할 경우에는 하이브리드 앱을 지원하기 위해서는 별도의 하이브리드 프레임워크가 필요합니다.

대표적으로 해외에서 개발한 폰갭(PhoneGap), 티타니움(Titanium) 등이 있고 국내에는 KTH 가 제공하는 앱스프레소(Appspresso) 가 있습니다.

폰갭

웹을 이용한 크로스플랫폼 모바일 앱 제작 플랫폼으로서 WebView를 이용해 웹 콘텐츠를 표현할 수 있도록 하고 폰갭 고유의 자바스크립트 API를 만들어 이 자바스크립트 API에서 카메라나 가속도계 같은 장치를 제어할 수 있도록 한 것입니다.

일종의 HAL(Hardware Abstraction Layer)라고도 할 수 있는데 네이티브 API와 웹 기술을 사용했기 때문에 하이브리드앱이라고도 합니다.

물론 네이티브앱을 개발하면서도 WebView를 사용하는 경우도 많지만 폰갭의 정체성은 크로스플랫폼이기 때문에 네이티브앱 개발 지식이 없어도 앱을 개발할 수 있다는 점에서 구별되고 있습니다.

모바일 앱 개발 방법의 경계

이렇게 모바일 앱을 개발하는 방식은 다양합니다.

그리고 엄밀히 따지자면 분명한 차이점은 있지만 커다란 개발 방법론 범주안에 모바일 웹을 포함시키기도 합니다.

이러한 네이티브앱, 하이브리드앱, 웹앱의 경계를 명확히 구분하는 것은 경우에 따라서 큰 의미가 없는 일이 되기도 합니다.

이런 상황에서 콘텐츠 제공 위주의 앱을 개발할 때 웹을 사용하는 것은'적용 여부'의 문제가 아니라'얼마나 사용할 것이냐'는 정도의 문제가 될 수 있습니다.

네이티브 API를 사용하면 웹을 사용하는 것보다 더 좋은 성능으로 더 미려한 UI를 제공할 수 있고 장치 제어를 원활하게 할 수 있지만 그만큼 개발에 부담이 따르게 됩니다.

하지만 하드웨어 성능의 향상과 브라우저 성능 개선으로 네이티브앱과 웹앱과의 성능 차이는 갈수록 줄고 있고 웹브라우저에서 장치 제어를 할 수 있는 다양한 방법이 제시되고 있기도 하기 때문에 이 개발 방법에 따라 네이티브앱이나 웹앱, 하이브리드 앱이냐를 구분하는 경계가 더욱더 모호해질 것으로 보입니다.

Responsive Web Design(반응형 웹)

이던 마코트(Ethan Marcotte)라는 웹 디자인 선구자가 이름 지은 '**반응형 웹 디자인(Responsive Web Design)**'이라는 기법은 브라우저 창의 폭에 따라 레이아웃을 바꿀 수 있도록 하는 기술입니다.

반응적 웹 디자인은 약자로 RWD 라고도 하며, 디스플레이 크기에 맞는 가장 읽기 쉬운 레이아웃의 페이지를, 여러 버전의 사이트를 만들지 않고도 사용자에게 제공할 수 있도록 합니다.

RWD는 하나의 기술로 이루어진 디자인 기법이 아니며, 몇 가지 CSS와 HTML 기술을 결합하여 화면 크기에 맞는 레이아웃의 페이지를 생성할 수 있도록 합니다

RWD에는 세 가지 개념이 결합되어 있습니다.

- 레이아웃에는 유연한 격자(flexible grid)를 사용한다.
- 이미지와 비디오에는 유연한 미디어(flexible media)를 사용한다.
- 화면 너비에 맞는 스타일을 만드는 부분은 CSS 미디어 질의(media query)를 통해 해결한다.

유연한 격자를 사용하게 되면 고정폭 레이아웃은 만들지 않습니다. 스마트폰 화면은 크기가 다양하므로, 고정폭 레이아웃은 전혀 어울리지 않기 때문입니다. 대신, 디스플레이 크기에 맞게 페이지를 키우거나 줄일 수 있어야 합니다.

이는 유동적 레이아웃 개념과 관련있습니다.

그리고 유연한 미디어는 이미지나 비디오가 주어진 영역에 맞게 표시될 수 있도록 해야합니다.

즉, 큰 모니터에는 사진을 크게 출력하고, 작은 화면에는 작게 출력하는 것입니다.

마지막으로, 미디어 질의는 브라우저에 보내는 스타일을 현재 상태에 따라 달라 할 수 있는 CSS 테크닉입니다.

예를 들어, 창의 너비가 480픽셀 이하일 때는 특정한 스타일이 렌더링되도록 하고 760픽셀 이상일 때에는 그와는 다른 스타일이 적용되도록 할 수 있습니다. 게다가, 폭만을 기준으로 삼을 수 있는 것은 아닙니다. 태블릿이 가로보기 모드일 때 적용되는 스타일과, 아이폰이나 아이패드에서 사용된 레티나 디스플레이처럼 픽셀 집적도가 높은 디스플레이에 적용될 스타일도 구별할 수 있습니다.

반응형 웹은 다음과 같은 특징이 있습니다.

- N-Screen 시대에 맞추어 다양한 디바이스에 최적화 된 화면 제공한다.
- 모바일과 PC의 홈페이지를 하나로 운영하여 제작/관리비가 저렴하다.
- 하나의 URL을 가져 검색엔진의 인덱싱과 노출에 유리하다.
- 로딩속도 지연을 발생시키는 리디렉션이 발생하지 않아 로딩시간이 줄어 사용자들이 더 쾌적한 사이트 방문경험하도록 제공한다.

N screen(Network Screen 또는 Numbers Screen, N 스크린)은 하나의 콘텐츠를 여러 디바이스(multiple devices)에서 소비할 수 있는 환경을 말한다.

예를 들어, 하나의 영화를 PC 모니터에서 감상하다 모바일에서 이어서 볼 수 있고 텔레비전 모니터에서 볼 수 있는 환경을 N 스크린이라고 한다.

Adaptive Web Design

반응형은 일반적으로 어떤 변화에도 빠르고 적극적으로 반응하는 것을 의미하는 반면 적응형은 (변화와 같은) 새로운 목적이나 상황에 쉽게 변경되는 것을 의미합니다. 반응형 디자인에서는 웹사이트가 뷰포트 너비와 같은 요인에 지속적이고 유동적으로 변하지만, 적응형 웹사이트는 미리 정해진 요인들에 맞추어 만들어집니다.

- 기준이 되는 스크린 사이즈에 미디어 쿼리나 스크립트를 활용하여 레이아웃을 변경, 화면을 구성한다.
- 반응형 웹과 유사한 개념이지만, 반응형 웹이 유동적인 그리드를 기준으로 화면을 구성하는 것과 달리 적응형 웹은 미리 정해진 몇 개의 스크린 사이즈를 기준으로 화면을 구성한다.
- 반응형 웹 에 비해 적은 기획과 소스가 필요한 반면 모든 스크린 사이즈에 최적화 시키기에는 정교함이 덜하다.