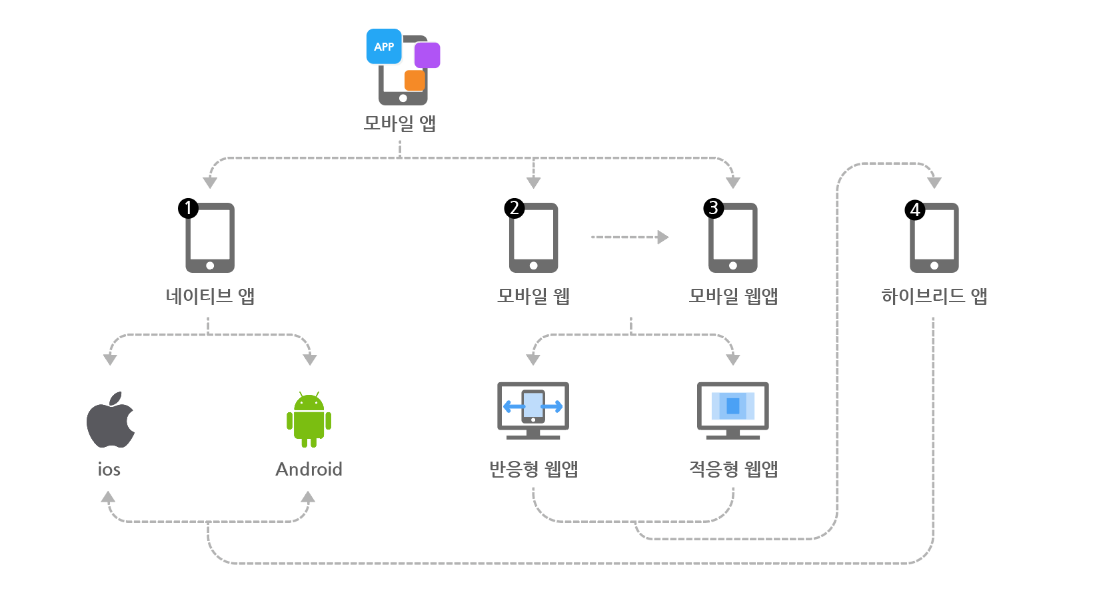
**Study Note for Android - 개요**

안드로이드(Android)? 휴대용 장치를 위한 운영체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램 등을 포함하고 있는 소프트웨어 스택이자 모바일 운영체제

애플리케이션(Application)? 응용 소프트웨어, 특정 목적을 위해 제작된 프로그램

모바일 애플리케이션(Mobile Application)? 모바일 환경에서의 애플리케이션

<모바일 애플리케이션의 종류와 구분>



1. 네이티브 앱
   1. 안드로이드 SDK(Software Development Kit)의 Java, Kotlin을 이용해 개발하거나, IOS SDK의 Swift, Object C를 이용해 개발된 앱으로 모바일 기기에 최적화된 앱
   2. 장점
      1. 성능이 매우 우수함
      2. 네이티브 API를 호출하여 사용함으로 플랫폼과 밀착되어 있음
   3. 단점
      1. 플랫폼에 한정적이며 요구하는 언어에 제약적임
2. 모바일 웹
   1. 데스크탑 브라우저에 실행되는 웹 애플리케이션을 모바일 스크린 크기로 줄여 놓은 것
   2. 페이지를 호출할 때마다 서버에서 정보를 다운로드해야 하는 “풀브라우징” 방식이므로 로딩 속도가 늦고, 모바일 특성상 이동 시에 서버 접속 장애가 발생할 수 있어서 매우 불안정한 앱임
3. 모바일 웹앱
   1. 모바일 웹에 비해 모바일에 최적화되어 네이티브 앱화 된 것
   2. 모바일 웹에 사용되는 기술적인 언어를 사용하지만 “풀브라우징” 방식이 아닌 단일 페이지 모델을 사용해서 전환되기 때문에 모바일 웹보다는 빠른 실행 속도를 유지함
   3. 장점
      1. 웹사이트를 보는 것이기 때문에 따로 설치할 필요가 없음
      2. 모든 기기와 브라우저에서 접근할 수 있음
      3. 별도 설치 및 승인과정이 필요치 않아 유지보수가 용이함
   4. 단점
      1. 플랫폼의 API를 사용할 수 없고 오로지 브라우저 API만 사용 가능
      2. 친화적인 터치 앱을 개발하기가 까다롭고 번거로움
      3. 네이티브, 하이브리드 앱에 비해 실행이 까다로움 (브라우저를 열고 검색해 실행)
4. 하이브리드 앱
   1. 앱의 기반이 되는 콘텐츠 영역은 HTML 기반의 웹 앱으로 제작, 최종 앱 배포에 필요한 패키징 처리만 안드로이드, 아이폰 플랫폼 안에서 처리한 애플리케이션
   2. 장점
      1. 네이티브 API와 브라우저 API를 이용한 다양한 개발이 가능
      2. 웹 개발 기술을 이용해 앱을 개발할 수 있음
      3. 한 번의 개발로 다수의 플랫폼에 대응할 수 있음
   3. 단점
      1. 네이티브 기능에 접근하기 위해선 결국 네이티브 개발 지식이 필요함
      2. 웹 뷰에서 앱을 실행하는 경우이기 때문에 앱의 성능이 곧 브라우저의 성능임
      3. UI 프레임워크 도구를 사용하지 않는다면 개발자가 UI를 제작해야 함

SDK(Software Development Kit)? 특정 프로그래밍 언어로 코딩하여 프로그램을 만드는데 필요한 소프트웨어 번들(ex. NDK : Native Development Kit, JDK : Java Development Kit)

IDE(Integrated Development Environment)? 통합 개발 환경, 공통된 개발자 툴을 하나의 GUI로 결합하는 애플리케이션을 구축하기 위한 소프트웨어(소스코드 편집기 + 로컬 빌드 자동화 + 디버거)

안드로이드 스튜디오(Android Studio)? Android 애플리케이션 개발을 위한 IDE

플랫폼(Flatform)? 특정 장치나 시스템, 서비스 등에서 이를 구성하는 기반이 되는 하드웨어나 소프트웨어 환경, 더 크게는 기초가 되는 틀이나 골격을 지칭

라이브러리(Library)? 사전에 만들어진 코드 집합, 응용 프로그램 개발을 위해 필요한 기능을 모아 놓은 소프트웨어, 개발자는 필요 시 공개된 라이브러리를 자신의 프로그램에 호출하여 사용할 수 있음

프레임워크(Framework)? 소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔 일련의 협업화된 형태로 클래스들을 제공하는 것, 라이브러리와 달리 애플리케이션의 틀과 구조를 결정할 뿐 아니라 그 위에 개발된 개발자들의 코드를 제어함, 프레임워크는 구체적이며 확장 가능한 코드를 가지고 있으며 설계자가 의도하는 여러 디자인 패턴의 집합으로 구성되어 있음

API(Application Programming Interface)? 응용프로그램에서 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스