# Hybrid Application

Hybrid applications are web applications (or web pages) in the native browser, such as UIWebView in iOS and WebView in Android (not Safari or Chrome). Hybrid apps are developed using HTML, CSS and Javascript, and then wrapped in a native application using platforms like [Cordova](http://cordova.apache.org/). This allows you to use any web-native framework you want, and there are plenty of these.

The application development is faster, **simpler**, more **rapid** and the application is easier to maintain. You can change platforms anytime you need, Cordova lets you build your application for more than one platform just by one adding line of code. As for the phone hardware such as the camera or Bluetooth, Cordova has a large repository of [plugins](http://plugins.cordova.io/#/) you may use.

The main problem with hybrid apps is that they still depend on the native browser, which means they are not as fast as native apps.

Developing with Cordova is just like developing a webpage. You create HTML, CSS and JavaScript local files, test them in the browser and then wrap them in a native web view with Cordova (you’ll still need native SDKs and development tools for this step).

# Merits of hybrid

By far the single biggest benefit that hybrid mobile apps can offer is the unified development.

Those who need to have their app in the App Store as fast as possible should seriously consider using hybrid applications.

Basic web applications are cut off from smartphones’ operating systems and built-in functionality. Even though they are getting smarter every day, they still don’t come anywhere near native applications.

# Demerits of hybrid

## Performance:

Hybrid apps add an extra layer between the source code and the target mobile platform: the particular hybrid mobile framework, such as Ionic, Cordova, Onsen, Kendo, and many others. The unsurprising result is a possible loss of performance. It really varies from application to application just how noticeable the difference can be, but the fact that Facebook migrated their mobile application from HTML5 to native shows that there really can be a significant difference, at least for large-scale applications. Mark Zuckerberg even went on to say that “The biggest mistake we’ve made as a company is betting on HTML5 over native.”

After all, 84 percent of users consider performance to be an important or very important factor, according to A Global Study of Consumers’ Expectations and Experiences of Mobile Applications by Dynatrace, an American application performance management (APM) software company with products aimed at the information technology departments and digital business owners of medium and large businesses.

## Debugging:

## 

That extra layer also makes debugging a potential nightmare. Developers have to rely on the framework itself to play nicely with the targeted operating system and not introduce any new bugs. Since developers are not likely to have a deep knowledge of the targeted platform, figuring out the exact cause of an issue can be a lengthy affair.

## Features:

Companies who want to stand at the very apex of progress and use all the latest and greatest features and hardware capabilities are probably going to experience difficulties trying to achieve their goals using hybrid frameworks. It can take quite a bit of time before new features are implemented by providers of these providers of these frameworks.

## Language:

Native and Web / web only.

**User Interface:**

Never give users a fully native experience due to the necessity to record web technology usage.

**Access to native APIs**

Moderate.

## Security:

Problems connected with the usage of additional layer, that is special framework for hybrid app writing. So compare to native apps, hybrid apps are less security like payment gateway.

**Compatibility with other apps on the device:**

Very Less.

## 