**Sources**

[**https://medium.freecodecamp.org/progressive-web-apps-bridging-the-gap-between-web-and-mobile-apps-a08c76e3e768**](https://medium.freecodecamp.org/progressive-web-apps-bridging-the-gap-between-web-and-mobile-apps-a08c76e3e768)

[**https://developers.google.com/web/ilt/pwa/offline-quickstart-slides**](https://developers.google.com/web/ilt/pwa/offline-quickstart-slides)

[**https://whatwebcando.today/**](https://whatwebcando.today/)

[**https://medium.com/progressive-web-apps/2018-state-of-progressive-web-apps-f7517d43ba70**](https://medium.com/progressive-web-apps/2018-state-of-progressive-web-apps-f7517d43ba70)

[**https://medium.freecodecamp.org/engage-your-users-and-enhance-their-experience-with-progressive-web-apps-de0e0bfb2fbf**](https://medium.freecodecamp.org/engage-your-users-and-enhance-their-experience-with-progressive-web-apps-de0e0bfb2fbf)

* **What are Progressive Web Applications**?
* The reasons for PWA vs native applications?
* Who are using them?
* The Main characteristics of Progressive Web Apps.
* Support for PWA’s.

Progressive Web Apps are a hot topic right now because its support is increasing in multiple platforms, and major companies are deciding to work on PWA versions of their web apps including Twitter, Lyft, Starbucks, NASA, Forbes, The Weather Channel and more.

**What are Progressive Web Applications?**

What are progressive web apps? Progressive Web Apps (PWA) are experiences that combine the best of the web and the best of apps. A progressive web application isn’t one thing to install but rather a collation of features. Features that can be added to any web application as enhancements. It’s not an all or nothing scenario, the core features of a progressive web apps can be fully adopted or individually adopted.

What makes progressive web apps special is really the fact that these technologies have been pulled together to serve this purpose and give it a name. Alex Russell, one of the forefathers of progressive web apps, states that, "it happens on the web from time to time that powerful technologies comes to exist without the benefit of marketing departments or slick packaging. They linger and grow at the peripheries, becoming old-hat to a tiny group while remaining nearly invisible to everyone else. Until someone names them." This phenomenon that Alex is talking about has happened several times in the past. Many different developers used XMLHttpRequest in their earlier versions of IE. But Ajax didn't become a thing until Jesse James Garrett gave it that name. Likewise, it took Ethan Marcotte to coin the term "responsive web design" to change the way that we think about content. So Alex Russell has given the name "progressive web app" to this collection of technologies. So what are they? What is a progressive web app? Well he lays out nine attributes of what makes something a progressive web app. First, it's responsive. Your application will fit any form vector just like Ethan Marcotte coined the term. Next, it's connectivity independent. It'll work online or offline. Or even on a spotty network, which is probably the most important. Third is app-like interactions for both the navigations and the way that you use the app feel like they're native. Fourth is that the app is fresh. It's always up to date, and those updates happen transparently thanks to the web and service workers. Next, it's safe, and it's safe because it's served over transport layer security or HTTPS to make sure that no third parties are snooping on your traffic. Sixth, Russell says that they're discoverable, and that they're identified as applications thanks to the W3C manifest specification that we'll be looking at in a little bit, as well as search engines that are able to find them. They're re-engageable. They're applications that can reach into their host operating system and tie into the mechanisms that are available there to re-engage you, most commonly thought of as push notifications. They're installable allowing you to add them to your home screen or to whatever desktop metaphor is being used in your OS. And, finally, they're linkable. There's no friction to install, no friction to access. A URL is the entry point at which you will find and access an application. So effectively, progressive web apps are just websites that took all the right vitamins. They started off as a website, and they grew up strong to become applications. And these applications can be especially strong, stronger than most people realize. The capabilities that the browser makes available to these types of applications is extensive and growing all the time. Let me tell you about just a few of them. First of all, you have a network proxy available to you to change, sniff, cache, do anything that you want to with traffic available on service worker. There's app packaging so that these things can be installable with W3C manifests. Push notifications are available. And background syncing so that you can do some offline tasks to get information from the network or send information up to some API somewhere. All of these will be covered in this Pluralsight course. Additionally, there's local storage so you can save data on the device. There's the gamepad API which allows you to interact with joy sticks and controllers that may be plugged into the machine. The page visibility API lets us know if the user is physically looking at your application. There's the ability to capture audio and video and photos from the camera and microphone, as well as the ability to play back audio and video with media playback. We can make devices vibrate. We can check the battery status and go into low power mode for our applications if the user's about to run out of battery. Integrated payments are coming to make it easy to do one tap and pay for something in one of our applications. Credential management to make logging in easy. Streaming of networks to get really low-level access to networking. We have the ability to do peer-to-peer communications, access Bluetooth accessories, and even share things across the web with the new web share API that's being worked on. In fact, there're so many capabilities that there's a bit of a humorous website called youmightnotneedelectron.com, which just proves to you that with all of these capabilities maybe you don't really need to install some giant framework to build an app. Something to think about. Let's go ahead and take a look at progressive web apps in action.

Native app store apps have become hugely popular in the past through features such as push notifications, working offline, smooth animations and transitions, loading on the homescreen and so on

 just features of native apps that we add on to Web Apps to get the best of both worlds. Apps that you can access directly from the Web, yet work smoothly and faster, are installable, and may even have notifications.

Mobile Web Apps are JavaScript applications that work in the browser and attempted to bring some of the native apps features to the web, but weren’t able to provide push notifications for example. With the mobile introduction of new Web APIs, Progressive Web Apps are now closing the gap, providing the full app-like experience on the mobile web.

Support for PWA’s

Today, progressive web apps are fully supported by Chrome and Opera. Firefox supports nearly all of their features. Microsoft Edge is working on them. The Samsung Internet browser has been making great strides this past year and has shown a strong commitment to progressive web apps, as demonstrated by Samsung’s leadership in some of the key standardization work. Apple has finally jumped on the mobile web train: service workers, a key component for supporting progressive web apps, are available in Safari 11.1 for iOS 11.3 and macOS 10.13.4.