**ANDROID APP DEVELOPMENT (RESEARCH 1)**

**UNDER NIC**

**What is scripting language and why to use it so?**

A **scripting language** is used to write **scripts**. These contain a series of commands that are interpreted one by one at runtime unlike programming languages that are compiled first before running. Scripting languages just interpret the scripts during the program run. The scripts are used for the performance improvement or common application task fulfillment.

Scripting languages are:

* Perl
* PHP
* JavaScript
* Python, and others.

Nowadays, scripts are generally associated with web development where they are widely used to make dynamic web applications. Scripting languages can be divided into two categories:

* Server Side Scripting Languages
* Client Side Scripting Languages

**Server-side** scripting languages create the scripts that run on the server and hence minimize the workload of a browser. The functionality of your website is written in those scripting languages. The most commonly used server-side scripting languages are **Perl**, **Ruby**, **Python**, **PHP**, etc.

**Client-side** scripting languages create the scripts that run on the client side (i.e. your browser). These are sent from the server by server-side scripts. Some good examples are **JavaScript**, **jQuery**, **CSS** etc

**Scripting vs Programming Languages — Differences Overview**

So the main differences between scripting vs programming languages are as follows:

|  |  |
| --- | --- |
| **Scripting languages** | **Programming languages** |
| Platform-specific | Platform-agnostic (cross-platform) |
| (Mostly) interpreted | Compiled |
| Faster at runtime | Slower at runtime |
| More code-intensive | Less code-intensive |
| Creates standalone apps | Creates apps as part of a stack |

**What is a Native App?**

Native apps are developed for specific mobile operating systems such as iOS or Android. Platform-specific language is used to build these apps.  **For example,** (native) Android app uses Java, while iOS apps use Objective – C or Swift.

The best thing about native apps is: it leverages inbuilt features of a device such as Camera, GPS, gestures and so on.

Native apps are usually high on UX!

**Examples of Native App:**

The most popular messaging app – WhatsApp is a native app.  WhatsApp has different interfaces for its iOS and Android users.  Apart from WhatsApp, the following are also native applications:

* Facebook
* Gmail
* Twitter

**What is hybrid android application>**

The heart of a hybrid-mobile application is still just an application that is written with HTML, CSS, and JavaScript. However, instead of the app being shown within the user’s browser, it is run from within a native application and its own embedded browser, which is essentially invisible to the user. For example, an iOS application would use the WKWebView to display our application, while on Android it would use the WebView element to do the same function.

Hybrid apps are an extension of regular web apps. They just come wrapped up in the native app’s shell.

A hybrid app uses HTML5, JavaScript and CSS languages. This means it works like a regular website but feels and looks like a mobile app.

**Examples of Hybrid apps**

Some of the popular hybrid apps are:

* Instagram
* Yelp
* Uber
* Amazon Store

### Key Features: Native, Web, & Hybrid

| **Feature** | **Native** | **Web-only** | **Hybrid** |
| --- | --- | --- | --- |
| **Device Access** | Full | Limited | Full (with plugins) |
| **Performance** | High | Medium to High | Medium to High |
| **Development Language** | Platform Specific | HTML, CSS, Javascript | HTML, CSS, Javascript |
| **Cross-Platform Support** | No | Yes | Yes |
| **User Experience** | High | Medium to High | Medium to High |
| **Code Reuse** | No | Yes | Yes |

**Advantages of Hybrid Mobile App Development**

* Low-cost app development
* Fast results
* Less time on testing the app
* Easier usage of plugins
* A single open code used for multiple platforms
* Scalability: Hybrid mobile applications are limited only by the underlying framework

[**Top Hybrid Mobile Application Development Tools**](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Top_Hybrid_Mobile_Application_Development_Tools)

* [Ionic](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Ionic)
* [Onsen UI](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Onsen_UI)
* [Intel XDK](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Intel_XDK)
* [Sencha Touch](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Sencha_Touch)
* [Kendo UI](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Kendo_UI)
* [Framework 7](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Framework_7)
* [jQuery Mobile](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#jQuery_Mobile)
* [Mobile Angular UI](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Mobile_Angular_UI)
* [Famo.us](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Famous)
* [Monaca](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#Monaca)
* [Trigger.IO – the Dark horse](https://firebearstudio.com/blog/hybrid-mobile-application-development-tools.html#TriggerIO_8211_the_Dark_horse)

**WHAT IS RESPONSIVE CSS?**

It is called responsive web design when you use CSS and HTML to **resize, hide, shrink, enlarge, or move the content to make it look good on any screen**. Responsive web design provides an optimal experience, easy reading and easy navigation with a minimum of resizing on different devices such as desktops, mobiles and tabs).

## **Responsive structure**

Below image shows the responsive structure of web pages.

