# What Is a Front-End Developer?

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

[*https://en.wikipedia.org/wiki/Front-end\_web\_development*](https://en.wikipedia.org/wiki/Front-end_web_development)

Typically, a person enters into the field of front-end development by learning to develop HTML, CSS, and JavaScript which commonly runs in a [web browser](https://en.wikipedia.org/wiki/Web_browser) but can also run in a [headless browser](https://en.wikipedia.org/wiki/Headless_browser), [WebView](http://developer.telerik.com/featured/what-is-a-webview/), or as compilation input for a native runtime environment. These four run times scenarios are explained below.

##### Web Browsers (most common)

A web browser is software used to retrieve, present, and traverse information on the [WWW](https://en.wikipedia.org/wiki/World_Wide_Web). Typically, browsers run on a desktop or laptop computer, tablet, or phone, but as of late a browser can be found on just about anything (i.e, on a fridge, in cars, etc.).

The most common web browsers are (shown in order of [most used first](https://en.wikipedia.org/wiki/Usage_share_of_web_browsers#Summary_tables)):

* [Chrome](http://www.google.com/chrome/)
* [Safari](http://www.apple.com/safari/)
* [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) (Note: not [Edge](http://dev.modern.ie/), referring to IE 9 to IE 11)
* [Firefox](https://www.mozilla.org/firefox/)
* [Edge](https://www.microsoft.com/en-us/windows/microsoft-edge)

##### Headless Browsers

Headless browsers are a web browser **without** a graphical user interface that can be controlled from a command line interface programmatically for the purpose of web page automation (e.g., functional testing, scraping, unit testing, etc.). Think of headless browsers as a browser that you can run from the command line that can retrieve and traverse web pages.

The most common headless browsers are:

* [Headless Chromium](https://chromium.googlesource.com/chromium/src/+/lkgr/headless/README.md)
* [Zombie](https://github.com/assaf/zombie)
* [slimerjs](http://slimerjs.org/)

##### Webviews

[Webviews](http://developer.telerik.com/featured/what-is-a-webview/) are used by a native OS, in a native application, to run web pages. Think of a [webview](http://developer.telerik.com/featured/what-is-a-webview/" \t "_blank) like an iframe or a single tab from a web browser that is embedded in a native application running on a device (e.g., [iOS](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIWebView_Class/), [android](http://developer.android.com/reference/android/webkit/WebView.html), [windows](https://msdn.microsoft.com/library/windows/apps/windows.ui.xaml.controls.webview.aspx)).

The most common solutions for [webview](http://developer.telerik.com/featured/what-is-a-webview/" \t "_blank) development are:

* [Cordova](https://cordova.apache.org/) (typically for native phone/tablet apps)
* [NW.js](https://github.com/nwjs/nw.js) (typically used for desktop apps)
* [Electron](http://electron.atom.io/) (typically used for desktop apps)

##### Native from Web Tech

Eventually, what is learned from web browser development can be used by front-end developers to craft code for environments that are not fueled by a browser engine. As of late, development environments are being dreamed up that use web technologies (e.g., CSS and JavaScript), without web engines, to create native applications.

Some examples of these environments are:

* [Flutter](https://flutter.io/)
* [React Native](https://facebook.github.io/react-native/)