**What is a polyfill?**

*A polyfill, or polyfiller, is a piece of code (or plugin) that provides the technology that you, the developer, expect the browser to provide natively.*

Coined by [Remy Sharp](https://twitter.com/rem) - <https://remysharp.com/2010/10/08/what-is-a-polyfill>

*We, as developers, should be able to develop with the HTML5 APIs, and scripts can create the methods and objects that should exist. Developing in this future-proof way means as users upgrade, your code doesn't have to change but users will move to the better, native experience cleanly. From the HTML5 Boilerplate team on polyfills -*[*https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills*](https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills)

**Further research:**

<https://en.wikipedia.org/wiki/Polyfill>

**An example polyfill**

The code below is a polyfill for the new ES6 String method, startsWith():

**if** (!String.prototype.startsWith) {

String.prototype.startsWith = **function** (searchString, position) {

position = position || 0;

**return** **this**.substr(position, searchString.length) === searchString;

};

}

As you can see, a polyfill is just regular JavaScript.

This code is a simple polyfill (check it out on MDN), but there's also a significantly more robust one, [here](https://github.com/mathiasbynens/String.prototype.startsWith/blob/master/startswith.js)

**QUIZ QUESTION**

Why does the startsWith() polyfill begin with the following line?:

**if** (!String.prototype.startsWith)

* 

Without it, the script would throw an error.

* 

It checks to make sure the String.prototype exists.

* It avoids overwriting the native startsWith method.