**Currying**

[Currying](https://en.wikipedia.org/wiki/Currying) is an advanced technique of working with functions. Currying in JavaScript transforms a function with multiple arguments (also called arity) into a nested series of functions, each taking a single argument. Currying doesn’t call a function. It just transforms it.

**Why Currying:**

1. Currying helps you avoid passing the same variable multiple times
2. it helps you create a higher order function. { First, higher order functions can help improve the legibility of your code by making it more concise and easy to understand. This can help speed up the development process and make it easier to debug code. Second, higher order functions can help organize your code into smaller chunks, making it easier to maintain and extend. Means [single-responsibility principle](https://en.wikipedia.org/wiki/Single-responsibility_principle).

The **single-responsibility principle** (**SRP**) is a computer programming principle that states that "A module should be responsible to one, and only one, actor.

}

<https://builtin.com/software-engineering-perspectives/currying-javascript>

1. To avoid a load on call stack. Parent function will not stay in the call stack.=>Bilal Hussain.-----

A curried function have a built-in iterator behavior. One argument is applied at once which is then returned to the calling function to be used for next step. Read [here](https://codeburst.io/javascript-es6-iterables-and-iterators-de18b54f4d4) about iterators.

<https://towardsdatascience.com/javascript-currying-vs-partial-application-4db5b2442be8>