

JS NUGGETS

CURRYING IN JAVASCRIPT

WHAT IS CURRYING?

- Currying is a technique of evaluating function with multiple arguments, into sequence of function with single argument.
- Currying is a transformation of functions that translates a function from callable as $f(a, b, c)$ into callable as $f(a)(b)(c)$.

```
function curry(f) {  
  return function(a) {  
    return function(b) {  
      return f(a, b);  
    };  
  };  
}  
  
function sum(a, b) {  
  return a + b;  
}  
  
let curriedSum = curry(sum);  
curriedSum(1)(2);
```

```
// 3
```

WHY IS IT USEFUL? 🤔

- Currying helps you to avoid passing the same variable again and again.
- Little pieces can be configured and reused with ease.

How to convert an existing function to curried version?

The curry function does not exist in native JavaScript. But libraries like lodash makes it easier to convert a function to curried one.

```
let add = function(a, b, c) {
  return a + b + c;
};
let curried = _.curry(add);
let addByTwo = curried(2);
console.log(addByTwo(0, 0)); // 2
console.log(add(2, 1, 1)); // 4
console.log(curried(4)(5)(6)); // 15
```