CURRYING IN JAVASCRIPT

Two ways to curry functions :-

1. To use bind functions
2. Using the concept of closures.

NOTE:

Currying is a functional programming technique that involves breaking down a function that takes multiple arguments into a series of functions that take one argument each.

Example - Currying using bind functions

- Create a function 'multiply' which multiplies 2 parameters x and y.

```
let multiply = function (x, y) {
    console.log (x * y)}
```

- Create another method 'multiplyByTwo' which is exactly the copy of 'multiply' We use bind method for this don't care about first arg.

```
let multiplyByTwo = multiply.bind(this, 2)
multiply.By.Two (5);
```

means that this '2' will be set as 'x' of multiply method.

the arg we pass here refers to the 'y' of multiply method.

Similarly, another function 'multiplyByThree':-

```
let multiplyByThree = multiply.bind(this, 3)
multiplyByThree (5);
```

Output - in console

```
10
15
```

This is what currying is.
We can use 'multiply' method to ~~use~~ create other functions called multiplyByTwo, multiplyByThree etc.

Example : using function closures.

Let's create a function that returns another function.

```
let multiply = function (x) {
    return function (y) {
        console.log (x * y);
}}
```

```
let multiplyByTwo = multiply(2);
multiplyByTwo (3);
```

Similarly, we can do 'multiplyByThree' method