Additional : [+=, !=, for each, debugging]

1. Basic understanding of JavaScript

Roadmap!

① * Variables and data types
(Var, let, Const) (string, boolean, numbers)

② * Basic operators (+, -, >, <, >=, <=, !=, +=)

③ * Control structures (eg: "if", "else", loops)

2. Functions [parameters, arguments]

- * How to declare and invoke functions
- * Understanding of parameters and arguments
- * Return statements

3. Arrays:

* How to create arrays and access their "elements"
* Common array methods like `.push()`, `.pop()` and `.slice`

[to avoid common pitfalls / bugs] hoisting & scope

4. Objets:

* How to create objects and access their "properties"

* understanding of key-value pairs

5. Destructuring: 

6. Working environment: [Vs code, html/node]

functions:

```
function display() {
    return "Hi, there!";
}

var res = display();
console.log(res);
```

```
function display(message, name) {
    // "message", "name" parameters
    console.log(message + ", " + name + "!");
}

display("Hello", "Guru");
```

parameters

arguments

Parameters: (function definition)

* Function definition [dummy value]

* They are like placeholders for the values that the function can accept.

Arguments: = actual data (function call)

* These are the "real Values" passed to the function when you call it.

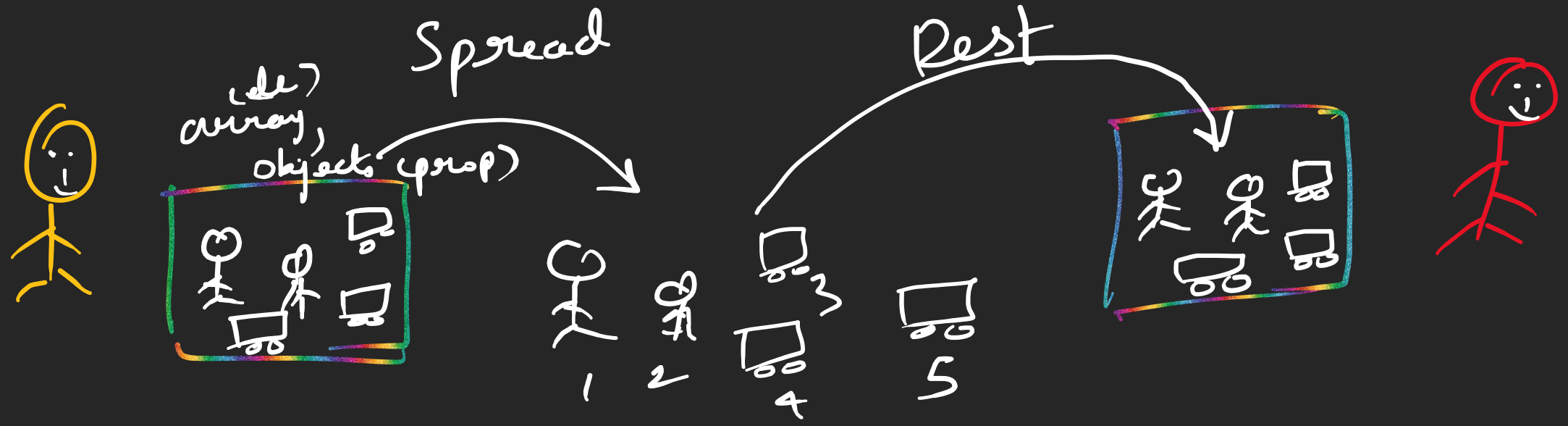
* "Arguments" are the actual data you provide to the function using its "parameters".

Spread & Rest : [ES6]

[...]

* Spread : [1,2,3] ⟹ 1 2 3

* Rest : 1 2 3 ⟹ [1,2,3]



(memory)

```
arrOld = [1, 2, 3];
```

literal copy!

(memory)

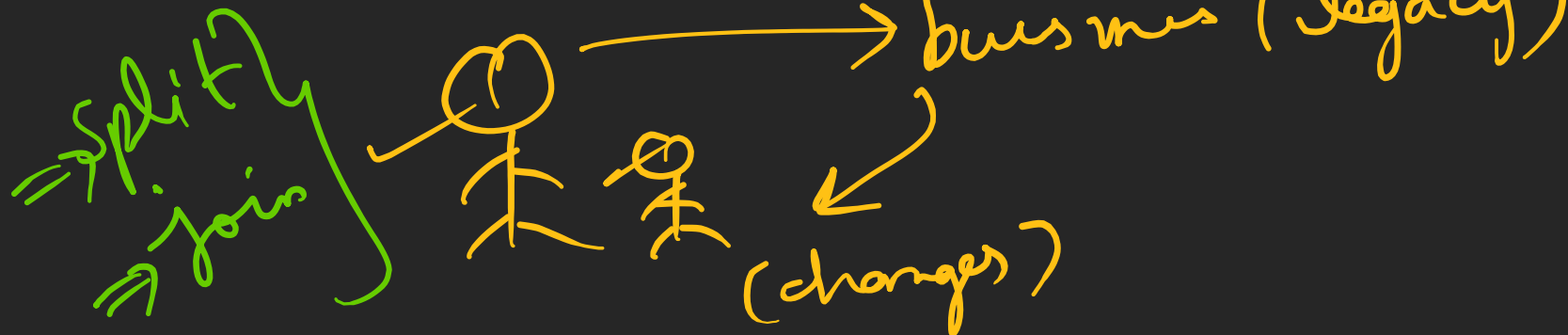
```
arrNew = arrOld;
```

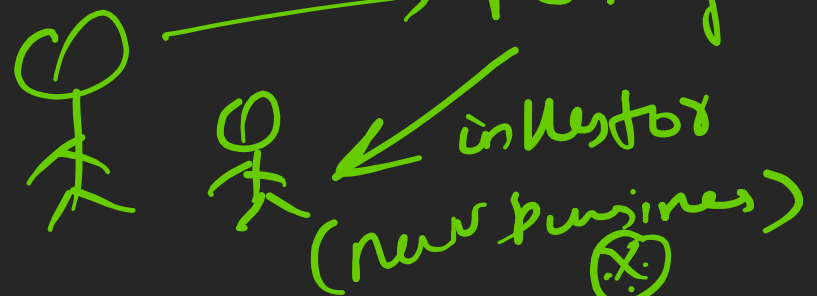
[1, 2, 3]

New memory (values)

```
arrNew = [...arrOld];
```

clone / shallow copy!





```
let arr = [1,2,3];   "normal";
let resultArray = copyArr("spread", arr);
function copyArr(type, value) {

    if(type == "spread") {
        let newArr = [...value];
        return newArr;
    }
    else {
        let newArr = value;
        return newArr;
    }
}
```

```
let spreadArray = copyArray("spread", arr);

spreadArray[0] = 96;

console.log("Actual arr:", arr);

console.log("spread Array:", spreadArray);

let normalArray = copyArr("normal", arr); normalArray[0] = 25;

// con.log("norArr", normalArray);
```

```
Var arr = [1, 2, 3, 4, 5];

// 1+2+3+4+5 = ?
   3 +3
     6 +4
       10 +5 = (15) ?
```

```
function addition
  (num1, num2, num3)
     num4, N5, N6
{
  let num7 =
    num1 + num2 +
          num3;
  return num7;
}
```

```
Var num1 = 10;
Var num2 = 20;
Var num3 = 30;
```

```
Var num4 = 40;
Var num5 = 50;
Var num6 = 60;
```

```
Var result = addition (num1, num2, num3);
                      num4, num5, num6
```

function addition (...numbers) {

[ ]

3

i === 2

!==

const numbers = [1, 2, 3, 4, 5];

0 1 2 3 4

for (let i = 0; i < numbers.length; i++) {

let res = [ ];

if (i !== 2 || i !== 3)

{ res.push(numbers[i]);

3 return res;