**Topics**

* **Spread operator**
* **Rest Operator**
* **Function overloading**
* **Function overriding**
* **Destructring**

[**Function overloading**](https://www.geeksforgeeks.org/function-overloading-and-const-functions/) is a feature of object-oriented programming where two or more functions can have the same name but different parameters. When a function name is overloaded with different jobs it is called Function Overloading. In Function Overloading “Function” name should be the same and the arguments should be different.

Unlike other programming languages, JavaScript **Does not support Function Overloading**.

However, in JavaScript, **if there are multiple functions with the same name, the function that is defined at the last gets executed**.

**Overriding occurs when two methods have the same method name and parameters**. One of the methods is in the parent class, and the other is in the child class.

**Rest Operator:**

The **rest parameter** syntax collect/gather a list of individual elements as an array.

The **rest parameter** syntax allows a function to accept/accept an indefinite number of arguments as an array.

**Spread Operator:**

spread syntax expands iterables into individual elements.

The spread (...) syntax allows an iterables, such as an array or string, to be expanded in places where zero or more arguments (for function calls) or elements (for array literals) are expected. In an object literal, the spread syntax enumerates the properties of an object and adds the key-value pairs to the object being created.

<https://www.freecodecamp.org/news/javascript-rest-vs-spread-operators/>

**shallow and deep copy in spread operator:**

**O.S**

when we use spread operator in destruction of array object and function copy depends on the type of value inside of array or object which we are spreading if inside of array non-primitive value and it will not generate reference means **SHOLLOW COPY** and if inside the array value is primitive then it will not generate reference between them means change in original will not effect on new one.

<https://www.freecodecamp.org/news/javascript-rest-vs-spread-operators/#info-3-beware-of-how-spread-works-when-used-on-objects-containing-non-primitives->

<https://dev.to/oluwatobiss/spread-operator-how-spread-works-in-javascript-4fdn#comment-node-767546>

**How to Complete Deep copy by the use of …spread Operator:**

[**https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a**](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a)

**Best Explanation of deep copy I have learned with spread operator**

## 5 Ways to Deep Copy Objects in JavaScript

In JavaScript, we can perform a copy on objects using the following methods:

|  |  |  |
| --- | --- | --- |
| **Method** | **Pros** | **Cons** |
| [shallow copy with =](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a#shallow) | clear and direct, the default | only shallow copies objects |
| [JSON.stringify() and JSON.parse()](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a#json) | deep copies nested objects | doesn't copy functions |
| [Object.assign()](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a#object-assign) | copies the immediate members of an object—including functions | doesn't deep copy nested objects |
| [the ... spread operator](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a#spread) | simple syntax, the preferred way to copy an object | doesn't deep copy nested objects |
| [Lodash cloneDeep()](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a#lodash) | clones nested objects including functions | adds an external dependency to your project |

These methods all have their pros and cons. Let's take a closer look at each of them.

[**https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a**](https://code.tutsplus.com/the-best-way-to-deep-copy-an-object-in-javascript--cms-39655a)

**Destructring:**

Destructuring makes it easy to extract only what is needed.

O.S: **to extract only those items from bulk that you need for the current perspective.**