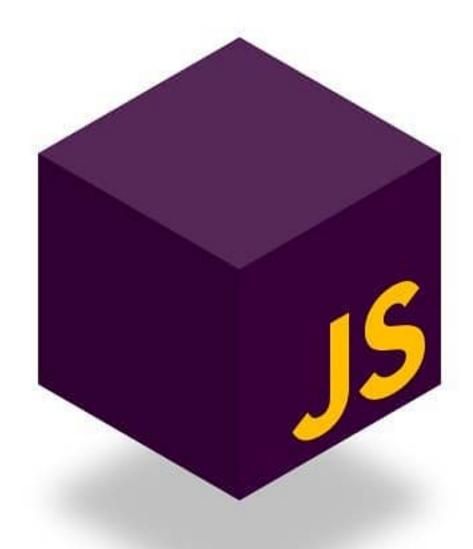THE COMPLETE GUIDE

# SPREAD OPERATOR
## USE CASES

JS

# STRING TO ARRAY WITH SPREAD

```js
JS
const string = 'ADITYA';

const array = [...string];

array;
// [ 'A', 'D', 'I', 'T', 'Y', 'A' ]


//Three Magic dots 💪
```

# SPREAD FOR MERGING ARRAYS

```js
const array1 = [1, 2, 3];

const array2 = [4, 5, 6];

//Let's merge these arrays

const mergedArray = [
  ...array1,
  ...array2
];

mergedArray;
// [ 1, 2, 3, 4, 5, 6 ]  👆 ✅ Nice flattened array
```

← *YES, IT'S THAT EASY :)*

# CLONING ARRAY USING SPREAD

```js
// Without using Spread

const original = ['zero', 'one'];
const newArray = original;

original; // ['zero', 'one']
newArray; // ['zero', 'one']

//Our original array is affected
```

```js
// With Spread

const original = ['zero', 'one'];
const newArray = [...original];

original; // ['zero', 'one']
newArray; // ['zero', 'one']

//✅ Original array is NOT affected
```

# GET MIN & MAX VALUE FROM AN ARRAY IN JAVASCRIPT

**Using apply**

```
//BEFORE ES6


var nums = [1, 2, 3]
Math.min.apply(Math, nums)   // -> 1
Math.max.apply(Math, nums)   // -> 3
```

**Using spread**

```
//With ES6/ES2016 destructuring
assignment it becomes easier


const nums = [1, 2, 3]
Math.min(...nums)    // -> 1
Math.max(...nums)    // -> 3
```

# SET TO ARRAY

```js
//Creating a new Set Object

const set = new Set([1, 2, 3]);

set;
// Set(3) {1, 2, 3}
```

```js
//Converting SET into Array

const array = [...set];

array;
// [1, 2, 3]
```

# NODELIST TO ARRAY

```js
// create a `NodeList` object

const divs = document.querySelectorAll('div');


// convert `NodeList` to an array

const divsArr = [...divs];
```

{...}

# DEFINITION FROM GOOGLE

A JavaScript **spread operator** lets you access the contents of an iterable object. The spread operator is a set of three dots (an ellipsis) followed by the name of the iterable you want to access.

**USE CASES TO UNDERSTAND IT BETTER**
←