

Operador Spread

Numa situação onde você precisa agrupar dados em arrays ou objetos, você talvez teria essa situação.

```
let myArray = [1,2,3,4,5];
let mySecondArray = myArray;
mySecondArray.push(6,7,8,9,10);
```

Com o operador Spread você consegue aumentar a legibilidade e deixar o código melhor:

```
let myArray = [1,2,3,4,5];
let mySecondArray = [...myArray, 6,7,8,9,10];
```

A grosso modo o Spread irá pegar todos os dados de "myArray" e por baixo dos panos já irá "pushar" para o novo array (mySecondArray).

Spread com Objetos (mais utilizado no dia a dia)

```
const addMoreInfoToUser = (info) => {
  const newInfo = {
    ...info,
    birthOfDate: new Date(),
    children: false
  }
};

let user = {
  id: 323,
  name: 'Gabriel Valin',
  address: {
    street: 'Brooklyn 99',
    number: 311,
    district: 'Brooklyn'
  }
};

addMoreInfoToUser(user);
```

Operador Rest

Bem menos utilizado que o Spread, porém muito legal e importante a utilização do mesmo.

```
const createUser = (numbers) => {
  console.log(numbers)
}

// Supondo que um usuário mande isso chamando minha função:
createUser('Gabriel', 21, 'Brooklyn', 'wtf');

// E outro mande assim:
```

```
createUser('Gabriel',1,1,1,1,1,1,1,1,1,1,1,1,1);
```

// Já que você não sabe quantos parâmetros o usuário irá mandar, você pode simplesmente utilizar o rest alterando o parâmetro da sua função desta maneira:

```
const createUser = (...numbers) => {  
  console.log(numbers)  
}
```