

Parâmetros



DIGITAL
INNOVATION
ONE

Valores padrão

Pré-ES2015:

```
function exponencial(array, num) {  
  if (num === undefined) {  
    num = 1;  
  }  
  
  const result = [];  
  
  for(let i = 0; i < array.length; i++) {  
    result.push(array[i] ** num);  
  }  
  
  return result;  
}  
  
exponencial([1, 2, 3, 4])  
// [1, 2, 3, 4]  
  
exponencial([1, 2, 3, 4], 4)  
// [1, 8, 27, 64]
```

Pós-ES2015:

```
function exponencial(array, num = 1) {  
  const result = [];  
  
  for(let i = 0; i < array.length; i++) {  
    result.push(array[i] ** num);  
  }  
  
  return result;  
}  
  
exponencial([1, 2, 3, 4])  
// [1, 2, 3, 4]  
  
exponencial([1, 2, 3, 4], 4)  
// [1, 8, 27, 64]
```



DIGITAL
INNOVATION
ONE

Objeto “arguments”

```
function findMax() {  
  let max = -Infinity;  
  
  for(let i = 0; i < arguments.length; i++) {  
    if (arguments[i] > max) {  
      max = arguments[i];  
    }  
  }  
  
  return max;  
}
```

```
> findMax(1, 2, 3, 6, 90, 1)  
< 90
```

Um array com todos os parâmetros passados quando a função foi invocada.



DIGITAL
INNOVATION
ONE

Objeto “arguments”

```
function showArgs() {  
  return arguments;  
}
```

```
> showArgs(1, 2, [2,3,4], "string")  
< Arguments(4) [1, 2, Array(3), "string", callee: f, Symbol(Symbol.iterator): f]  
  0: 1  
  1: 2  
  2: (3) [2, 3, 4]  
  3: "string"  
  callee: f showArgs()  
  length: 4  
  Symbol(Symbol.iterator): f values()  
  __proto__: Object
```



DIGITAL
INNOVATION
ONE

Arrays

Spread: uma forma de lidar separadamente com elementos

```
function sum(x, y, z) {  
  return x + y + z;  
}  
  
const numbers = [1, 2, 3];  
  
console.log(sum(...numbers));
```

O que era parte de um array se torna um elemento independente.



Arrays

Rest: combina os argumentos em um array

```
function confereTamanho(...args) {  
  console.log(args.length)  
}  
  
confereTamanho() // 0  
confereTamanho(1, 2) // 2  
confereTamanho(3, 4, 5) // 3
```

O que era um elemento independente se torna parte de um array.



Objetos

Object Destructuring

```
const user = {  
  id: 42,  
  displayName: 'jdoe',  
  fullName: {  
    firstName: 'John',  
    lastName: 'Doe'  
  }  
};  
  
function userId({id}) {  
  return id;  
}  
  
function getFullName({fullName: {firstName: first, lastName: last}}) {  
  return `${first} ${last}`;  
}  
  
userId(user)  
// 42  
  
getFullName(user)  
// John Doe
```

Entre chaves {}, podemos filtrar apenas os dados que nos interessam em um objeto.