Nicoly Almeida

JAVA SCRIPT

Manipular funções

Declaração de função - Forma 1

```
function addition(param1, param2) {
    return param1 + param2;
}

console.log(addition(1)); //=> NaN
console.log(addition(1, 2)); //=> 3
console.log(addition(1, 2, 3)); //=> 3
```

Declaração de função - Forma 2

```
const addition = function (param1, param2) {
    return param1 + param2;
};

console.log(addition(1, 2)); //=> 3
```

Declaração de função - Forma 3

```
const addition = (param1, param2) => {
    return param1 + param2;
};

console.log(addition(1, 2)); //=> 3
```

Declaração de função - Forma 4

```
const addition = (param1, param2) => param1 + param2;

console.log(addition(1, 2)); //=> 3
```

Case Sensitive

```
function addition(param1, param2) {
    return param1 + param2;
}

function Addition(param) {
    return param + 1;
}

console.log(Addition(1)); //=> 2
console.log(Addition(1, 2)); //=> 2
```

Redefinição de Função

```
function addition(param1, param2) {
    return param1 + param2;
}

function addition(param) {
    return param + 1;
}

console.log(addition(1)); //=> 2
console.log(addition(1, 2)); //=> 2
```

Parâmetro Default

```
function addition(param1, param2 = 0) {
    return param1 + param2;
}

console.log(addition(1)); //=> 1
console.log(addition(1, 2)); //=> 3
```

Parâmetro Rest

```
function addition (...params) {
    let summation = 0;
    for (let value of params) {
        summation += value;
}
    return summation;
}

console.log(addition(1)); //=> 1 ([1])
console.log(addition(1, 1)); //=> 2 ([1, 1])
console.log(addition(1, 1, 1, 1)); //=> 4 ([1, 1, 1, 1])
```

Callback

```
function calc(param1, param2, callback) {
    return callback(param1, param2);
}

console.log(calc(2, 1, (x, y) => x + y)); //=> 3
console.log(calc(2, 1, (x, y) => x - y)); //=> 1
console.log(calc(2, 1, (x, y) => x * y)); //=> 2
console.log(calc(2, 1, (x, y) => x / y)); //=> 2
console.log(calc(2, 1, (x, y) => x ** y)); //=> 2
```

Obrigada!

nicolyejady@gmail.com