Funções

As funções dentro do JavaScript são uma das coisas mais importantes das quais você deve aprender, elas são a base de muitas coisas que veremos daqui pra frente, portanto as estude com bastante carinho e atenção.

Afinal de contas, o que é uma função?

Segundo a própria documentação:

> "Funções são blocos de construção fundamentais em JavaScript. Uma função é um procedimento de JavaScript - um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la."

> "Assim como o programa em si, uma função é composta por uma sequência de instruções chamada corpo da função. Valores podem ser passados para uma função e ela vai retornar um valor."

E é exatamente como está dito acima, uma função é um conjunto de código que executa uma tarefa, e você pode reaproveitar isso onde quiser dentro do seu código, por exemplo se precisar somar números mais de uma vez você cria uma função para tal e usa essa função quando precisar somar números.

function declaration

Abaixo temos a criação de uma função dentro do JavaScript, existem algumas formas diferentes de criarmos funções dentro do JavaScript, e vamos começar pela mais clássica que chamamos de **function declaration:**

```
function minhaFuncao(parametro) {
  return alguma coisa
}
```

Exemplo com a soma de números:

```
function somaNumeros(num1, num2) {
  return num1 + num2;
}
```

Agora como podemos chamar e usar essa função? É bem simples:

```
somaNumeros(5, 10);
// isso retorna 15, mas poderia ser outro a depender dos valores passados nos parametros
```

Use o **return** para quando desejar que a função te retorne algum valor que será usado em seu programa, por exemplo:

```
function somaNumeros(num1, num2) {
    return num1 + num2;
}

function usaOutraFuncao(nome) {
    console.log(
`Olá ${nome}, o retorno da outra função é a seguinte: ${somaNumeros(3, 3)}`);
}

usaOutraFuncao('Lucas');
```

Observe mais um exemplo de funções com parâmetros :

```
function funcaoComVariosParametros(nome, num1, num2) {
    console.log(`Olá ${nome}, a soma de ${num1} + ${num2} é ${num1 + num2}.`);
}

funcaoComVariosParametros('Lucas', 2, 3);
// -> Olá Lucas, a soma de 2 + 3 é 5.
```

Dados os exemplos acima vamos entender melhor como uma função funciona, o que são parâmetros, retorno, e tudo que precisamos saber sobre ela.

A function declaration como vimos é a maneira mais "clássica" e mais vista em projetos que utilizam JavaScript puro, sua sintaxe é simples como a maioria das coisas no JavaScript, existe a palavra chave function, em seguida temos o nome dessa função que o criador da função escolhe esse nome, logo após o nome da função temos os (parênteses) onde incluímos os parâmetros dessa função se for necessário, e por último temos { as chaves } onde colocamos a lógica da função, o que ela irá fazer por nós, ou então usar o return com o que quisermos que ela nos retorne.

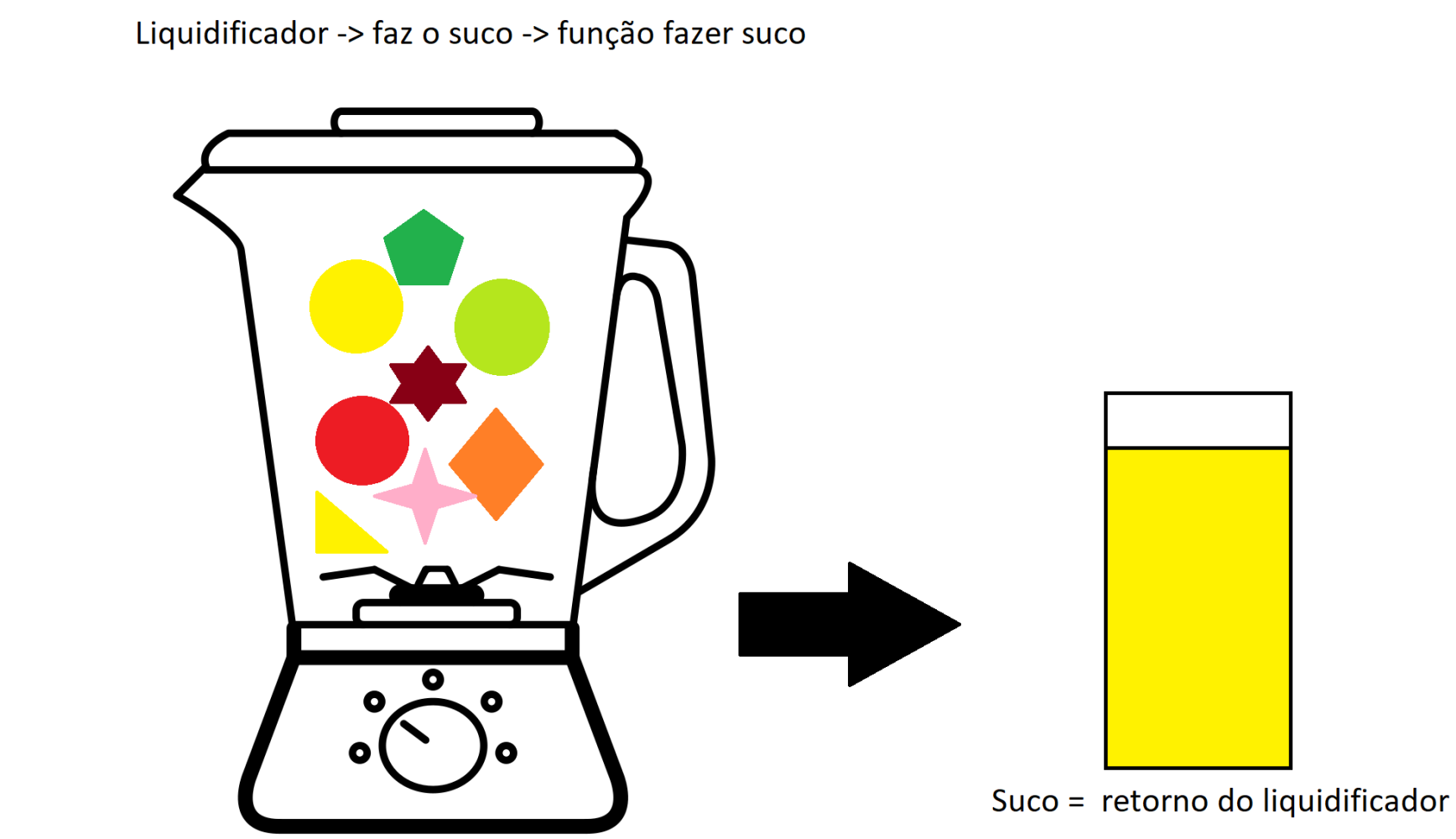
```
// função com parametro e return
function nomeDaFuncao1(parametros) {
    return alguma coisa;
}

// função com parametro sem return
function nomeDaFuncao2(parametros) {
    // faz alguma coisa e retorna implicitamente
}

// função sem parametro e sem return
function nomeDaFuncao3() {
    // faz alguma coisa e retorna implicitamente
}

// função sem parametro e com return
function nomeDaFuncao4() {
    return alguma coisa;
}
```

Agora precisamos entender melhor sobre os parâmetros e o return, para isso vamos usar como exemplo uma função que possui esses dois itens, podemos comparar essa função com um liquidificador:



De uma forma simples e apenas didática, no JavaScript o exemplo acima seria algo como abaixo:

```
function liquidificador(fruta1, fruta2) {
    const suco = fruta1 + fruta2;

    return suco;
}

liquidificador(laranja, acerola);
```

Portanto entenda que os parâmetros são usados como variáveis dentro de uma função, onde dizemos o que será feito com esses parâmetros dentro das chaves (escopo) dessa função, podemos dar qualquer nome aos parâmetros desde que faça sentido, e na hora de chamar a função passamos valores reais para que a função faça o que foi feita para fazer. A função é como uma instrução: "Faça isso com essas coisas que darei em suas mãos", e quando você chama essa função e dá as coisas na mão dela ela faz o que foi feita para fazer e te retorna algo (se você quiser isso).

Entenda a função como um sub programinha dentro do seu programa que faz determinada tarefa preestabelecida, e sempre que você quiser ela entra em ação e faz o que você programou ela para fazer.

Uma função pode também não ter um retorno, apenas executar alguma coisa sem executar nada, em breve você aprenderá coisas mais avançadas sobre a web, mas imagine um formulário de cadastro, ao terminar o preenchimento das informações o usuário clica em um botão "Salvar", por trás desse botão existe uma função por exemplo "salvarUsuario" que salva as informações do usuário em um banco de dados, apenas isso, sem retornar nada.

---

function expression

Outro tipo de função que temos para começar é a function expression, esse tipo de função se baseia em criar uma variável e atribuir uma função como valor dessa variável, o resto é bem parecido com o que vimos até agora:

```
const multiplica = function (n1, n2) {
    return n1 * n2;
}

const result = multiplica(2, 5);

console.log(result);
// -> 10
```

De maneira inicial você irá usar mais a function declaration em seus projetos, em breve você aprenderá sobre funções anônimas e arrow functions das quais irão te acompanhar mais por serem menos verbosas e mais rápidas de serem utilizadas.