Programação I
Javascript

Professor
Marcus Aurelius

JavaScript Funções

Características

- Ideia de modularização;
- Elemento básico do javascript;
- Executa uma ação;
- Bloco de código nomeado ou não;
- Deve ter um nome que deixe claro o que ela faz;
- Pode ou não receber valores de entrada (parâmetros), não sendo necessário definir o tipo desses parâmetros;
- É possível definir um valor padrão para os parâmetros;
- A função é executada retornando ou não valores (NaN, por exemplo) a depender do que deve ser feito;

Estrutura de uma função

- Nome da função;
- Parâmetros (entre parênteses e separados por vírgula);
- Bloco de comandos a serem executados;
- Retorno do resultado gerado pela função;
- Deve ter um nome que deixe claro o que ela faz;
- Pode ou não receber valores de entrada (parâmetros), não sendo necessário definir o tipo desses parâmetros;
- É possível definir um valor padrão para os parâmetros;
- A função é executada retornando ou não valores (NaN, por exemplo) a depender do que deve ser feito;

Funções - Estrutura Básica

Estrutura de uma função

Sintaxe:

```
function <nome>([<parâmetros>]) {
    <comandos>
    return [<valor retornado>]
}
```

Exemplo:

```
function media(valor1 = 0, valor2 = 0) {
    var valorMedia = 0
    valorMedia = ((valor1 + valor2) / 2)
    return valorMedia
}
```

Funções - Estrutura Básica

Estrutura de uma função

Exemplo 1:

```
prompt = require('prompt-sync')();
// cálculo da média entre dois valores
function media(valor1 = 0, valor2 = 0) {
    var valorMedia = 0
    valorMedia = ((valor1 + valor2) / 2)
    return valorMedia
}

let v1 = 0
let v2 = 0
let mediaCalculada = 0
console.log("Calculando a média entre dois valores")
console.log()
v1 = parseInt(prompt("Valor 1: "))
v2 = parseInt(prompt("Valor 2: "))
mediaCalculada =  media(v1, v2)
console.log(`Média dos valores ${v1} e ${v2}: ${mediaCalculada.toFixed(1)}`)
```

Funções - Estrutura Básica

Estrutura de uma função

Exemplo 2:

```
// programa para ler dois valores e exibor o maior valor dentre eles

// função para retornar o maior valor entre dois números
function maior(valor1 = 0, valor2 = 0) {
    let maiorValor = valor1;
    if (valor2 > valor1) {
        maiorValor = valor2;
    }
    return maiorValor;
}

// código principal
const prompt = require('prompt-sync')();
let v1 = 0
let v2 = 0
let maiorValor = 0
console.log("Buscando o maior valor entre dois números lidos")
console.log()
v1 = parseInt(prompt("Valor 1: "))
v2 = parseInt(prompt("Valor 2: "))
maiorValor =  maior(v1, v2)
console.log(`Maior valor entre ${v1} e ${v2}: ${maiorValor.toFixed(1)}`)
```

Estrutura de uma função

Exemplo 3:

```
// programa para calcular o fatorial de um número

// funcao para calcular o fatorial de um número
function fatorial(numero = 0) {
    let fat = 1;
    for(let contador = 1; contador <= numero; contador++) {
        fat *= contador
    }
    return fat
}

// código principal
const prompt = require('prompt-sync')();
let numero = 0
let fatorialCalculado = 0
console.log("Cálculo do fatorial de um número")
console.log()
numero = parseInt(prompt("Número: "))
fatorialCalculado =  fatorial(numero)
console.log(`O fatorial de ${numero} é: ${fatorialCalculado}`)
```

Funções - Estrutura Básica

Estrutura de uma função

Exemplo 4:

```
// programa para calcular a média entre dois valores

// importando funções de outros módulos
const prompt = require('prompt-sync')();
const funcoes = require('./funcoes')

let v1 = 0
let v2 = 0
let mediaCalculada = 0
console.log("Calculando a média entre dois valores")
console.log()
v1 = parseInt(prompt("Valor 1: "))
v2 = parseInt(prompt("Valor 2: "))
mediaCalculada =  funcoes.media(v1, v2)
console.log(`Média dos valores ${v1} e ${v2}: ${mediaCalculada.toFixed(1)}`)
```

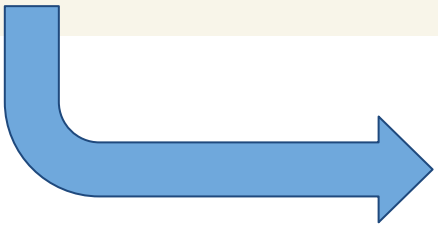
Funções - Estrutura Básica

Estrutura de uma função

Exemplo 5:

```
module.exports = {
// cálculo da média entre dois valores
media(valor1 = 0, valor2 = 0) {
    var valorMedia = 0;
    valorMedia = ((valor1 + valor2) / 2);
    return valorMedia;
},

// função para retornar o maior valor entre dois números
maior(valor1 = 0, valor2 = 0) {
    let maiorValor = valor1;
    if (valor2 > valor1) {
        maiorValor = valor2;
    }
    return maiorValor;
},
```



```
// funcao para calcular o fatorial de um número
fatorial(numero = 0) {
    let fat = 1;
    for(let contador = 1; contador <= numero; contador++)
{
        fat *= contador
    }
    return fat
}

}
```

Estrutura de uma função

Exemplo 6:

```
// código principal
const prompt = require('prompt-sync')();
const meuModulo = require('./funcoes')
let numero = 0
let fatorialCalculado = 0
console.log("Cálculo do fatorial de um número")
console.log()
numero = parseInt(prompt("Número: "))
fatorialCalculado =  meuModulo.fatorial(numero)
console.log(`O fatorial de ${numero} é: ${fatorialCalculado}`)
```

Funções - Estrutura Básica

Estrutura de uma função

Exemplo 7:

```
// identificação do maior valor entre dois números
const funcoes = require('./funcoes');
prompt = require('prompt-sync')();
let maiorValor = 0
let numero1 = 0
let numero2 = 0
console.log('Achar o meior valor dentre dois números')
console.log()
numero1 = parseInt(prompt('Informe o primeiro número: '))
numero2 = parseInt(prompt('Informe o segundo número: '))
maiorValor = funcoes.maior(numero1,numero2);
console.log(`O maior número é: ${maiorValor}`)
```

JavaScript Funções - Atribuída a uma variável

Estrutura de uma função atribuída a uma variável

Sintaxe:

```
<variável> = function(<parâmetro>,...,<parâmetro>) { <comandos> }
```

Exemplo 8:

```
// somando dois valores
// atribuindo a função a uma variável
const somar = function (a, b) { return a + b }  // funcão anônima
prompt = require('prompt-sync')();
let soma = 0
let numero1 = 0
let numero2 = 0
console.log('Soma de dois números')
console.log()
numero1 = parseInt(prompt('Informe o primeiro número: '))
numero2 = parseInt(prompt('Informe o segundo número: '))
// usando a função atribuída à variável
soma = somar(numero1, numero2);
console.log(`A soma é: ${soma}`)
```

Estrutura de uma *arrow function*

Sintaxe:

```
<variável> = (a + b) => { <comandos> }
```

Exemplo 9:

```
// verificando se um número é par
// atribuindo a arrow function a uma variável
const par = (valor) => {
    return (valor % 2) == 0
}
prompt = require('prompt-sync')();
let soma = 0
let numero = 0
console.log('Verificando se um número é par')
console.log()
numero = parseInt(prompt('Informe o número: '))
// usando a função atribuída à variável
if (par(numero)) {
    console.log("Número par.")
}
else {
    console.log("Número ímpar.")
}
```

Funções - função de retorno implícito

Estrutura de uma função de retorno implícito

Sintaxe:

```
<variável> = (<parâmetros>) => <comando>
```

Exemplo 10:

```
// Contando a quantidade de caracteres de um nome
// atribuindo a função de retorno implícito a uma variável
const tamanho = (nome) => nome.length
prompt = require('prompt-sync')();
let nome = null
console.log('Contando a quantidade de caracteres de um nome')
console.log()
nome = (prompt('Informe o seu nome: '))
// usando a função atribuída à variável
console.log("Quantidade de caracteres: "+ tamanho(nome))
```

Estrutura de uma função com sequência de valores como parâmetros

Exemplo 11:

```
// somando dois valores
// atribuindo a função a uma variável
// a variável 'a' recebe 'n' valores
const somar = function (...a) {
    let soma = 0
    for (let contador = 0; contador < a.length; contador++) {
        soma += a[contador]
    }
    return soma
 }

prompt = require('prompt-sync')();
console.log('Soma de números')
console.log()
// usando a função atribuída à variável
soma = somar(5,91,10,25,79,83);
console.log(`A soma é: ${soma}`)
```