



Enxurrada de Bits

Funções

Funções

Uma função JavaScript é um bloco de código projetado para executar uma tarefa específica e é executada quando "algo" o chama.

- Uma função é definida com a function, seguido por um nome, seguido de parênteses ()
- Os parênteses podem incluir nomes de parâmetros separados por vírgulas
- O código a ser executado pela função é colocado dentro de chaves: {}

Obs.: Uma função deve, sempre, ser implementada no começo do código.

Sintaxe:

```
function minhaFuncao (parametro1, parametro2, parametro3) {  
    //código a ser executado  
}
```

Retorno da função

- Quando o JavaScript atinge um `return`, a função irá parar a execução e voltar para executar o código após a instrução de chamada. Sendo que o valor de retorno é devolvido de volta para o chamador.

Exemplo:

Calcule o produto de dois números e retorne o resultado

```
function calcularProduto(a, b) {  
  return a * b;  
}  
  
const x = calcularProduto(4, 3); // x = 12
```

Reutilização de código

- Você pode usar o mesmo código muitas vezes com parâmetros diferentes para produzir resultados diferentes.

Exemplo:

Converter Fahrenheit para Celsius

```
function celsius(fahrenheit) {  
  return (5/9) * (fahrenheit - 32);  
}
```

```
const x = celsius(32); // x = 0°C  
const y = celsius(86); // y = 30°C
```

Funções como variáveis

- Funções podem ser utilizadas da mesma forma que se usa variáveis

Exemplo:

Em vez de usar uma variável para armazenar o valor de retorno de uma função

```
const x = celsius(77);  
const texto = "A temperatura é " + x + " Celsius";
```

Use a função diretamente, como um valor variável:

```
const texto = "A temperatura é " + celsius(77) + " Celsius";
```

Variáveis locais

- Variáveis declaradas dentro de uma função JavaScript são locais para a função, ou seja, só podem ser acessadas dentro da função
- As variáveis locais são criadas quando a função começa e são excluídas quando a função é concluída

Exemplo:

```
// o código aqui NÃO PODE usar carro

function minhaFuncao() {
  const carro = "Volvo";
  // o código aqui PODE usar carro
}

// o código aqui NÃO PODE usar carro
```

Funções pré-definidas

- JavaScript tem várias funções pré-definidas
- Para conhecermos todas elas, precisamos estudar a documentação

Exemplos de funções pré-definidas:

- `parseInt(texto)`: Recebe uma string e retorna um número inteiro
- `parseFloat(numero/texto)`: Recebe uma string ou número e retorna um float

Referências

Material da disciplina de Programação Web do prof. Flávio Coutinho:

<https://fegemo.github.io/cefet-web/classes/js1/#39>

Documentação da Mozilla Developer Network:

<https://developer.mozilla.org/pt-BR/docs/Glossario/Fun%C3%A7%C3%A3o>

Documentação da W3Schools:

https://www.w3schools.com/js/js_functions.asp

Leiam

Funções recursivas em JavaScript:

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions#Recursion>

Dúvidas?

Colocando em Prática

- Implemente uma função que recebe 2 números inteiros como parâmetro e retorna a multiplicação desses números.
- Implemente uma função que calcula e retorna a potência de um número. A função deve receber como parâmetro o número (base) e o expoente.
 - Dica: utilize a função `Math.pow(base, expoente)`
- Implemente novamente todas as atividades das duas últimas aulas (Aula 07 e Aula 08), mas dessa vez utilizando funções. Basta copiar o código que você já tem e adaptar para que funcionem como funções.