

função:

em um programa determinadas tarefas ou declarações se repetem por vezes para evitar ou otimizar o código, essas tarefas ou declarações podem ser agrupadas em blocos esses blocos podem ser chamados, quando necessário, para realizar aquelas tarefas a serem repetidas

esses blocos são denominados - função - logo,

funções são ações executadas assim que são chamadas ou invocadas ou quando ocorre algum evento

podem receber parâmetros e retornar resultados

vantagens: proteção dada a const em JS (alerta no caso de repetição ...), sintaxe mais curta

função:

```
<script>
function ola() {
    alert("Olá Mundo! ")
}
ola()

const ola2 = () => {
    alert("Olá Mundo! ")
}
ola2()
</script>
```


};

função:

```
const prompt = require("prompt-sync")();
const situacao = (nota, media) => {
    if(nota >= media) {
        console.log(" aprovado");
    } else {
        console.log(" reprovado");
    }
}
const prova1 = Number(prompt("qual a nota: ");
    situacao(prova1, 7.0) // chama função situacao
passando dois parâmetros;
```

função:

```
function multiplica(num1, num2) {
    return num1 * num 2
}
console.log("chamando a função muliplica e o resultado: ", multiplica(3, 8));
```

a função é chamada no console.log, os parâmetros 3 e 8 são passados para a função multiplica que multiplica 3 * 8 e retorna o valor 24 para a chamada mostra na tela o valor 24

função:

função:

nesse caso, a chamada da função não enviou parâmetros apresentou o resultado internamente

```
.vscode \rangle aula14-M \rangle Js function01.js \rangle ...
               // função sem passagem de parâmetros
               function somar() {
         10
                   let num1 = 3;
         11
                   let num2 = 12;
         12
                   let soma = num1 + num2;
                   console.log(`a soma de ${num1} e ${num2} é ${soma}`)
         13
         14
               somar();
         15
2
         16
                                           ENTRADA E SAÍDA
        PROBLEMS
                     OUTPUT
                               TERMINAL
                                                             DEBUG CONSOLE
                                                                                PORTS
         /usr/local/bin/node ./.vscode/aula14-M/function01.js
         a soma de 3 e 12 é 15
```

função:

nesse caso, as variáveis foram declaradas fora da função essas variáveis podem ser globais, válidas para outras quaisquer funções

```
.vscode > aula14-M > JS function01.js > ...
              // função sem passagem de parâmetros2
        21
              let num1 = 3;
        22
              let num2 = 12;
         23
              let soma;
              function somar() {
        24
        25
                   soma = num1 + num2;
        26
                   console.log(`a soma de ${num1} e ${num2} é ${soma}`)
        27
        28
              somar();
2
                                         ENTRADA E SAÍDA
        PROBLEMS
                    OUTPUT
                              TERMINAL
                                                           DEBUG CONSOLE
                                                                            PORTS
         /usr/local/bin/node ./.vscode/aula14-M/function01.js
         a soma de 3 e 12 é 15
```

função:

as variáveis num1 e num2 são variáveis globais e podem ser utilizadas pelas outras funções ...

```
.vscode > aula14-M > JS function01.js > ...
              // função sem passagem de parâmetros2
        21
              let num1 = 3;
        22
              let num2 = 12;
              function somar() {
        23
        24
                  soma = num1 + num2;
                  console.log(`a soma de ${num1} e ${num2} é ${soma}`)
              function multiplicar() {
29
                  let multi = num1 * num2;
                  console.log(`a multiplicação de ${num1} e ${num2} é ${multi}`)
31
        32
        33
              function dividir() {
                  let div = num1 / num2;
                  console.log(`a divisão de ${num1} e ${num2} é ${div}`)
        37
              somar();
              multiplicar();
              dividir();
        41
        42
        PROBLEMS
                   OUTPUT
                             TERMINAL
                                         ENTRADA E SAÍDA
                                                          DEBUG CONSOLE
                                                                           PORTS
         /usr/local/bin/node ./.vscode/aula14-M/function01.js
         a soma de 3 e 12 é 15
         a multiplicação de 3 e 12 é 36
         a divisão de 3 e 12 é 0.25
```

função exemplo:

chamando uma função dentro de uma outra função

```
.vscode > aula14-M > JS functionrecusiva.js > ...
              // função recursiva que calcula o fatorial
              function fatorial(n) {
                  let fat = 1
                  for(let i = n; i > 1; i--) {
                      fat*=i
                  return fat
              console.log("fatorial de = ", fatorial(5))
// função que calcual o fatorial de um número resursivamente
        11
12
        13
              function fatorialrec(r) {
                  if (r == 1) {
        15
                      return 1
                  } else {
W)
                      return r * fatorialrec(r-1)
              console.log("fatorialrec = ", fatorialrec(7))
                                        ENTRADA E SAÍDA
       PROBLEMS
                   OUTPUT
                             TERMINAL
                                                         DEBUG CONSOLE
                                                                          PORTS
         /usr/local/bin/node ./.vscode/aula14-M/functionrecusiva.js
         fatorial de = 120
         fatorialrec = 5040
```

função exercício:

construir uma função que:

receba dois parâmetros, num1 e num2 e calcule as funções da calculadora também a função de calcular mensalidade

capital investido (inicial)

i - juro mensal

n numero de meses

resposta: mensalidade em yy meses

formula da calculo usando juros compostos:

$$ParcelaMensal = \frac{Capital * i}{1 - (1 + i)^{-n}}$$
 (Math.pow(base, potencia))

reescrever a função sem passagem de parâmetros usar entrada via prompt no JavaScript

função: exemplo de uma função cujo parâmetro é um vetor:

```
DOM > Function-Aula16 > Js functio_vet.js > ...
       let Vetor = [1, 2, 3, 4, 5];
       function tratavet(v) {
          v.slice(1, 3);
          return v;
      // tratavet(Vetor);
       console.log(tratavet(Vetor));
                                  ENTRADA E SAÍDA
PROBLEMS
            OUTPUT
                      TERMINAL
                                                   DEBUG CONSOLE
 /usr/local/bin/node ./DOM/Function-Aula16/functio_vet.js
> (5) [1, 2, 3, 4, 5]
```

função

exercício:

dado um vetor Numeros = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55] construa uma função que receba o vetor Numeros e separe os elementos pares construa uma função que receba o vetor Numeros e separe os elementos ímpares construa uma função que receba o vetor Números e acrescente os valores 55< fib < 600

construa funções, pois muitas podem ser blocos repetidos!!!!

...

função:

exercício

escrever uma função que, tendo um vetor C como variável global e, como parâmetros, os possíveis parâmetros do slice(r, s), retorne o vetor alterado!

lembrar que tem slices que têm um único parâmetro de entrada

os parâmetros de entrada, são os parâmetros do xxxx.slice(r, s)

tome como exemplo, os exercícios anteriores e utilize apenas o material de javascript, visto nessa disciplina!

vetor global:

C = ["cachorra", "gata", "papagaio", "pardal", "bem-te-vi", "búfalo", "sabiá", "boi", "pato"] em grupo de 2 alunos, entrega até o final da aula,

Tecnologia em Análise e Desenvolvimento de Sistemas

Rever:

Dado o vetor C = ["cachorra", "gata", "papagaio", "pardal", "bem-te-vi", "búfalo", "sabiá", "boi", "pato"]

Efetue as operações usando, para manipulação dos valores, métodos do objeto array e a propriedade length.

- a) Criar um novo vetor inserindo duas aves de rapina no início do vetor original, mostrar o vetor na tela, mostrar a posição dos novos inseridos
- b) Criar um novo vetor copiando os elementos ou valores entre os índices 1 e 7, mostrar o vetor na tela
- c) Criar um novo vetor cujo conteúdo sejam as aves, mostrar o vetor na tela
- d) Criar um novo vetor cujo conteúdo sejam animais de quatro patas, mostrar o vetor na tela
- e) Criar um novo vetor que seja resultado da concatenação do vetor do item c), com os 3 elementos iniciais do vetor original, mostrar o vetor na tela
- f) Criar um novo vetor que tenha em seu conteúdo animais de quatro patas e aves, nessa sequência, mostrar o vetor na tela
- g) A partir do vetor inicial, criar um vetor C_Ordenado que contenha nessa sequência: aves, do vetor inicial, animais de quatro patas e as aves de rapina sempre usando apenas métodos do objeto array, vistos em sala

função:

bibliografia

Silva, M. S., JavaScript Essencial, Novatec, 2022

Alguns exercícios de livros de C

IEPSEN, E. F. Lógica de Programação e Algoritmos em JavaScript, 2.a ED, Novatec – 2022 site do ECMAScript

https://ecma-international.org/publications-and-standards/standards/ecma-262/

Haverbeke, M., Eloquent Javscript, 4th Ed. no starch press 2024 creative communs - https://eloquentjavascript.net

template string:

definição de template string ou template literals usada para fazer interpolação entre strings e valores

```
Js templatestring_ex.js \times
       Explorador (公眾E)
        Js templatestring_ex.js > ...
              const numero1 = 10;
              const numero2 = 5;
              const soma = numero1 + numero2;
              const fraseComTemplateString = `0 resultado de ${numero1} + ${numero2} é igual a ${soma}`;
              const fraseSemTemplateString = "0 resultado de " + numero1 + " + " + numero2 + " é igual a " + soma;
              console.log(fraseComTemplateString); console.log(fraseSemTemplateString);
        PROBLEMAS
                    SAÍDA
                             TERMINAL
                                        ENTRADA E SAÍDA
                                                         CONSOLE DE DEPURAÇÃO
                                                                                 PORTAS
Info: Start process (6:12:26 PM)
        O resultado de 10 + 5 é igual a 15
        0 resultado de 10 + 5 é igual a 15
        Info: End process (6:12:26 PM)
```