# Maneiras de declarar funções

* Clássica:

function falaOi(){

    console.log('oi')

}

falaOi()

* Function Expression

const falaOi = function(){

    console.log('oi')

}

falaOi()

* Arrow Function

const falaOi = () =>{

    console.log('oi')

}

falaOi()

# Parâmetros da função

Quando uma função é declarada com *function*, ainda que a função não receba parâmetros, podemos executá-la enviando parâmetros, que ficam armazenados em *arguments*

function parametro() {

    console.log(*arguments*)

}

parametro('ola','mundo','!')

-------------------------

CONSOLE: [Arguments] { '0': 'ola', '1': 'mundo', '2': '!' }

# Retorno da função

O argumento *return* retorna um valor e termina a execução da função.

//retorno de função

function multiplicador(*multiplicador*){

    return function(*n*){

        return *n* \* *multiplicador*

    }

}

const duplica = multiplicador(2) //duplica recebe uma função

                                 //que retornará número vezes multiplicador

//console.log

console.log(duplica(3)) //resultado "6"

# Escopo léxico

Nada mais é do que a função “saber” o que pode se acessar, através do seu escopo e dos escopos próximos.

//Escopo léxico

const num = 1

function contar(){

    console.log(num)

}

function usarContar(){

    const num = 2

    contar()

}

//console.log

usarContar() //resultado “1”