FUNÇÕES

AULA 01 - TIPOS DE FUNÇÃO:

**1.APRESENTAÇÃO DO CURSO:**

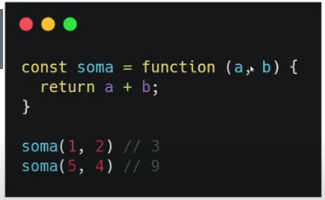
**2.ESTRUTURA E FUNÇÃO ANÔNIMA:**



\*variáveis criadas dentro de uma função só podem ser utilizadas nela mesma.

\*quando o return é invocado ele para a execução da função.

\*função anônima:



-é armazenada dentro de uma variável.

-função anônima só pode ser utilizada uma vez por não possuir nome.

**3.FUNÇÃO AUTOINVOCÁVEL E CALLBACKS:**

\*função autoinvocável: iffe



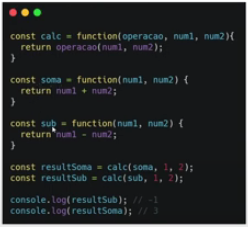
-uma função anônima entre parênteses, seguida por outro parêntesis, que representa sua chamada.

-podem receber parâmetros quando chamada.

-também pode ter seu resultado armazenado em uma constante.

\*callbacks:

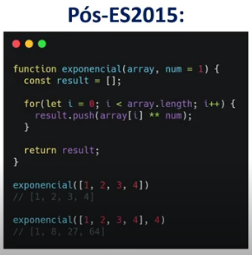
-é uma função passada como argumento para outra função.



AULA 02 - PARÂMETROS:

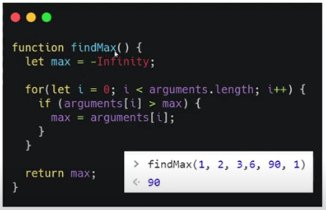
**1.VALORES PADRÃO E OBJETO ‘ARGUMENTS’:**

\*valores padrão:



\*objeto arguments:

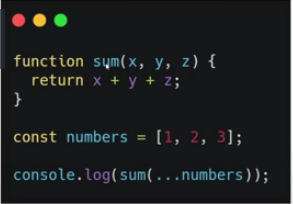
-é um array com todos os parâmetros passados quando a função foi invocada.



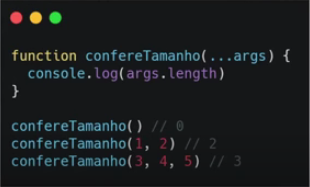
**2.ARRAYS E OBJETOS:**

\*arrays:

-spread: é uma forma de lidar separadamente com elementos. ‘...’. usado quando se chama a função.

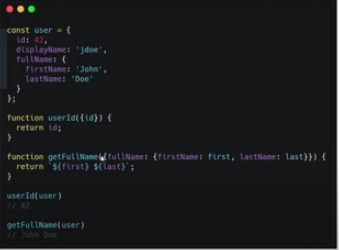


-rest: combina os argumentos em um array. o que era um elemento independente se torna um array. usado quando se declara a função.



\*objetos:

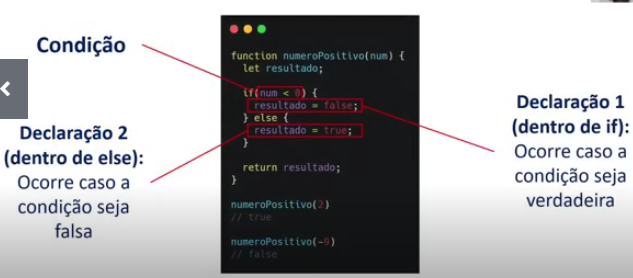
-object destructing: desestruturação de objeto. Entre chaves {} podemos filtrar os dados que nos interessam no objeto.



AULA 03 - LOOPS:

**1.IF/ELSE E SWITCH:**

\*if/else:



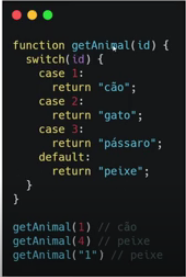
-condição, declaração 1 e declaração 2.

\*switch:

-serve para quando há muitas verificações.

-sempre compara tipo e valor.

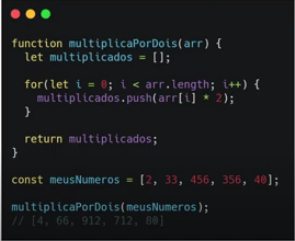
-sempre precisa de um valor ‘default’



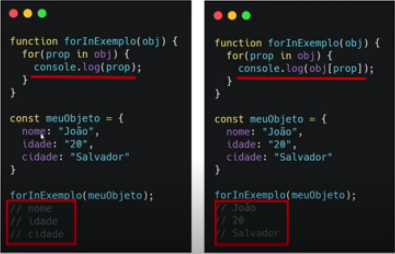
**2.FOR E WHILE:**

\*for:

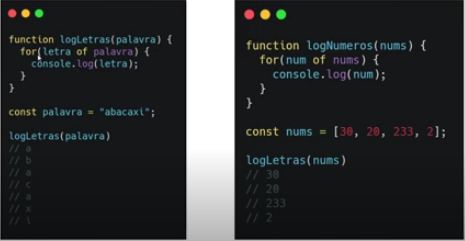
-é um loop dentro de elementos iteráveis (arrays e strings)



\*for…in: loop entre propriedades enumeráveis de um objeto.

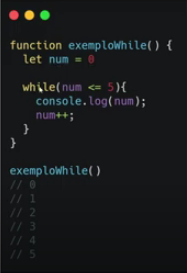


\*for…of: loop entre estruturas iteráveis (arrays, strings)



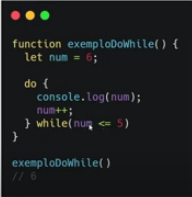
\*while:

-executa as funções até que a condição se torne falsa (enquanto)



\*do…while:

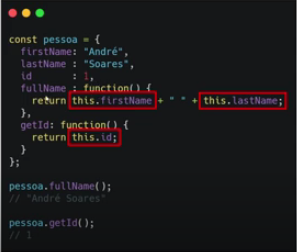
-executa a função até que se torne falsa. Porém a primeira execução sempre ocorre.



AULA 04 - THIS:

**1.O QUE É THIS:**

-a palavra reservada this é uma referência de contexto: this.xxx



-quando se usa uma função dentro de um objeto é chamado de método.

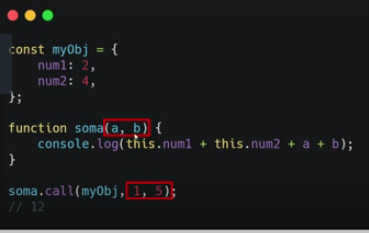


-quando se usa this dentro de um método, ele fará referência ao objeto pai do método.

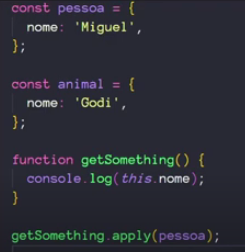
-no html refere-se ao evento.

**2.MANIPULANDO SEU VALOR (THIS):**

\*call (xxx): chama o especificado.



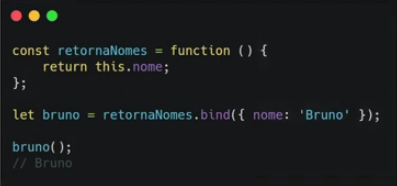
\*apply:



-os argumentos de um apply eles devem ser enviados dentro de um array

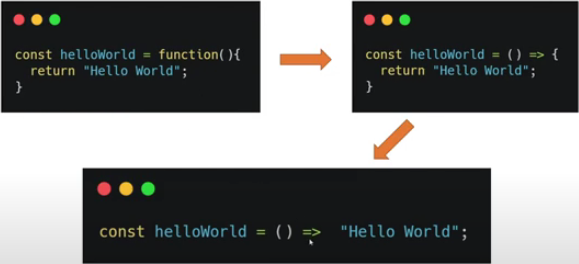
\*bind:

-clona a estrutura da função onde é chamada e aplica o valor do objeto passado como parâmetro.

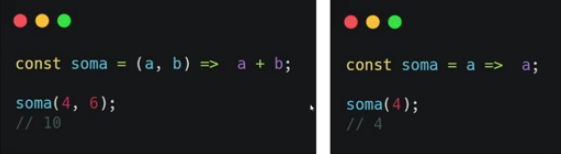


AULA 05 - ARROW FUNCTIONS:

**1.SINTAXE E OUTRAS RESTRIÇÕES:**



-o terceiro caso só é possível pois o return só tem uma linha (dispensa de chaves e return)



-caso exista apenas um parâmetro, os parêntesis são dispensáveis.

-arrow não faz hoisting(chamar a variável ou a função antes dela ser declarada).

\*outras restrições:

-this sempre será o objeto global, não é possível modificar seu valor em arrow.

-não existe o objeto arguments em arrow.

-construtor não pode ser utilizado em arrow.

**2.ATIVIDADE PRÁTICA:**

//comentários no exercício