MÓDULO F – AULA 15 – AVANÇANDO OS ESTUDOS EM JAVASCRIPT

VARIÁVEIS SIMPLES => SÓ CONSEGUEM ARMAZENAR UM VALOR POR VEZ.

EX: v1,v2,v3,v4...

VARIÁVEIS COMPOSTOS => SÃO CAPAZES DE ARMAZENAR VÁRIOS VALORES EM UMA MESMA ESTRUTURA.

EX: DENTRO DA VARIÁVEL a, TEREMOS VÁRIAS OUTRAS VARIÁVEIS

Vaga a => v1,v2,v3,v4.v5...

TEMOS A VARIÁVEI va QUE ARMAZENA VÁRIOS VALORES IDENTIFICADOS POR UMA CHAVE.

A VARIÁVEL A É UM array, QUE É COMPOSTPO POR VÁRIOS elementos QUE SÃO IDENTIFICADOS POR SEUS índices chave (QUE COMECAM COM O VALOR 0) E CADA ELEMENTO CONTEM UM contéudo/valor.

AS VARIÁVEIS COMPOSTOS SÃO REPRESENTADAS E SE DISTIGUEM PELA UTILIZAÇÃO DE CHAVES E NÃO POR COLCHETES, COMO COMUMENTE SE UTILIZA.

Ex:

Let num = [5,8,4]

let num = [5,8,4]

CASO QUEIRA ADICIONAR UM VALOR EXTRA DENTRO DA VARIÁVEL, BASTA FAZER DA SEGUINTE FORMA:

num[3] = 6

NESSE CASO, SABENDO QUE NA POSIÇÃO 3 DA VARIÁVEL COMPOSTA num NÃO EXISTE NENHUM ELEMENTO, É POSSÍVEL ADICIONAR UMA VARIÁVEL NA POSIÇÃO EM ESPECÍFICO. (LEMBRANDO QUE A CONTAGEM DOS ELEMENTOS COMEÇA COM O NÚMERO 0)

OUTRA FORMA REALIZAR ESSE PROCEDIMENTO É DA SEGUINTE FORMA:

num.push(7)

UTILIZANDO O COMANDO push APÓS A IDENTIFIACÇÃO DA VARIÁVEL QUE DESEJA, NO CASO A VARIÁVEL “num” É POSSÍVEL ADICIONAR O VALOR DENTRO DO PARENTESES NA PRIMEIRA VAGA DISPONIVEL APÓS A CONTAGEM DOS OUTROS ELEMENTOS PREVIAMENTE DECLARADOS. LOGO, TERIAMOS:

let num = [5,8,4]

num[3] = 6

num.push(7)

console.log(`Nosso vetor é o ${num}`)

COMANDO FUNCIONANDO:

Nosso vetor é o 5,8,4,6,7

É POSSÍVEL TAMBÉM SABER O COMPRIMENTO DO array, UTILIZANDO O SEGUINTE COMANDO

num.length

E TAMBÉM UTILIZANDO O SEGUINTE COMANDO PARA COLOCAR OS ELEMENTOS DENTRO DE UMA ORDEM NUMÉRICA CRESCENTE

num.sort()

EXEMPLO DA AULA

let num = [5,8,2,9,3]

num.push(1)

num.sort()

console.log(num)

console.log(`O vetor tem ${num.length} posições`)

console.log(`O primeiro valor do vetor é ${num[0]}`)

NO EXEMPLO, O PROFESSO DECLARA UMA VARIÁVEL COMPOSTA COM VÁRIOS ELEMENTOS, DEPOIS, UTILIZA O COMANDO num.push(1) PARA ADICIONAR O NÚMERO 1 NA PRIMEIRA VAGA DISPONÍVEL DENTRO DA VARIÁVEL COMPOSTA, EM SEGUIDA REALIZA A ORGANIZAÇÃO DOS ELEMENTOS EM ORDEM NUMÉRICA CRESCENTE (num.sort()). UTILIZA TAMBÉM O COMANDO “num.length” PARA CONTAR O NÚMERO DE POSIÇÕES DO ELEMENTO E UTILIZA O COMANDO “num[0]” PARA QUE NA EXECUÇÃO DO PROGRAMA APAREÇA O PRIMEIRO ELEMENTO DA VARIÁVEL COMPOSTA.

EXECUÇÃO DO PROGRAMA:

[ 1, 2, 3, 5, 8, 9 ]

O vetor tem 6 posições

O primeiro valor do vetor é 1

EXEMPLO DE AULA

let valores = [8,1,7,4,2,9]

for (let pos=0; pos < valores.length; pos++) {

    console.log(`A posição ${pos} tem o valor ${valores[pos]}`)

}

A posição 0 tem o valor 8

A posição 1 tem o valor 1

A posição 2 tem o valor 7

A posição 3 tem o valor 4

A posição 4 tem o valor 2

A posição 5 tem o valor 9

NO EXEMPLO ACIMA, UTILIZAMOS TODAS OS COMANDOS APRENDIDOS NESSA AULA SOMADOS A UTILIZAÇÃO DE UMA ESTRUTURA DE REPETIÇÃO (for), PARA OBTERMOS OS VALORES DA VARIÁVEL COMPOSTA POR POSIÇÃO.

EXEMPLO => SIMPLIFICANDO O EXEMPLO ANTERIOR

for(let pos in valores) {

    console.log(`A posição ${pos} tem o valor ${valores[pos]}`)

}

A posição 0 tem o valor 8

A posição 1 tem o valor 1

A posição 2 tem o valor 7

A posição 3 tem o valor 4

A posição 4 tem o valor 2

A posição 5 tem o valor 9

É POSSÍVEL OBTER O MESMO RESULTADO DO EXEMPLO ANTERIOR, UTILIZANDO ESSA FORMA SIMPLIFICADA DO COMANDO for.

BUSCANDO VALORES DENTRO DAS VARIÁVEIS COMPOSTAS.

Num.indexOf(8)

JAVASCRIPT, TEM O VALOR 7 DENTRO DA VARIÁVEL COMPOSTA?

let num = [5,8,2,9,3]

num.push(1)

num.sort()

console.log(num)

console.log(`O vetor tem ${num.length} posições`)

console.log(`O primeiro valor do vetor é ${num[0]}`)

let pos = num.indexOf(8)

console.log(`O valor 8 estpa na posição ${pos}`)

[ 1, 2, 3, 5, 8, 9 ]

O vetor tem 6 posições

O primeiro valor do vetor é 1

O valor 8 estpa na posição 4

O VALOR 8 ESTÁ NA POSIÇÃO 4. APÓS A ADIÇÃO DO NÚMERO 1 (num.push(1)) E DA REORGANIZAÇÃO DOS ELEMENTOS DENTRO DA VARIÁVEL num (num.sort()), É POSSÍVEL IDENTIFICAR ATRAVÉS DO COMANDO num.indexOf(8) A POSIÇÃO DO ELEMENTO 8.

EXEMPLO MODIFICANDO O ELEMENTO

let num = [5,8,2,9,3]

num.push(1)

num.sort()

console.log(num)

console.log(`O vetor tem ${num.length} posições`)

console.log(`O primeiro valor do vetor é ${num[0]}`)

let pos = num.indexOf(4)

if (pos == -1) {

    console.log("O valor não foi encontrado")

} else {

    console.log(`O valor 8 estpa na posição ${pos}`)

}

[ 1, 2, 3, 5, 8, 9 ]

O vetor tem 6 posições

O primeiro valor do vetor é 1

O valor não foi encontrado

PARA CASO DE SER COLOCADO DENTRO DO COMANDO indexOf UM VALOR QUE NÃO ESTEJA DENTRO DA VARIÁVEL COMPOSTA, AUTOMATICAMENTE A RESPOSTA DO JS É DE -1, LOGO, O PROFESSOR ADICIONOU UMA CONDIÇÃO UTILIZANDO if PARA QUE, QUANDO FOSSE IDENTIFICADO UM NÚMERO QUE NÃO ESTIVESSE DENTRO DA VARIÁVEL, O JS DARIA UMA RESPOSTA ESCRITA, INFORMANDO QUE O NÚMERO NÃO FOI ENCONTRADO. CASO CONTRÁRIO, DARIA A RESPOSTA DA POSIÇÃO DO ELEMENTO.

MÓDULO F – AULA 16 – FUNÇÕES

AS FUNÇÕES SÃO AÇÕES EXECUTADAS ASSIM QUE SÃO CHAMADAS OU EM DECORRÊNCIA DE ALGUM EVENTO.

UMA FUNÇÃO PODE RECEBER UM PARÂMETRO E RETORNAR UM RESULTADO.

EXEMPLO NÚMERO PAR OU IMPAR

function parimpar(n) {

    if (n%2==0) {

        return "Par!"

    } else {

        return "Impar!"

    }

}

let res = parimpar(4)

console.log(res)

O PROFESSOR UTILIZA O COMANDO function PARA RESOLVER O PROBLEMA SE O NÚMERO É PAR OU IMPAR, JUNTAMENTE COM UMA ESTRUTURA DE REPETIÇÃO.

NO FINAL, A CHAMADA let res = parimpar(4) DETERMINA QUAL NÚMERO VOCÊ DESEJA QUE SEJA REALIZADO O TESTE.

EXEMPLO SOMA

function soma(n1=0,n2=0) {

    return n1 + n2

}

console.log(soma(2,5))

/\*

console.log(soma(2,5))

o chamada acima substitui

let res = soma(n1,n2)

console.log(res)

\*/

O PROFESSOR UTILIZA O COMANDO function PARA SABER A SOMA DE DOIS NÚMEROS PRÉ DETERMIANDOS. DENTRO DA FUNÇÃO soma, GUANABARA INFORMA QUE, CASO NÃO SEJA ESPECÍFICADOS VALORES PARA N1 OU N2, O PROGRAMA AUTOMATICAMENTE OS DETERMINA COMO SENDO ZERO, COMO MOSTRADO ABAIXO.

function soma(n1=0,n2=0)

EXEMPLO: MULTIPLACAÇÃO UTILIZANDO UMA FUNÇÃO DENTRO DE UMA VARIÁVEL

let v = function (x) {

    return x\*2

}

console.log(v(5))

NO EXEMPLO, O PROFESSOR ALOCA UMA FUNÇÃO DENTRO DE UMA VARIÁVEL

EXEMPLO FATORIAL

function fatorial (n) {

    let fat = 1

    for (let c = n; c > 1; c--) {

        fat \*= c

    }

    return fat

}

console.log(fatorial(5))

EXEMPLO FATORIAL COM RECURSIVIDADE

// RECURSIVIDADE

function fatorial(n) {

    if (n == 1) {

        return 1

    } else {

        return n \* fatorial(n-1)

    }

}

console.log(fatorial(5))

MÓDULO F – AULA 17 – PRÓXIMOS PASSOS

RECAPTULANDO

1. FUNDAMENTOS DA LINGUAGEM
2. VARIÁVEIS E DADOS
3. OPERADORES
   1. ARITIMÉTICOS
   2. LÓGICOS
   3. RELACIONAIS
4. CONDIÇÕES
   1. IF SIMPLES
   2. IF COMPOSTOS
   3. SWITCH
5. REPETIÇÕES
   1. WHILE
   2. DO WHILE
   3. FOR
6. ARRAYS
7. FUNCTIONS
8. DOM

QUAIS OS PRÓXIMOS PASSOS?

1. FAÇA O CURSO DE HTML5 E CSS3 (TO FAZENDO)
2. ESTUDE MUITO SOBRE FUNCTION
3. OBJETOS
4. MODULARIZAÇÃO
5. EXPRESSÕES REGULARES (RegEx)
6. JSON
7. AJAX
8. NodeJS

NOÇÕES BÁSICAS SOBRE OBJETOS

INTRODUÇÃO

ARRAYS = VETOR (DEIXANDO AQUI SÓ PRA LEMBRAR)

OBJECTS SÃO VARIÁVEIS COMPOSTAS, ASSIM COMO OS ARRAYS

EXEMPLO DE OBJECT

let amigo = {nome: 'José', sexo: 'M', peso: 85.4, engordar(p=0){

    console.log('Engordou')

    this.peso += p

}}

amigo.engordar(2)

console.log(`${amigo.nome} pesa ${amigo.peso}kg`)

A VARIÁVEL AGORA POSSUI IDENTIFICAÇÕES ATRAVÉS DE NOMES ESPECÍFICOS

NO CASO, É POSSÍVEL ACESSAR UMA INFORMAÇÃO DA VARIÁVEL ATRAVÉS DA OBJETO (amigo) UTILIZANDO UMA PROPRIEDADE (KEY) INTERNA

Amigo.nome = josé

Amigo.sexo = M

Etc.