***Java Script***

*Conversão*

***String -> Número***

Number.parseInt(n) Inteiro

Number.parseFloat(n) Real

Number(n) Consegue identificar o tipo numérico

***Número -> String***

String(n)

n.toString()

// Formatando Strings

var s = ‘JavaScript’

‘Eu estou aprendendo s’ //não faz interpolação

‘Eu estou aprendendo’ + s // usa concatenação

`Eu estou aprendendo ${s}` // usa template string

s.length // quantos caracteres a string tem

s.toUpperCase() // tudo para MAIÚSCULAS

s.toLowerCase() // tudo para minúsculas

// Formatando Números

var n1 = 1543.5

n1.toFixed(2) // duas casas decimais

n1.toLocaleString(‘pt-BR’, {style: ‘currency’, currency:’BRL’}) // R$ 1.545,50

**OPERADORES**

***Aritméticos***

5 + 2 🡪 7

5 – 2 🡪 3

5 \* 2 🡪 10

5 / 2 🡪 2.5

5 % 2 🡪 1

5 \*\* 2 🡪 25

***ORDEM DE PRECEDÊNCIA***

( )

\*\*

\* / %

+ -

***Atribuição***

*Atribuição Simples*

Var a = 5 + 3 8

Var b = a % 5 3

Var c = 5 \* b \*\* 2 45

Var d = 10 – a / 2 6

Var e = 6 \* 2 / d 2

Var f = b % e + 4 / e 3

*Auto-atribuições*

Var n = 3 3

N = n + 4 7 n += 4

N = n – 5 2 n -= 5

N = n \* 4 8 n \*= 4

N = n / 2 4 n /= 2

N = n \*\* 2 16 n \*\*= 2

N = n % 5 1 n %= 5

*Incremento*

Var x = 5 5

X = x + 1 6 x += 1 x ++

X = x – 1 5 x -= 1 x --

***Relacionais***

5 > 2 🡪 true

7 < 4 🡪 false

8 >= 8 🡪 true

9 <= 7 🡪 false

5 == 5 🡪 true

4 != 4 🡪 false

// Exemplos

Preço <= 200. 50 // o preço é maior ou igual a 200.50?

Idade <18 // a idade é menor do que 18?

Curso == ‘JavaScript’ // o curso é JavaScript?

N1 != n2 // o primeiro número é diferente do segundo?

*Identidade*

5 == 5 🡪 true

5 == ‘5’ 🡪 true

5 === ‘5’ 🡪 false

5 === 5 🡪 true

***Lógicos***

! negação

&& conjunção

|| disjunção

Negação

! true 🡪 false

! false 🡪 true

Conjunção

True && true 🡪 true

True && false 🡪 false

False && true 🡪 false

False && false 🡪 false

Disjunção

True || true 🡪 true

True || false 🡪 true

False || true 🡪 true

False || false 🡪 false

***Ordem de execução:***

1 - ! Não

2 - && E

3 - || Ou

// Exemplos

Idade <= 15 && <= 17

// a idade está entre 15 e 17?

Estado == ’RJ’ || estado == ‘SP’

// o estado é RJ ou SP?

Salário < 1500 && sexo != ‘M’

// o salário é acima de 1500 e não é homem?

**PRECEDÊNCIA**

( ) \*\* / ...

> < >= ...

!

&&

||

***Ternário***

TESTE ? TRUE : FALSE

Média >= 7.0 ? “Aprovado” : “Reprovado”

**DOM**

*O que é?*

Document Object Model

**Árvore DOM**

window

location document history

html

head body

meta title h1 p p div

Strong

***Selecionando***

*Por Marca*

getElementsByTagName()

*Por ID*

getElementById()

*Por Nome*

getElementsByName()

*Por Classe*

getElementsByClassName()

*Por Seletor*

querySelector()

querySelectorAll()

OBS: Toda div é representada por uma “#” e toda classe é representada por um “.”

**Eventos DOM**

**<Div>**

*mouseenter*

*mousemove*

*mousedown*

*mouseup*

*click*

*mouseout*

***Funções***

*function* {

bloco

}

*function* ação() {

}

Sequências

var n = 3

n += 2

window.alert(n)

Condições

if (condição) {

true

} else {

false

}

// Tipos de Condição

Condição simples

if (condição) {

true

}

condição

if (condição) {

true

} else {

false

}

Condições Aninhadas

If (cond1) {

Bloco1

} else {

If (cond2) {

Bloco2

} else {

Bloco3

}

}

Condição Múltipla

switch (expressão) {

case valor 1:

break

case valor 2:

break

case valor 3:

break

default:

break

}

OBS: Dentro da estrutura ***SWITCH***, em cada bloco é preciso colocar o comando ***BREAK.***

Estrutura de Repetição

Com teste lógico no início

let c = 1

while (c <= 6) {

    console.log(`Passo ${c}`)

    c++

}

Passo 1

Passo 2

Passo 3

Passo 4

Passo 5

Passo 6

Com teste lógico no final

let c = 1

do {

    console.log(`Passo ${c}`)

    c++

}  while (c <= 6)

Passo 1

Passo 2

Passo 3

Passo 4

Passo 5

Passo 6

Com variável de controle

for (let c = 1;c <= 5;c++) {

    console.log(c)

}

1

2

3

4

5

console.log('Vai começar...')

for (let c = 1;c <= 5;c++) {

    console.log(c)

}

console.log('FIM!')

Vai começar...

1

2

3

4

5

FIM!

**VARIÁVEIS COMPOSTAS**

Devem ser capazes de armazenar vários valores em uma mesma estrutura.

* Um ARRAY,VETOR ou uma VARIÁVEL COMPOSTA é uma variável que tem vários elementos. Cada elemento é composto por seu valor e por uma chave de identificação.

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

num.push(1) //Vai colocar valor 1 no final

num.sort() //Método interno dentro de um vetor que coloca em ordem

console.log(num)

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

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

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

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]}`)\*/

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

**FUNÇÕES**

* São ações executadas assim que são chamadas ou em decorrência de algum evento.
* Uma função pode receber parâmetros e retornar um resultado.

***... DAQUI PRA FRENTE***

**ESTUDAR:**

* Functions
* Objetos
* Modularização
* RegEx // Expressões Regulares
* JSON // Estrutura de objetos que consegue transmitir dados, salvar
* AJAX
* NodeJS

**OBJETOS:**

**Vetor**

Array para coisas simples

**let** num = [ 5 , 8 , 4 ]

|  |  |  |
| --- | --- | --- |
| **5** | **8** | **4** |

0 1 2

**PARA DECLARAR OBJETO:**

**Objeto**

**let** amigo = { nome:’José’ , sexo:’M’ , peso:85.4 , engordar(p){} }

|  |  |  |  |
| --- | --- | --- | --- |
| José | M | 85.4 | [function] |

Nome sexo peso engordar()