Modulo B

Aula 1 – Variaveis

* Comentários: // para linha unica ou /\* ---- \*/ para várias linhas

Aula 2 –

Typeof var

Aula 3 – Manipulação de dados

var nome = window.prompt(‘qual seu nome?’)

alert(“olá ‘ + nome) –> + representa concatenação

+ serve para adição e concatenação, logo há certa confusão já que o valor do prompt é recebido como string e deve ser convertido com uma das seguintes:

\* Number.parseInt(n) – para inteiros

\*Number.parseFloat(n) – para quebrados

que pode ser feito na declaração da variavel

Number.parseInt(var nome = window.prompt(‘qual seu nome?’))

também pode usar só Number(n), mas é o js que decide e pode ser pouco confiavel

//para string

String(n)

n.toString()

// string template para escrever strings com variaveis em seu meio

- é delimitado por crases `` e recebe variaveis como ${n}

window.alert(`seu número é ${(n1+n2)}`)

// métodos sobre strings

n.length > retorna a largura da string

n.toUpperCase

n.toLowerCase

// formatando números

n1=1543.5

n1.toFixed(2) > 1543.50, força o número a ter n casas decimais

para usar virgula

n1.toFixed(2).replace(‘.’, ‘,’)

n1.toLocaleString(‘pt-br’, {style: ‘currency’, currency:‘BRL’})

#Aula 7 operadores

5/2 = 2.5, / indica divisão real, ou seja, numeros reais

5%2=1, resto da divisão inteira

5\*\*2=25, potência

//Ordem de precedência

()

\*\*

\* / %

+ -

e das esquerda para a direita

//Atribuição

n = 3

\* auto atribuições

n = n/2 + 1

\* para atribuições apenas com a própria váriavel pode-se simplificar:

\* n+=4 --→ n=n+4

\* n-= 5

n\*=4, n=n\*4

n/=2, n=n/2

n\*\*=2, n=n\*\*2

n%=5, n=n%5

//incremento e decrecimo

x++ –> x=x+1

x-- → x=x-1

# Aula 08 Operadores parte 2

5== ‘5’ – true > JS não considera tipo

- Igualdade Restrita

5 === ‘5’ - false > esse operador testa identidade em valor e tipo

Desigualdade restrita

!==

* Operadores lógicos

! 🡪 Implica negação

Operador unário, segue de true ou false, ou expressão equivalente, algo que !true (não verdadeiro) é false, !false é true

&& 🡪 Implica conjunção

Tem dois operando e só resulta em true se ambos forem true, ou seja, se todos os requisitos forem preenchidos

|| 🡪 Implica disjunção

Tem dois operando e resulta em true se qualquer um deles for true

Ordem de solução:

1 – aritméticos

2 – relacionais (da esquerda para a direita)

3 – lógicos ( executados na ordem !, && por fim ||)

Operador Ternário ? :

Se dá em três blocos:

Teste?(Se o teste é true) : (Se o teste é false)