Segunda - Java

Terça - Node.js

Quarta - JavaScript

Quinta – JavaScript

Sexta – JavaScript

Sabado – Trabalhos/Tarefas

Domingo – Trabalhos/Tarefas

Bibliografia:

JavaScript o guia definitivo.

JavaScript guia do programador.

Guia de referência da mozilla.

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide>

Guia de referência da ecma.

<https://www.ecma-international.org/publications/standards/Ecma-262-arch.htm>

Comentário dentro do Script ou seja do JS:

// Uma única linha

/\*

Mais de uma linha de código

\*/

Sinais:

= (atribuição)

+= (atribuição de adição)

-= (atribuição de subtração)

\*= (atribuição de multiplicação)

/= (atribuição de divisão)

%= (atribuição de resto)

\*\*= (atribuição exponencial)

<<= (atribuição bit-a-bit por deslocamento à esquerda)

>>= (atribuição bit-a-bit por deslocamento à direita)

&= (atribuição AND bit-a-bit)

^= (atribuição XOR bit-a-bit)

|= (atribuição OR bit-a-bit)

Igual (==) Retorna verdadeiro caso os operandos sejam iguais.

Não igual(!=) Retorna verdadeiro caso os operandos não sejam iguais.

Estritamente igual (===) Retorna verdadeiro caso os operandos sejam iguais e do mesmo tipo.

5 == ‘5’ (true); 5 === ‘5’ (false)

Estritamente não igual (!==) Retorna verdadeiro caso os operandos não sejam iguais e/ou não sejam do mesmo tipo.

5 != ‘5’ (false); 5 !== ‘5’ (true)

Maior que (>) Retorna verdadeiro caso o operando da esquerda seja maior que o da direita.

Maior que ou igual (>=) Retorna verdadeiro caso o operando da esquerda seja maior ou igual ao da direita.

Menor que (<) Retorna verdadeiro caso o operando da esquerda seja menor que o da direita.

Menor que ou igual (<=) Retorna verdadeiro caso o operando da esquerda seja menor ou igual ao da direita.

++ (incremento)

Pré incremento (++x)

Pós incremento (x++)

-- (decremento)

Pré decremento (--x)

Pós decremento (x--)

Bit a bit:

AND (a & b)

OR (a | b)

XOR (a ^ b)

NOT (~ a)

Lógico:

AND (&&)

OR (||)

NOT (!)

No JavaScript moderno, além de utilizar a palavra var, também podemos usar a palavra let.

No JavaScript pode usar três formas de delimitar uma string. “ “ ‘ ‘ ` `

Identificadores:

Podem começar com: Letra, $ ou \_

Não podem começar com números.

É possível utilizar acentos e símbolos.

Não podem conter espaços.

Não podem ser palavras reservadas.

Variáveis:

number: 5,18,-12,0.5,-15.9,3.14,8.0

string: “Google” , ‘JavaScript’ , `Maria`

Boolean: true , false

Operadores Aritmétricos:

+ 5 + 2 -> 7

- 5 – 2 -> 3

\* 5 \* 2 -> 10

/ 5 / 2 -> 2,5

% 5 % 2 -> 1

\*\* 5 \*\* 2 -> 25

Ordem de Precedência

()

\*\*

\* / %

+ -

Operador ternário:

? :

Teste ? true : false

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

Var r = Idade >= 18 ? “Maior” : “Menor”

DOM – Document Object Model.

Conjunto de objetos dentro do navegador que vai dar acesso aos components internos do web site.

Árvore DOM do site ex005

window

Location Document History

html

head body

meta title h1 p p div

strong

………………………………………………………nao vai ate o fim, é so um exemplo para ficar claro……………

Metodos de acessos

Por Marca (tag)

getElementsByTagName(‘p’)[0] (entre colchetes é o número do paragrafo sendo o primeiro 0)

Por ID

getElementById()

Por Nome

getElementsByName()[0]

Por Classe

getElementByClassName()

Por Seletor

querySelector()

querySelectorAll()

quando é id(div) se usa #

quando é class se usa .

Eventos DOM

<https://developer.mozilla.org/en-US/docs/Web/Events>

Mouse: so alguns exemplos

mouseenter

mousemove

mousedown

mouseup

click

mouseout

Funções

Linhas q sao executados quando o evento ocorrer

Function ação(parametros){

bloco

}

Cursos para fazer:

HTML5, CSS.

Functions.

Objetos.

Modularização.

RegEx.

JSON.

AJAX.

NodeJS.