**Informações importantes para JavaScript**

*A linguagem Javascript se divide em 3 tecnologias HTML5( para leitura do arquivo), CSS para estética do arquivo e Javascript propriamente dito para a questão de interatividade com o usuário.*

**Para leitura do arquivo ( Html5)**

- Sempre criar os arquivos com .html no final ( selecionar o html 5 no visual studio)

- Alterar o idioma pra pt-br

- Alterar a parte do código Title para dar um nome ao arquivo

- Usar h1/ para escrever o titulo do arquivo na tela.

- Usar p/ para escrever os parágrafos

Para fazer comentário em html usar /\*

**Estética do arquivo ( CSS)**

- Para começar a usar CSS pra fazer as mudanças estéticas do arquivo use a tag Style na parte head do Arquivo.

- Para fazer alteração do corpo do texto, usar a variável Body no meio da Tag Style e executar os comandos entre chaves {...}

Pra mudar a cor do texto usar o comando Color e pra mudar a fonte Font ( inserir: normal, tamanho com o final pt [ exemplo 25 pt] e nome da fonte[ arial, times, comic sans etc..]

Para fazer comentário em CSS usar !--

**Interatividade do Arquivo (Javascript)**

Para criar um script em JS digite Script na parte Body do arquivo e no meio do comando script digite o comando em letra minúscula ( exemplo window.alert) seguido de parênteses e aspas para execução do comando.

Para adicionar comentários a respeito das ações no javascript vc deve usar // para comentários de uma linha e /\* para comentários de mais linhas

Treino de variáveis no Node JS

Para a criação de uma variável usar var +( identificação da variável) + sinal de recebimento

( = ) exemplo: var nome = Joãozinho

Os tipos de variáveis são chamados tipos primitivos, os principais são number ( números), string( para palavras ou frases) e boolean ( juízos de valor, como verdadeiro ou falso, ou certo e errado)

**Operadores JavaScript**

Tipos de Operadores do JS:

- Aritméticos ( +, - , \*, / , % e \*\*)

São responsáveis pela realização de cálculos através do JS

Exemplos :

5+2 = 7

5- 2 = 3

5\*2 = 10 ( o asterisco representa a multiplicação na linguagem JS)

5/2 = 2.5 ( divisão real dos números)

5%2 = 1 ( nessa operação utiliza-se o resto da divisão inteira)

5\*\*2= 25 ( nessa operação o cálculo é de potência nessa situação a raiz quadrada)

*Ordem de precedência dos operadores aritméticos*

1° ()

2° \*\*

3° \*, / e %

4° + e -

- Atribuições

As atribuições são os operadores responsáveis pelo armazenamento das informações, normalmente representado pelas variáveis.

Existe mais de um tipo de atribuição:

. Atribuição Simples

. Auto Atribuição ( informação atribuída pela própria variável substituindo o comando anterior automáticamente)

. Atribuição com Incremento ( É um operador que serve para simplificar as atribuições simples e auto atribuições) [Não consegui compreender muito bem, olhar com mais calma depois]

Exemplo de atribuição simples:

Var a = 5+3 ( na utilização da variável a apareceria o 8 na tela.)

Var b = a % 5

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

Var d = 10-a/2

Var e = 6\*2/d

Var f = b%e + 4/e

Obs: No momento de criar as variáveis no node JS vai aparecer undefined porém é normal, basta digitar novamente a variável que a informação atribuída irá aparecer.

Exemplo de auto atribuição:

Var n = 3

N= n+4 ( Nesse caso seria feita uma auto atribuição transformando a variável n que antes era 3 em 7 sem que você precise dar esse comando)

Obs: Também é possível usar uma simplificação da sintaxe quando se tratar da repetição da mesma variável como no caso a cima. Um exemplo do que poderia ser feito é o seguinte n+=4 ( Lê-se n mais n recebe 4, mas com uma sintaxe simplificada)

Isso pode ser replicado com todos os operadores aritméticos e é aceito em todas as linguagens exemplo -= , /= ,%= e etc...

Na atribuição com incremento o +=, -= ou qualquer outra redução é substituída pela repetição do operador aritimético caso precise ser utilizada novamente ficando ++ ou – por exemplo

Com o incremento a variável a cima ficaria n++( dando o valor de 7 para n)

Var num= 2

Num = Num + 2 ( novo valor de Num é 5)

Num++

- Relacional

São operações com valor booleano( verdadeiro ou falso), representado pelos símbolos: maior que( >) , menor que (<) , maior ou igual (>=) , menor ou igual (<=) , igual a (==) e diferente ou não igual (!=) .

Exemplos:

5 > 2 Verdadeiro

6 == 6 Verdadeiro

9 < 2 Falso

Obs: Os operadores relacionais podem utilizar variáveis e strings também.

Exemplo:

Var a = 5

Var b = 10

a > b falso

ou no caso das strings

idade >= 21 ai o operador faria uma comparação com a resposta do usuário

*Nos operadores relacionais temos também as chamadas operações de identidade ou de identidade ou igualdade restrita sinalizada por === com o objetivo de diferenciar tipos diferentes que possuam os mesmos símbolos.*

Exemplo:

8 === ‘8’ Falso (sendo que no primeiro caso o 8 é um numero e no segundo uma string.*)*

- Lógico

Serve para operações de negação ( representada pelo símbolo !), operações de conjunção ( &&) e disjunção ( representado pelo ||)

Na negação existe apenas uma única operação e é emitido a resposta verdadeiro ou falso

Na Operação de conjunção ocorre uma operação binaria com dois valores diferentes devendo ser necessário que os dois valores sejam verdadeiros para que seja atendida a operação, sendo todas as outras possibilidades falsas.

Na operação de disjunção analisa-se um valor ou outro devendo apenas um deles ser verdadeiro para ser atendida a operação.

Exemplo:

var a = 5

var b = 2

a > 7 || b==2

true

( No primeiro valor temos a maior que 7 o que seria falso já que a é 5 e no segundo valor temos b igual a 2 o que é verdadeiro fazendo com que a operação seja ‘atendida’)

a > 7 && b==2

false

No segundo caso lidando com os mesmos valores porém por se tratar de uma conjunção a operação não é atendida por ter um dos valores como falso.

- Ternário

Esse tipo de operação possui esse nome por ser dividido em três partes, a parte teste, a parte verdadeiro e a parte falso, sendo representada apenas por dois símbolos ( : ) e ( ?)

Da seguinte maneira teste ? true : false sendo o primeiro um valor especifico e os outros dois as consequências do valor ser verdadeiro ou falso

Exemplo

Var média = 5

Média >= 7 ? ‘aprovado’ : ‘reprovado’ ( sendo aqui o resultado reprovado)

DOM- Document Object Model

(Control +Shift+ P pra abrir o Watch in Chrome)

Para ter acesso ao DOM deve se iniciar o processo normal da criação de um site, html + css + JS no virtual studio.

O DOM foi criado para gerir os documentos do seu servidor em JS no site web, e não é utilizado no node JS apenas em aplicações web.

Arvore DOM é o nome dado a estrutura dos elementos dentro de um programa ( começando por window, que se divide em location, document, e history. Já o document se divide em head e body( a estrutura inicial que usamos no JS.

(Inserir uma imagem da estrutura aqui)

É possível selecionar os elementos através de Marca, ID, Nome, Classe ou Seletor.

O comando *strong* serve para deixar as palavras emnegrito.

O comando *getElementsByTagname* ()[]serve pra utilizar elementos que já foram digitados anteriormente modificando apenas um número.

O comando *innerText*, serve para reconhecer o texto que deve ser mostrado na tela, já o *innerHTML*, serve para reconhecer todas as informações do elemento que você deseja utilizar(incluindo tags e outros comandos).

O comando *getElementsById* serve pra chamar os elementos ID das divs do seu projeto.

**Eventos DOM**

Os eventos DOM servem para realizar e manipular as ações propostas em tela, normalmente acionadas pela expressão function( ){ }.

Existem centenas de elementos DOM, todos eles estão listados na Mozila Desenvolviment Network(MDN) na seção de elementos Javascript, dentre os elementos mais utilizados estão as funções: click( para interatividades usando cliques do mouse),input para trabalhar com alterações nos atributos de um determinado elemento( diferente do Python que tem a função armazenar dados) e etc...

Os eventos DOM podem ser feitos no HTML ou no Javascrit puro.