Anotações JavaScript

Comandos HTML:

<!—Comentário que não aparece no texto-->

<h1>Adicionar Título</h1>

<pt>Adicionar Parágrafo</pt>

    <style>

        body {

        background-color: rgb(68, 0, 255); <!-Alterando a cor da página->

        color: white; <!-Alterando a cor das fontes->

font: normal 20pt Arial; <!-Alterando o tamanho e tipo de fonte->

        }

    </style>

Comandos CSS:

/\* Comentário em CSS \*/

Todo final de comando em CSS colocamos ponto e vírgula (;).

    <style>

        Tudo que eu colocar aqui dentro é referente ao estilo

    </style>

Comandos JavaScript:

Adicionar comentário:

//Para comentários em uma única linha

/\*Para comentários em mais de uma linha\*/

Criando um Script:

</head>

<body>

   <h1>Adicionar Título</h1

   <pt>Adicionar Parágrafo</pt>

   <script>O script será criado aqui</script>

   </body>

</html>

Criando um Alerta:

window.alert("Texto de Alerta")

Criando uma Confirmação:

window.confirm("Digto aqui a confirmação")

Criando uma pergunta em uma caixa de texto com opção do cliente responder:

window.prompt("Digite aqui sua pergunta")

**Variáveis em JavaScript**

**Variável é representada pela sigla “var”**

**Para darmos nomes a essa variável fazemos: var nome**

= significa “recebe”. Ex.: var nome = 10

variável = null Para deixar a variável vazia. Ex.: var nome = null

O nome de cada variável é chamado de Identificador e existem algumas regras para declara-lo:

1. Não pode começar com números, é possível usar letras, número, assentos e símbolos
2. Não podem conter espaços (substitua por “\_”)
3. Não podem ser palavras reservadas pelo JavaScript.

Dicas para criar Indicadores:

1. Maiúscula e minúscula fazem a diferença
2. Utilize nomes coerentes e que remetem a função da variável

Os Tipos Primitivos Primordiais de Variáveis são:

1. Number – Números inteiros ou fracionários
2. String – São cadeias de caracteres: Nomes, números de celulares, número de documento
3. Boolean – True or False

Para identificar o tipo de variável e o que ela é no Node.js, digitamos:

>Typeof NomeDaVariável

Para armazenar um determinado nome digitado em uma caixa de texto como variável devemos utilizar:

var nome = window.prompt("Alerta com caixa de texto")

Para que a variável apareça em outra caixa de texto:

window.alert(É um prazer te conhecer,' + nome)

O “+” é o sinal de concatenação (unir duas coisas).

O window,prompt trata qualquer variável como String, mesmo que seja um Number. Para fazer uma conversão de tipo String para Number utilizamos os seguintes comandos:

Number.parseInt para números inteiros

Number.parseFloat para números decimais

var n1 = Number.parseInt(window.prompt('Digite um número'))

Para realizar um cálculo, armazenando os valores digitados em uma caixa de mensagem e mostrando o resultado em outra:

   <script>

    var n1 = Number(window.prompt('Digite um número'))

    var n2 = Number(window.prompt('Digite outro número'))

    var n3 = Number(window.prompt('Digite outro número'))

    s = (n1 + n2) \* n3

    window.alert('A soma dos números é: ' + s)

      </script>

Para converter Number para String eu digito: String(var) ou n.toString()

Para mostrar um String, armazenando a variável digitada em uma caixa de mensagem e mostrando o resultado em outra:

<script>

    var palavra = String(window.prompt('Seu nome'))

    window.alert('Seu nome é: ' + palavra)

      </script>

Para montarmos uma frase com diversas variáveis, existe uma novidade no ECAMAScript que permite com que façamos isso de forma muito mais eficiente.

No método antigo seria:

> var n='rafa'

undefined

> var nome = 'Rafa'

undefined

> var sobrenome = 'Frias'

undefined

> var idade = '26'

undefined

> ' O ' + nome + ' ' + sobrenome + ' tem ' + idade + ' anos '

' O Rafa Frias tem 26 anos '

No método nome utilizamos o ${var}, ficando desta maneira:

> `O ${nome} ${sobrenome} tem ${idade} anos`

'O Rafa Frias tem 26 anos'

Importante reparar que nesse caso a crase é diferente (`)

Outros comandos de Strings são:

s.lenght //mostra quantos caracteres tem a String

s.toUpperCase() //muda tudo para Maiúscula

s.toLowerCase() //tudo para Minúscula

Outros comandos de Number são:

n.toFixed() //para definir a quantidade de casas decimais

n.toFixed().replace(‘.’ , ‘,’) //para substituir o ponto por vírgula

n.toLocaleString(‘DigiteoLocal’, {style: ‘tipodeitem’, tipodeunidade: ‘subdivisãodaunidade’}) //para colocar o número em alguma unidade específica

**OPERADORES DO JAVASCRIPT**

**Operadores Aritméticos**:

+ //para somar

- //para subtrair

\* //para multiplicar

/ //para dividir

% //para mostrar o resto de uma divisão

\*\* //para elevar a potência

Ordem de Precedência do JavaScript:

Primeiro o que está entre (), depois \*\*, depois \* / % e por último + -.

**Operadores de Atribuição:**

Para declarar uma variável usamos: var n = 3. Nesse caso a variável n passou a valer 3.

Para alternarmos o valor dessa variável podemos fazer: n = n + 2, por exemplo. Nesse caos n fica valendo 5. Para simplificarmos essa operação fazemos: n += 2.

Operadores de Incremento:

n++ //para adicionar um ao valor da variável

n-- //para subtrair um ao valor da variável

**Operadores Relacionais:**

> //maior

< //menor

>= //maior ou igual

<= //menor ou igual

== //igual

!= //diferente

Para toda expressão com operador relacional, o resultado será em valor booleano (true or false).

Operadores Relacionais de Identidade:

=== //testa se os valores são idênticos (tanto de identidade como de tipo)

!== //para saber se tem algo de diferente (identidade ou tipo)

**Operadores Lógicos:**

! //negação (não)

&& //conjunção (e lógico)

|| //disjunção (ou lógico)

Operador de negação (!): Operador unário, a expressão dará true or false. Uma coisa que não é true será false e uma coisa que não é false será true.

Operador de Conjunção (&&): Operador binário que faz a junção de dois valores lógicos e só será true se os dois forem true, caso contrário será false.

Operador de Disjunção (||): Ao contrário do operador de conjunção, basta que um dos valores seja true para que o resultado final seja true.

Se tivermos operadores lógicos, aritméticos e relacionais na mesma frase ele fará primeiro os aritméticos, depois os relacionais e por último os lógicos, como no exemplo:

> a >= 2 || b + 0 == 2

Para colocarmos um espaço entre dois números em que a variável pode se encaixar, fazemos:

> idade = 18 //variável idade é igual a 18

18

> idade >= 18 && idade <= 65 //idade é maior ou igual a 18 e menor ou igual a 65?

True

Nesse caso declaramos que a variável idade é igual a 18 e pedimos para o node.js nos dizer se a variável idade está entre 18 e 65.

**Operador Ternário:**

Esse operador junta 3 partes: o bloco de teste, o bloco de true e o bloco de false.

? //quando for true

: //quando for false

Exemplo:

idade >= 18 ? "maior de idade" : "menor de idade ou aposentado"

'maior de idade'

**Entendendo o DOM (Document Object Model)**

É um conjunto de objetos, dentro do seu navegador, que dará acesso aos componentes internos do seu website.

**Árvore DOM**

**Elementos da árvore:**

window: Começamos a árvore pela raiz, e a raiz dentro do navegador tem o nome de window. Tudo dentro do JS está dentro da window, que é a janela do seu navegador.

location: O local onde está a página, o URL, a posição dela no seu navegador.

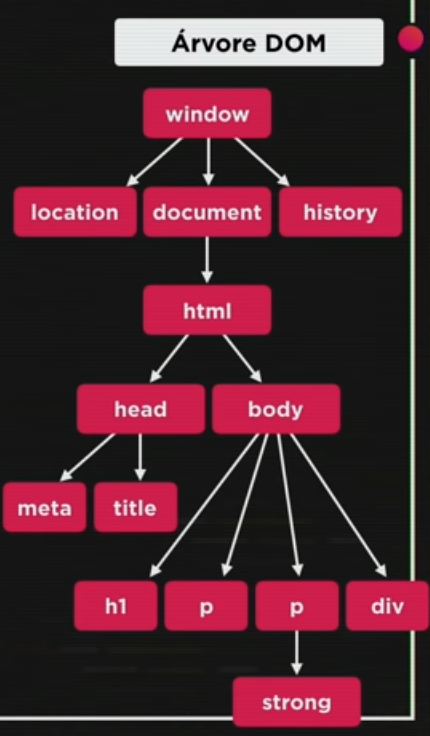
document: É o documento atual.

history:

html: Fica dentro do document.

Head: Filho do HTML. Cabeçalho.

Body: Filho do HTML. Corpo

.

**Tipos de SELEÇÃO**

Seleção por MARCA (getElementsByTagName()[]): Armazena determinados elementos.

    <p>Aqui vai o parágrafo</p>

        <script>

            var p1 = window.document.getElementsByTagName('p')[0]

            window.document.write('Está escrito assim: ' + p1.innerText)

 </script>

Esse comando fará com que os elementos armazena em “p” sejam armazenados. Neste caso, estamos armazenado eles em uma variável “p1” e fazendo com que ele apareça na window através de um “window.document.write()<script>”.

Seleção por ID (getElementById()): Armazena o ID.

Seleção por Name (getElementsByName()[]): Armazena o nome do objeto.

Seleção por Classe (getElementsByClassName()[]): Armazena por tipo.

Seleção por Seletor (querySelector() ou querySelectorAll()): Uma maneira nova e um pouco diferente de fazer essa seleção.

    <div class="msg">Primeiros Passos</div>

        <script>

            var divineia = window.document.querySelector('div.msg')

            divineia.style.color = 'red'

            divineia.style.background = 'white'

      <script>

Toda “id” é representada pela #, toda “class” é representada por “.”. Nesse caso utilizamos “div.msg” porque a “msg” é uma class. Se fosse um id seria “div#msg”.

No exemplo acima utilizando o “querySelector” armazenamos o que estava em div em uma variável chamada “divineia”, fizemos alterações na cor da fonte e em seguida no background da linha em que ele etá inserido.

Alguns comandos de CSS para arrumar o design da página:

<style>

        div#area { //abre a área da div para formatação

        background-color: red; //muda o fundo em que a div está

        color: white; // muda a corda fonte da div

        font: normal 20pt Arial; //muda o tipo e tamanho da fonte da div

        width: 200px; // diz a largura do background da div

        height: 200px; // diz a altura do background da div

        line-height: 200px; // alinha a o texto da div no centro do fundo

        text-align: center; //centraliza a div no background dela

        }

</style>

Eventos é tudo que pode acontecer com um determinado elemento.

Função: É um conjunto de códigos que será executado quando um evento ocorrer.

O bloco que contém os códigos da função deve estar entre {} e ser nomeado como uma “function”. No JS para executarmos uma ação temos que unir a ela o nome da ação que ela fará. Ex,: function ação() {bloco de códigos}. Dentro dos parênteses podem ser inseridos um ou mais parâmetros.

Função em HTML5:

<body>

<div id ='area' onclick="clicar //criei uma função para a div caso cliquem nela

Interaja...

<script>

var clique = window.document.getElementById('area') //crio variável ‘clique’

function clicar() { //crio uma função para acontecer caso acionem ‘clicar’

   clique.innerText = 'Lindo!' //Declarei o texto que aparecerá depois do click

   clique.style.background = 'red' //cudo a cor do fundo após o clique

}

</script>

Mais de uma ação pode ser adicionada a mesma variável como sair do local, entrar no local, etc... Ex.:

<body>

    <div id ='area' onclick="clicar()" onmouseenter="entrar()" onmouseout="saiu()">

        <strong>Você...</strong>

    </div>

    <script>

        var clique = window.document.getElementById('area')

        function clicar() {

             clique.innerText = 'Lindo!'

             clique.style.background = 'red'

        }

        function entrar() {

             clique.innerText = 'É um...'

             clique.style.background = 'blue'

        }

        function saiu() {

             clique.innerText = 'Você...'

             clique.style.background = 'black'

        }

    </script>

Criando uma calculadora de soma com HTML5, CSS6 e JS:

<body>

    <h1>Somando Valores!</h1> //Título

    <input type="number" name="txtn11" id="txtn1"> //criando botão para inserir valor

    <input type="number" name="txtn2" id="txtn2"> //criando botão para inserir valor

    <input type="button" value="Somar" onclick="somar()"> //criando botão para somar

    <div id="res">Resultado</div> //resultado aparecerá nesta linha

    <div>Fiquei 200 anos para achar um erro. Que raivaaaaa!</div>

        <script>

        function somar() { //criando uma função de soma

        var tn1 = window.document.getElementById('txtn1') //identificando o que foi digitado

        var tn2 = window.document.querySelector('input#txtn2') //identificando o que foi digitado

        var res = window.document.getElementById('res') //identificando o id “res”

        var n1 = Number(tn1.value) //convertendo variável em Number

        var n2 = Number(tn2.value) //convertendo variável em Number

        var s = n1+n2 //Somando as variáveis

        res.innerHTML = `A soma entre ${n1} e ${n2} é igual a ${s}` //Mostrando resultado

        }

    </script>

**CONDIÇÕES EM JAVASCRIPTS**

A estrutura “if” é responsável por determinar uma condição, significa SE na linguagem do JS. Já a palavra SENÃO é representada pela estrutura “else”, ela ocorrerá caso a condição determinada não se aplicar a if.

**Condição Simples:** Quando a condição só tem um bloco (if).

Exemplo doc.js:

var nac = 'Espanha'//Declarei uma variável

console.log('Em que país você nasceu?') //console.log para exibir na tela

if (nac == 'Brasil') { //se a var for igual a Brasil executo a função

console.log(`Brasileiro nascido no ${nac}`) //exibo na tela caso if seja true

}

**Condição Composta:** Quando há duas condições (if e else).

var nac = 'Espanha'//Declarei uma variável

console.log('Em que país você nasceu?') //console.log para exibir na tela

if (nac == 'Brasil') { //se a var for igual a Brasil executo a função

console.log(`Brasileiro nascido no ${nac}`) //exibo na tela caso if seja true

} else { //executo caso if seja false

console.log(`Estrangeiro nascido no país: ${nac}`) //exibir caso if seja false

}

Programa que diz se você está ou não acima da velocidade permitida, digitando o valor da sua velocidade em uma caixa de texto:

<body>

<h1>Site do DETRAM</h1>

Velocidade do carro: <input type="number" name="vel" id="vel"> Km/h

<input type="button" value="Verificar" onclick="calcular()">

<div id= "res"></div>

<script>

function calcular(){

var velcar = document.querySelector('input#vel') var res = document.querySelector('div#res')

var velocidade = Number(velcar.value)

res.innerHTML = `</br><p1><b>Sua velocidade é de ${velocidade}Km/h</p>`

if (velocidade > 60) {

res.innerHTML += '<p3><b>Você está acima do limite de velocidade permitido. Reduza!</p>'

} else {

res.innerHTML += '<p4><b>Dentro do limite de velocidade permitido.</p>'

}

res.innerHTML += '<p2><b>Dirija sempre com cinto de segurança!</b></p>'

}

</script>

Site que mostra seu nome na tela, calcula seu IMC e te dá informações sobre o assunto:

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        body {

        background-color: rgb(0, 0, 0);

        color: rgba(255, 255, 255, 0.808);

        font: normal 15pt Arial;

        }

        h1{

            font: normal 40pt Impact;

            color: rgb(143, 226, 105);

        }

    </style>

</head>

<body>

    <h1>DESCUBRA O SEU IMC</h1>

    <pt>DESCUBRA SEU IMC. RÁPIDO E SIMPLES!</pt></br></br>

    Digite seu Peso: <input type="number" name="txpeso" id="txpeso">Kg

    Digite sua altura: <input type="number" name="txaltura" id="txaltura"> m

    <input type="button" value="Meu IMC" onclick="calcular()">

    <div id = 'imc'></div>

    <script>

        var nome = String(window.prompt('Olá! Qual é seu nome?'))

        window.alert('A seguir faremos algumas perguntas com a finalidade de descobrir o seu IMC. Para continuar clique em OK.')

        document.write(`</br>Olá, <b>${nome.toUpperCase()}</b>. Seja bem vindo!</br>`)

        function calcular(){

        var pes = document.querySelector('input#txpeso')

        var alt = document.querySelector('input#txaltura')

        var res = document.querySelector('div#imc')

        var peso = Number(pes.value)

        var altura = Number(alt.value)

        resultado = peso/(altura\*altura)

        res.innerHTML = (`</br>Sabendo que sua altura é <b>${altura.toFixed(2)}m</b> e seu peso é <b>${peso.toFixed(2)}kg</b>, concluimos que seu IMC é igual a <b>${resultado.toFixed(2)}</b>.`)

        }

        document.write(`Abaixo há uma tabela para que você saiba como está sua saúde em relação ao IMC.</b>.</br></br>`)

        document.write('</br>Abaixo de 18,5: Você está abaixo do peso ideal.</br>')

        document.write('Entre 18,6 e 24,9: Você está no peso ideal. Parabéns!</br>Entre 25 e 29,9: Levemente acima do peso.</br>')

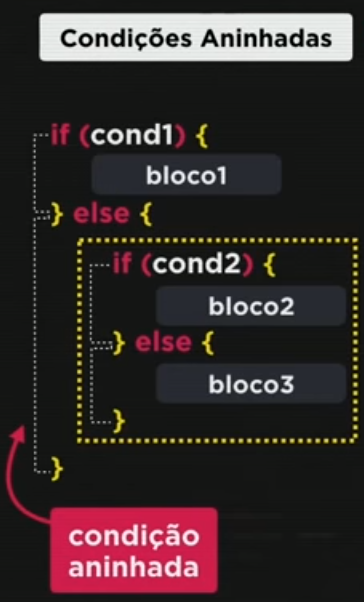
        document.write('Entre 30 e 34,9: Obesidade Grau I.</br>Entre 35 e 39,9 Obesidade Grau II (severa).</br>Acima de 40: Obesidade Grau III (mórbida).')

    </script>

</body>

</html>

**Condições Alinhadas (else if()):** Condições que tem mais de duas respostas. Para isso alinho outros blocos a um dos blocos da função.



Exemplo de “else if” em um programa.js que diz se o indivíduo tem voto opcional, obrigatório ou não tem direito de voto:

var id = 64

console.log(`Sua idade é ${id}.`)

if (id < 16) {

    console.log('Não Vota!')

} else if (id < 18 || id >= 65) {

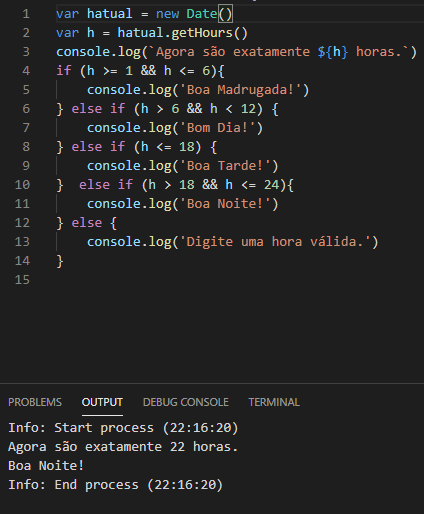
    console.log('Voto Opcional!')

} else {

    console.log('Voto Obrigatório!')

}

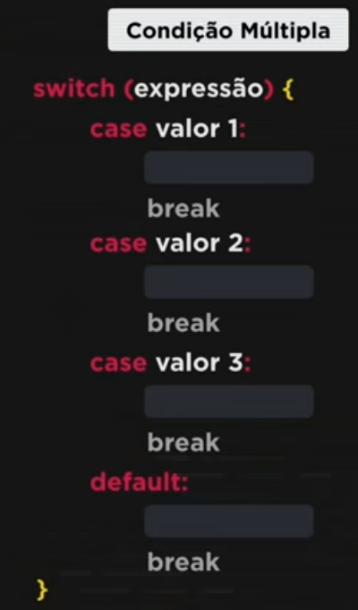
Utilizando vários “else if” no mesmo programa e utilizando uma variável capaz de identificar a hora atual:



**Condição Múltipla “switch (expressão)”:** Usada, principalmente, para trabalharmos com valores fixos.



Estrutura de Condição Múltipla (switch) dentro do JavaScript:



Exemplo de um programa que utiliza o switch para identificar o dia da semana:

var agora = new Date()

var dia = agora.getDay()

var dia = 02

switch (dia) {

    case 0:

        console.log('Domingo')

        break

    case 1:

        console.log('Segunda')

        break

    case 2:

        console.log('Terça')

        break

    case 3:

        console.log('Quarta')

        break

    case 4:

        console.log('Quinta')

        break

    case 5:

        console.log('Sexta')

        break

    case 6:

        console.log('Sábado')

        break

    default:

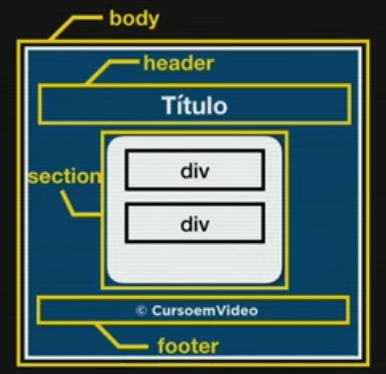
        console.log('Desculpe! Algo deu errado.')

        break

}

**Exercícios Utilizando HTML5, CSS3 e JavaScript**

Monte um site com a seguinte estrutura:



Código para realizar o exercício:

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Modelo de Exercício</title>

    <link rel="stylesheet" href="estilo.css"> //abrir nova guia\*

</head>

<body>

    <header>

        <h1>Exercícios</h1>

    </header>

    <section>

        <div>

            Teste

        </div>

        <div>

        </div>

    </section>

    <footer>

        <p>&copy; pt-RafaBR</p>

    </footer>

\*Digitando “link” e clicando em “css” você abre essa linha de código. Clicando em cima do nome que você deu para o link e apertando crtl, o Visual Studio Code vai dizer que o arquivo não existe e perguntará se você quer abrir um arquivo separado do código. Ao clicar em sim, uma aba paralela irá se abrir e nela você pode configurar toda a parte de css do seu código. Funciona da mesma maneira para os comandos de JavaScript utilizando o comando:

<script src="scriptdosite"></script>

A seguir, parte de CSS do exercício 1 que está em aba “estilo.css” no código acima:

body{ //configurações de CSS do corpo

    background-color: rgb(101, 119, 221);

    color: rgb(255, 255, 255);

    font: normal 15pt arial;

}

header { //configurações de CSS da cabeça

    text-align: center;

    text-shadow: 2px 2px 2px black;

}

section { //configurações de CSS da seção

    background-color: rgb(255, 255, 255); //define a cor do background

    color: rgb(0, 0, 0); //define a cor da fonte

    border-radius: 10px; //deixa as bordas do preenchimento arredondadas

    padding: 25px; //aumenta o tamanho do preenchimento

    width: 400px; //muda a largura do preechimento

    margin: auto; // //centraliza o preenchimento

    text-align: center; //centraliza o texto

    box-shadow: 5px 5px 5px rgba(0, 0, 0, 0.363); //cria um sombreamento

}

footer { //configurações de CSS do rodapé

    text-align: center;

    font-style: italic;

    text-shadow: 2px 2px 2px black;

}

Resultado do código acima:

