***RESUMO CURSO JAVASCRIPT***

***Curso Em Vídeo - Professor Gustavo Guanabara*** *(200 a 299)*

**200) - Módulo-A - Conhecendo o JavaScript**

**200.1) - Aula-01 - O que o JavaScript é capaz de fazer**

1.1) - Case sensitive

1.2) - O que o JavaScript faz

1.3) - Para desabilitar o JavaScript no Google

**200.2) - Aula-02 - Como chegamos até aqui**

2.1) - O início

2.2) - A DARPA criou a rede ARPANET

2.3) - A origem do nome JavaScript

2.4) - ECMA

2.5) - O Google Chrome

**200.3) - Aula-03 - Dando os primeiros passos**

3.1) - Instalação dos programas Chrome, Visual Studio Code e NodeJS.

**200.4) - Aula-04 - Criando o seu primeiro programa**

4.1) - A tag <script> </script

4.2) - Comando window.alert

4.3) - Comando window.confirm()

4.4) - Comando window.prompt()

**201) - Módulo-B - Comandos básicos do JavaScript**

**201.1) - Aula-05 - Variáveis e tipos primitivos**

1.1) - Comentários

1.2) - Node JS

1.3) - Variáveis e Tipos primitivos

1.4) - Nomes de variáveis em JavaScrip

1.5) - Tipos Primitivos primordiais de variáveis em JavaScript

1.6) - *Comando typeof*

**201.2) - Aula-06 - Tratamento de dados**

2.1) - Armazenar dados digitados na janelinha prompt

2.2) - Conversão de tipos string p/ Inteiros ou Float e vice-versa

2.3) - Manipulando strings

2.4) - Formatando números

**201.3) - Aula-07 - Operadores (Parte-1)**

3.1) - Os operadores são:

3.2) - Ordem de precedência das operações

3.3) - Auto Atribuição, Pós-Incremento e Pré-Incremento

**201.4) - Aula-08 - Operadores (Parte-2)**

4.1) - Operadores Relacionais

4.2) - Operadores Lógicos

4.3) - Operador Ternário

4.4) - Ordem Geral de Precedência dos Operadores

**202) - Módulo-C - Entendento o DOM**

**202.1) - Aula-09 - Introdução ao DOM**

1.1) - O que é o

1.2) - A árvore DOM

1.3) - Árvore DOM do ex005.html

1.4) - Selecionando Elementos para se navegar dentro da Árvore DOM

1.5) - Informações complementares obtidas no site abaixo sobre fontsize e fontStyle

**202.2) - Aula-10 - Eventos DOM**

2.1) - O mais comum são eventos de *mouse*

2.2) - Funções

**203) - Módulo-D - Condições em JavaScript**

203.1 - Aula-11 - Condições (Parte-1)

203.2 - Aula-12 - Condições (Parte-2)

203.3 - Exercícios JavaScript (Parte-1)

203.4 - Exercícios JavaScript (Parte-2)

203.5 - Exercícios JavaScript (Parte-3)

**204) - Módulo-E - Repetições em JavaScript**

204.1) - Aula-13 - Repetições (Parte-1)

204.2) - Aula-14 - Repetições (Parte-2)

204.3) - Exercícios JavaScript (Parte-4)

204.4) - Exercícios JavaScript (Parte-5)

204.5) - Exercícios JavaScript (Parte-6)

**205) - Módulo-F - Avançando os estudos em JavaScript**

205.1) - Aula-15 - Variáveis compostas

205.2) - Aula-16 - Funções

205.3) - Exercícios JavaScript (Parte-7)

205.4) - Exercícios JavaScript (Parte-8)

205.5) - Aula-17 - Proximos passos

**-------------------------------------------------------------------------------------------------------------------------------------------**

**300) - Tópicos a pesquisar** *(citados em livros mas não apresentados)*

**400) - Dicas e configuração**

**500) - Siglas**

**-------------------------------------------------------------------------------------------------------------------------------------------**

***Resumo Curso JavaScript -***

***Professor Gustavo Guanabara*** *(200 a 299)*

**200) - Módulo-A - Conhecendo o JavaScript**

**200.1) - Aula-01 - O que o JavaScript é capaz de fazer**

**1.1) - Case sensitive -** Atenção**,** JavaScript é uma linguagem Case Sensitive, ou seja, faz diferença entre maiúsculas e minúsculas.

**1.2) - O que o JavaScript faz** - promove a interatividade dos sites como abrir menus, digitar usuário e senha, abrir vídeos, aumentar uma foto, clicar em botões, etc.

**1.3) - Para desabilitar o JavaScript no Google** - acesse a página do Google, clique com o botão direito na área em branco e escolha inspecionar. Em seguida clique em Configurações, role a tela pra baixo e marque Desativar JavaScript. Você verá que a interatividade do site desaparece. Experimente desabilitar no YouTube, tudo vai sumir, pois ele é totalmente dependente do JavaScript.

**200.2) - Aula-02 - Como chegamos até aqui? A origem da Internet.**

**2.1) - O início** - Na década de 60, em plena guerra fria, o presidente dos USA Dwight D. Eisenhower *(foi o*[*34º*](https://pt.wikipedia.org/wiki/Lista_de_presidentes_dos_Estados_Unidos)[*Presidente*](https://pt.wikipedia.org/wiki/Presidente_dos_Estados_Unidos)*dos*[*Estados Unidos*](https://pt.wikipedia.org/wiki/Estados_Unidos)*de 1953 até 1961.)*

**DARPA** - **Defense Advanced Research Projects Agency**Agencia de pesquisa tecnológica cuja função era pesquisar tecnologias para a guerra. O que era é porque foi criada a DARPA?

- A Agência de Projetos de Pesquisa Avançada de Defesa (**DARPA**) dos militares dos EUA **foi criada** em 1958 em resposta ao lançamento do satélite soviético Sputnik. Seu objetivo **era** garantir que os Estados Unidos nunca mais ficassem para trás na corrida da tecnologia.

**2.2) - A DARPA criou a rede ARPANET** - Criada para o governo e os militares mas tina também muitas universidades dentro.

**2.3) -** **A origem do nome JavaScript** - O desenvolvedor ***Brandon*** ***Eich*** da ***Netscape*** *(criada em 1994)*, recebeu a tarefa de criar uma linguagem que desse mais funcionalidade ao simples ***HTML*** que era uma linguagem estática. E assim em 1995 foi criada a linguagem ***JavaScript***. O nome não tem nada a ver com linguagem Java. O nome inicial era ***Mocha*** *(pronuncia-se Moca)* que depois passou a se chamar ***LiveScript***, o qual foi substituído por ***JavaScript***, simplesmente porque a linguagem Java na época fazia um enorme sucesso. Portanto a adoção do nome JavaScript foi uma grande jogada de marketing.

**2.4) - ECMA** - Em 1997 a Netscape cedeu o código do JavaScript para a ECMA que é uma empres padronizadora da europa a qual é como a ISO nos Estados Unidos. Em 2002 a Netscape acabou falindo, pois a Microsoft espertamente fornecia o sistema operacional Wondows com o ***Internet Explorer*** embutido, ai ninguém precisava mais baixar o Netscape. Porem as pessoas que trabalhavam na Netscape resolveu criar a ***Fundação*** ***Mozilla*** que não desistiu de fazer navegador, ele só não se chamava ***Netscape***. Ganhou uma nova roupagem, um novo código e passou a se chamar ***Firefox***.

**2.5) - O Google Chrome** - Criado em 2008 tinha um motor *(engine)* de *JavaScript* muito poderoso e foi em 2009 que surgiu a *versão* *V8* *(uma referência aos motores V8)*.

- O *V8* é o motor debaixo do capô do *Google* *Chrome* que roda *JavaScript*. Uma das grandes vantagens do *V8* é que ele gerava código *JIT* - *Just* *In* *Time* que é a capacidade de compilar apenas quando o programa é executado, em vez de dividir a compilação e a execução inicial em dois estágios separados.

- Cada navegador tem seu próprio motor: o *Firefox* tem o *SpiderMonkey*, o *Safari* tem o *Nitro*, o *Opera* tem o *Karakan* e no *Edge* da *Microsoft* é o *Chakra*.

- O *V8* por ser um código aberto acabou dando origem, em 2010, a uma nova ferramenta chamada *NODE.JS*, que é uma máquina que roda *JavScript* fora do navegador, e isso possibilitou que o *JavaScript* rodasse também no lado do servidor e não só no lado do cliente.

**200.3) - Aula-03 - Dando os primeiros passos**

**3.1) - Instalação dos programas Chrome, Visual Studio Code e NodeJS.**

**200.4) - Aula-04 - Criando o seu primeiro script**

**4.1) -** **A tag <script> </script>** - Em um código HTML, dentro desta tag vai os comandos JavaScript.

- *Os comandos podem podem ou não ser finalizados com ponto-e-vírgula.*

- Para adicionar a tag <hr> dentro da tag <script> usa-se o comando ***document.write(“("<hr>”).***

**4.2) -** **Comando window.alert()** - Esse comando é executado antes do código e abre uma janela de mensagem contendo um botão ‘OK’.

- A mensagem precisa estar entre aspas simples ou dupla.

**4.3) -** **Comando window.confirm()** -

**4.4) -** **Comando window.prompt()** -

- A palavra window é opcional e pode não ser digitada, ver abaixo:

***<script>***

***window.alert("Esta é minha primeira mensagem.");***

***alert('Minha segunda mensagem!');***

***window.confirm("Está gostando de JS?");***

***window.prompt("Qual é o seu nome?");***

***</script>***

-------------------------------------------------------------------------------------------------------------------------------------------

**201) - Módulo-B - Comandos básicos do JavaScript**

**201.1)** - **Aula-05 - Variáveis e Tipos Primitivos**

**1.1) - Comentários**

**-** Comentários de linha - Use duas barras - *//Isto é um comentário de linha.*

- Comentários de bloco de linhas - Use /\* para abrir o bloco e \*/ para fechar - /\*Isto é um comentário de bloco.\*/

**1.2) - Node JS**

- Para abrir o Node no Windows clicar no ícone do programa. Para fechar digitar comando ***.exit*** *(ponto exit)*

- Para abrir dentro do *Visual Code* clicar no menu *Terminal* e em *Novo* *Terminal*. Será aberto na parte inferior da tela:

- Para ***expandir*** clique na setinha na parte superior ao lado esquerdo de X.

- Para ***fechar*** digitar comando ***.exit*** *(ponto exit)* ou clique no *X.*

- Para ***limpar*** a tela digitar ***CRTL+L***.

**- Para repetir um comando digitado clique em seta para cima.**

**1.3) -** **Variáveis e Tipos primitivos**

- Os nomes de variáveis são precedidos pelas palavras *var*, *let* ou *const*.

- **Atenção**: O sinal de ***=*** *(igual)* é lido como *RECEBE*: var nome recebe “João”.

- *var* nome = “João”

- *let* idade = 23

- *const* cel = “99999-9999”

**1.4) -** **Nomes de variáveis** **em JavaScrip**

- Os nomes das variáveis são chamados de identificadores. Use nomes coerentes para as variáveis.

- Podem começar com uma ***letra****)*, um ***sifrão*** e ***traço de sublinhado \_.***

- Podem utilizar letras e números

- Podm utilizar acentos e símbolos

- Maiúsculas e minúsculas fazem diferença. Ex. var Idade = 34, var IDADE = 25

- Não podem começar com números.

- Não podem conter espaço

- Não podem ser palavras reservadas - ex. *function*, *alert*, etc.

**1.5) -** **Tipos Primitivos primordiais de variáveis em JavaScript**

- O JavaScript infere o tipo da variável pelo valor que ela recebe. Diferentemente do *Java* não é preciso especificar o tipo da variável antes do nome da mesma.

***- Tipo number*** - Subdivide-se em o *Infinity* e *NaN* *(Not a Number)*

***- Tipo string*** - Para concatenar strings use o sinal de ***+*** ou a técnica de ***Template String*** presente nas versões mais atuais do ***EcmaScript*** utilizando o ***placeholder*** ***${ }.*** O comando precisa estar entre ***sinais de crase*** ***` `***.

Ex. *var nome = “Antonio”;*

*var idade = 67;*

*var nota = 9.5;*

- Concatenação com o sinal ***+:*** "O aluno " ***+ nome +*** " com " ***+ idade +*** " anos tirou " ***+ nota***

- Concatenação com o ***placeholder*** ***${ }:*** `O aluno ***${nome}*** com ***${idade}*** anos tirou nota ***${nota}`***

***- Tipo boolean***

***- Tipo null***

***-Tipo undefined***

***- Tipo object -*** *Array*

- ***Tipo function***

- ***Tipo number*** - Engloba os números inteiros e Reais *(ou ponto flutante com casas decimais)*.

Ex. 5, 18, -12, 0.5, 5.9, 3.14, 8.0.

- ***Tipo string*** - São cadeias de caracteres que podem conter qualquer coisa desde que estejam dentro de aspas.

- E existem três formas de atribuir strings em JavaScript:

- var s1 = “Javascript” - usando aspas duplas

- var s2 = ‘Curso em vídeo’ - usando aspas simples

- var s3 = `Guanabara` - usando crase

- var CPF = “123.456.789-00”

- var nCel = ‘99999-9999”

- ***Tipo*** ***boolean*** - Valor booleano. Pode receber *true* ou *false.*

**1.6) -** ***Comando typeof***

- Este comando associado a um valor literal retorna o tipo da variável.

- *typeof 6* - retorna tipo *number*

*- typeof 8.5* -retorna tipo *number*

*- typeof NaN* - retorna tipo *number*

*- typeof infinity* - retorna tipo *number*

*- typeof “8.5”* - retorna tipo *string*

*- typeof []* - retorna tipo *object* *(qualquer coisa entre colchetes)*

*- typeof {}* - retorna tipo *object* *(qualquer coisa entre chaves) - Array*

*- typeof null* - retorna tipo *object*

*- typeof function() {}* -retorna tipo *function*

*- typeof undefined* - retorna tipo *undefined*

**201.2) -** **Aula-06 - Tratamento de dados**

**2.1) - Armazenar dados digitados na janelinha prompt**

***<script>***

***var nome = window.prompt("Qual é o seu nome?");***

***window.alert("É um grande prazer te conhecer " + nome + '!');***

***</script>***

**2.2) - Conversão de tipos string p/ Inteiros ou Float e vice-versa**

- Para converter de ***string*** para número ***inteiro*** use ***Number.parseInt(n)*** ou apenas ***parseInt(n)***

- Para converter de ***string*** para número ***decimal*** use ***Number.parseFloat(n)*** ou apenas ***parseFloat(n)***

- ***Atenção:*** Se a conversão for para ***inteiro*** e for digitado um ***decimal*** a eventual soma irá mostrar apenas a parte inteira do resultado.

- Nas versões mais novas do ***EcmaScript*** basta colocar apenas o ***Number*** ele infere (identifica) se é número ***Inteiro*** ou ***Float***:

- Ao invéz de ***Number.parseInt(n)*** ou apenas ***parseInt(n)*** digite apenas ***Number(n).***

- Ao invéz de ***Number.parseFloat(n)*** ou apenas ***parseFloat(n)*** digite apenas ***Number(n).***

- Para converter de número ***inteiro*** para ***string*** use ***String(n)*** ou ***n.toString*** tanto faz.

***- Veja as 04 soluções no código abaixo:***

***<script>***

***/\*O window.prompt retorna uma string  portanto é preciso converter o número digitado de string para número.\*/***

***//Solução-01 - Armazena o número já convertido***

***var n1 = Number.parseInt(window.prompt("Solução-01: Digite um número: "));***

***var n2 = Number.parseInt(window.prompt("Solução-01: Digite outro número: ")) ;***

***var soma = n1 + n2;***

***window.alert("Solução-01: A soma dos valores é: " + soma)***

***//-----------------------------------------------------------------------------------------------------***

***//Solução-02 - Armazena uma string e a conversão é feita na soma dos números***

***var n1 = window.prompt("Solução-02: Digite um número: ");***

***var n2 = window.prompt("Solução-02: Digite outro número: ");***

***var soma = Number.parseInt(n1) + Number.parseInt(n2);***

***window.alert("Solução-02: A soma dos valores é: " + soma)***

***//-----------------------------------------------------------------------------------------------------***

***//Solução-03 - Nas versões mais novas do EcmaScript basta colocar apenas o Number e ele infere (identifica) se é número Inteiro ou Float***

***var n1 = Number(window.prompt("Solução-03: Digite um número: "));***

***var n2 = Number(window.prompt("Solução-03: Digite outro número: ")) ;***

***var soma = n1 + n2;***

***window.alert("Solução-03: A soma dos valores é: " + soma)***

***//-----------------------------------------------------------------------------------------------------***

***//Solução-04 - Soma os valores individuais (n1 + n2)***

***window.alert("Solução-04: Soma os valores individuais (n1 + n2): " + (n1 + n2));***

***//-----------------------------------------------------------------------------------------------------***

***//Solução-05 - Para converter de número Inteiro ou Float para string***

***window.alert("Solução-05: Opção toString() - Concatena os valores: " + n1.toString() + n2.toString());***

***window.alert("Solução-05: Opção String() - Concatena os valores: " + String(n1) + String(n2));***

***</script>***

**2.3) - Manipulando strings**

- ***Atributo length*** - para saber quantos caracteres tem uma string.

- ***Método toUpperCase()*** - para passar todas as letras de uma string para *MAIÚSCULAS*.

- ***Método toLowerCase()*** - para passar todas as letras de uma string para *minúsculas*.

***- n1.toString() -*** Solução-01: Converter número para string

- ***String(n1) -*** Solução-02: Converter número para string

***<script>***

***var nome = window.prompt('Digite seu nome.')***

***document.write(`Olá, <strong> ${nome} </strong>! Seu nome tem ${nome.length} letras <br/>`)***

***document.write(`Seu nome em maiúsculas é <strong> ${nome.toUpperCase()}</strong> <br/>`)***

***document.write(`Seu nome em minúsculas é <strong> ${nome.toLowerCase()}</strong>`)***

***</script>***

**2.4) - Formatando números**

- Use o ***método toFixed(2)*** - para formatar a qtde de casas decimais. Qtde de casas decimais entre parêntesis.

- Use o ***método replace('.' , ',') -*** para substituir o ponto por vírgula.

- Use o ***método toLocaleString('pt-br', {style: 'currency', currency: 'BRL'}) -*** para Inserir o tipo de moeda, separar os milhares com ponto e os centavos por vírgula.

***<script>***

***document.write("<h3>Formatando números</h3>")***

***var salario = 1545.5***

***document.write(`${salario.toFixed(2)} - Método toFixed() fixa 02 casas decimais<br/><hr>`)***

***document.write(`${salario.toFixed(2).replace('.', ',')} - Método replace() substitui ponto por vírgula<br/><hr>`)***

***document.write(`${salario.toLocaleString('pt-BR', {style: 'currency', currency: 'BRL'})} - Método toLocaleString('pt-br') - Insere tipo moeda, separa os milhares com ponto e os centavos por vírgula<br/><hr>`)***

***document.write(`${salario.toLocaleString('en', {style: 'currency', currency: 'USD'})} - Método***

***toLocaleString('en') - Insere tipo moeda, separa os milhares com VÍRGULA e os centavos por***

***PONTO<br/><hr>`)***

***</script>***

**201.3) -** **Aula-07 - Operadores Aritméticos (Parte-01)**

**3.1) - Os operadores são:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operação** | **Valor** | **Operador** | **valor** | **Resultado** |
| Adição | 5 | **+** | 2 | 7 |
| Subtração | 5 | **-** | 2 | 3 |
| Multiplicação | 5 | **\*** | 2 | 10 |
| Divisão | 5 | **/** | 2 | 2.5 |
| Resto da Divisão | 5 | **%** | 2 | 1 |
| Potênciação | 5 | **\*\*** | 2 | 25 |

**3.2) - Ordem de precedência das operações**

- As operações são executadas na seguinte ordem: primeiro executa tudo que está entre ***Parêntesis***, depois vem a ***Potenciação***, em seguida a ***Multiplicação***, ***Divisão*** e ***Resto da Divisão*** *(destes três é executado o que vier primeiro)*, e por último a ***Soma*** e ***subtração*** *(destes dois é executado o que vier primeiro).*

|  |  |  |
| --- | --- | --- |
| **Precedência** | **Operação** | **Operador** |
| **1º** | Parêntesis | **( )** |
| **2º** | Potenciação | **\*\*** |
| **3º** | Multiplicação Divisão Resto da Divisão | **\* / %** |
| **4º** | Soma e Subtração | **+ -** |

**3.3) - Auto Atribuição, Pós-Incremento e Pré-Incremento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribuição simples** | **Simplificando com Auto Atribuição** | **Simplificando com Pós-Incremento** | **Simplificando com Pre-Incremento** |
| var n = 3 | - | - | - |
| var n = n + 4 | **var n += 4** | **var n++** | **var ++n** |
| var n = n - 5 | **var n -= 5** | **var n--** | **var --n** |
| var n = n \* 4 | **var n \* = 4** | - | - |
| var n = n / 2 | **var n /= 2** | - | - |
| var n = n \*\* 2 | **var n \*\*= 2** | - | - |
| var n = n % 5 | **var n %= 5** | - | - |

**201.4) -** **Aula-08 - Operadores Relacionais, Lógicos e Ternário (Parte-02)**

**4.1) - Operadores Relacionais**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operador Relacional** | **Valor** | **Sinal** | **valor** | **Resultado**  **Bolleano** | **Nota** |
| Maior que | 5 | **>** | 2 | **true** |  |
| Menor que | 7 | **<** | 4 | **false** |  |
| Maior ou igual | 8 | **>=** | 8 | **true** |  |
| Menor ou igual | 9 | **<=** | 7 | **false** |  |
| Igual | 5 | **==** | 5 | **true** |  |
| Igual | 5 | **==** | “5” | **true** | O sinal de igualdade não testa o tipo da variável apenas o valor |
| Igual restrito *(Idêntico)* | 5 | **===** | “5” | **false** | Operador de identidade ou igualdade restrita testa o valor e o tipo. |
| Igual restrito *(Idêntico)* | 5 | **===** | 5 | **true** | Operador de identidade ou igualdade restrita testa o valor e o tipo. |
| Não igual (diferente) | 4 | **!=** | 4 | **false** | Não testa o tipo da variável apenas o valor |
| Não igual (diferente) | 4 | **!=** | ‘4’ | **true** | Não testa o tipo da variável apenas o valor |
| Desigual restrito *(Não idêntico)* | 4 | **!==** | “4” | **true** | Operador de desigualdade restrita testa o valor e o tipo. |
| Desigual restrito *(Não idêntico)* | 4 | **!==** | 4 | **false** | Operador de desigualdade restrita testa o valor e o tipo. |

- Em uma expressão contendo operador relacional e aritmético o aritmético tem precedência. Ex.:

*- var a = 8*

*- var b = 15*

*- a <= b - 10 - resultado é falso*

**4.2) - Operadores Lógicos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operador Lógico** |  | **Sinal** | **valor** | **Resultado**  **Bolleano** | **Nota** |
| Negaçaõ | true | **!** |  | **false** | Operador unário |
| Negaçaõ | false | **!** |  | **true** | Operador unário |
| Conjunção (E) | true | **&&** | true | **true** | Operador binário |
| Conjunção (E) | true | **&&** | false | **false** | Operador binário |
| Conjunção (E) | false | **&&** | true | **false** | Operador binário |
| Conjunção (E) | false | **&&** | false | **false** | Operador binário |
| Disjunção (OU) | true | **||** | true | **true** | Operador binário *(sinal duas barras verticais chama-se Pipe)* |
| Disjunção (OU) | true | **||** | false | **true** | Operador binário |
| Disjunção (OU) | false | **||** | true | **true** | Operador binário |
| Disjunção (OU) | false | **||** | false | **false** | Operador binário |

- Em uma ***expressão*** contendo operadores ***aritméticos***, ***relacionais*** e ***lógicos***, primeira precedência são os operadores ***aritméticos***, segunda ***precedência*** são os operadores ***relcionais*** e em seguida os ***lógicos***. Ex.:

- var a = 5

- var b = 8

- (a > b) && (b % 2 == 0) - resultado é falso

- Em uma ***expressão*** contendo um ***&& (E),*** depois um ***|| (OU)*** e um ***! (NÃO),*** a ourdem de execução é sempre:

- Primeiro executa o **!** (NÃO)

- Segundo executa o **&&** (E)

- Terceiro executa o **||** (OU)

**4.3) - Operador Ternário**

- A expressão do operador ternário é composta de 05 partes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Teste lógico** | **?** | **true** | **:** | **false** |

- var média = 5,7

- A expressão abaixo é lida do seguinte modo:

- Se a média for maior ou igual a 7.0 o resultado é “Aprovado” senão resultado é “Reprovado”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **média >= 7.0** | **?** | **“Aprovado”** | **:** | **“Reprovado”** |

- Resultado = “Reprovado”

**4.4) - Ordem Geral de Precedência dos Operadores**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operadores** | **Ordem** | **Operação** | **Símbolo** | **Nota** |
| Aritméticos | **1º** | Parêntesis | **( )** |  |
| **2º** | Potenciação | **\*\*** |  |
| **3º** | Multiplicação | **\*** | Executa o que vier primeiro |
| Divisão | **/** |
| Resto da Divisão | **%** |
| **4°** | Soma | **+** | Executa o que vier primeiro |
| Subtração | **-** |
| Relacionais | **5º** | Maior que | **>** | Executa o que vier primeiro |
| Menor que | **<** |
| Maior ou igual | **>=** |
| Menor ou igual | **<=** |
| Igual | **==** |
| Igual restrito *(Idêntico)* | **===** |
| Desigual (diferente) | **!=** |
| Desigual restrito *(Não idêntico)* | **!==** |
| Lógicos | **6º** | Negaçaõ | **!** |  |
| **7º** | Conjunção (E) | **&&** |  |
| **8º** | Disjunção (OU) | **| |** |  |

**202) - Módulo-C - Entendendo o DOM - Document Object Model**

- O ***DOM*** é a vertente ***Web*** do ***JavaScript*** para a qual a linguagem foi criada. É uma poderosa ferramenta que está dentro do navegador e é um conjunto de objetos ***JavaScript*** para o tratamento de ***componentes visuais*** tornando as aplicações mais belas.

- **Extensão** **Live Server** - Para executar e atualizar automaticamente os arquivos *HTML*, instale a extensão *Live* *Server*: vá em extensões do *VS* *Code* procure por *Live* *Server* e instale. Em seguida clique com o botão direito no *nome do arquivo* e escolha a *opção* *Open with Live Server*. Se não abrir automaticamente, abra seu navegador preferido e digite a URL [http://127.0.0.1:5500/<nome\_do\_seu\_arquivo](http://127.0.0.1:5500/%3cnome_do_seu_arquivo)>. Por exemplo, se o nome do seu arquivo é index.html, vá para <http://127.0.0.1:5500/index.html>. Enquanto o Live Server estiver em execução, você deverá ver sua página.

- **E*xtensão*** ***HTML Play*** - Para executar os arquivos ***HTML*** dentro do ***VS Code***. Com o comando ***Ctrl-alt-h*** ela abre um navegador ***output*** dentro do ***VS Code***, dispensando o ***Chrome***.

**202.1) - Aula-09 - Introdução ao DOM - Document Object Model** *(Modelo de Objetos para Documentos)*

**1.1) - O que é o DOM** - É um acrônimo para Document Object Model *(Modelo de Objetos para Documentos).*

- Basicamente é um conjunto de objetos dentro do navegador que vai dar acesso aos componentes internos do website. O DOM não funciona dentro do NodeJS, ele só está presente quando se está rodando o JavaScript dentro do navegador.

**1.2) - A árvore DOM**

- Algo importante que é preciso conhecer e saber fazer, é a árvore ***DOM*** do site.

- Basicamente a árvore DOM começa da raiz e essa raiz é chamada de ***window*** *(com letra minúscula)*.

- Tudo dentro do JavaScript está dentro deste objeto chamado window *(janela)*. Quando se digita window. *(window ponto)* dentro das tags ***script*** são mostrados todos os objetos dentro do objeto window. Isso vale também para document. *(document ponto)*

**1.3) - Árvore DOM do ex005.html**

- Dentro do objeto ***window*** existem vários outros ***objetos*** também chamado de ***elemento***.

- ***window*** - Tudo dentro do ***JavaScript*** está dentro deste objeto chamado ***window*** *(janela)*.

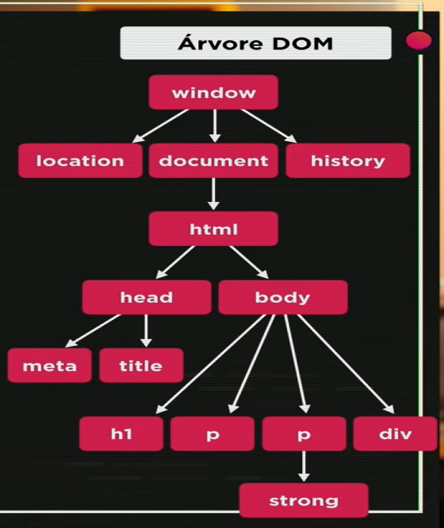
- ***location*** - diz qual é a localização do site, ou seja, qual é a ***URL***, qual é a página atual e a anterior.

- ***document*** - indica o documento atual

- ***html*** - é exatamente a parte ***html*** do site. E dentro do ***html*** tem dois objetos ***filhos*** (*child*).

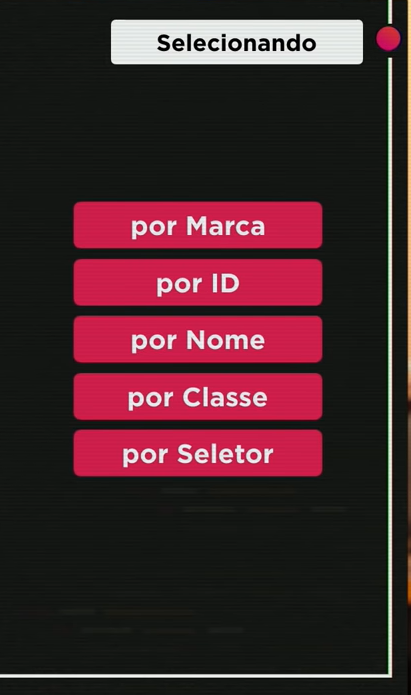
- ***history*** - que guarda de onde você veio e para onde você vai. Isso facilita a navegação.

- Abaixo estão relacionados os objetos da ***árvore DOM*** do ***ex005.html***:



**1.4) - Selecionando Elementos para se navegar dentro da Árvore DOM**

- Existem vários métodos para se selecionar e poder navegar dentro da árvore DOM, abaixo são mostrados cinco métodos de seleçaõ:



- ***Marca*** - Para selecionar elementos pelas ***marcas*** (*tags)*, usa-se o método ***getElementsByTagName()[ ]*** - Dentro dos *parêntesis* e entre *aspas* duplas ou simples, vai a *tag* a selecionar, por exemplo *‘p’* de *parágrafo*, dentro dos *conchetes* vai o *índice* desta *tag* para especificar exatamente qual das *tag* *parágrafo*, dado que esta é uma *tag* que se repete. A primeira ocorrência da *tag* tem o *índice* *zero (0)*, a segunda ocorrência o *índice um (1)* e assim por diante. A *tag* a selecionar é armazenada em uma variável. Ex.:

***- var par1 = window.document.getElementsByTagName('p')[0];***

***- var par2 = window.document.getElementsByTagName("p")[1];***

- ***id*** ***da tag*** ***<div></div>*** - Para selecionar o elemento pelo ***id*** usa-se o método ***getElementById('msg').*** Dentro dos *parêntesis* e entre *aspas* duplas ou simples, vai o nome da tag ***div id.***  Ex.:

***- <div id="msg">Clique em mim</div>***

***- var d0 = window.document.getElementById('msg');***

***- d0.style.background = 'green';*** - Este comando especifica a *cor de fundo* do texto da ***div***.

- **Nome** da tag ***<div></div>*** - Para selecionar o elemento pelo ***nome*** usa-se o método ***getElementsByName('msg2')[0]***. Dentro dos *parêntesis* e entre *aspas* duplas ou simples, vai o nome a selecionar, por exemplo *‘msg2’*, dentro dos *conchetes* vai o *índice* desta ***tag*** ***div*** para especificar exatamente qual das *tags* *div*, dado que esta é uma *tag* que se repete. A primeira ocorrência da *tag* tem o *índice* *zero (0)*, a segunda ocorrência o *índice um (1)* e assim por diante. A *tag* a selecionar é armazenada em uma variável. Ex.:

***- <div name="msg2">Clique de novo</div>***

***- var d1 = window.document.getElementsByName('msg2')[0]***

***- d1.style.background = 'red';*** - Este comando especifica a *cor de fundo* do texto da ***div***.

***- d1.innerText = 'Você clicou de novo...';*** - Este comando define um novo texto em substituição ao texto *‘Clique de novo’.*

- ***Classe*** da tag ***<div></div>*** - Para selecionar o elemento pela ***classe*** usa-se o método ***getElementsByClassName('msg3')[0],*** dentro dos *parêntesis* e entre *aspas* duplas ou simples, vai o nome da classe a selecionar, por exemplo *‘msg3’*, dentro dos *conchetes* vai o *índice* desta ***tag*** ***div*** para especificar exatamente qual das *tags* *div*, dado que esta é uma *tag* que se repete. A primeira ocorrência da *tag* tem o *índice* *zero (0)*, a segunda ocorrência o *índice um (1)* e assim por diante. A *tag* a selecionar é armazenada em uma variável. Ex.:

***- <div class="msg3">Clique mais uma vez</div>***

***- var d1 = window.document.getElementsByClassName('msg3')[0]***

***- d1.style.background = 'blue';*** - Este comando especifica a *cor de fundo* do texto da ***div***.

***- d1.innerText = 'Você clicou outra vez...';*** - Este comando define um novo texto em substituição ao texto *‘Clique mais uma vez’.*

- ***Seletor*** - seleção por CSS, é o recurso mais recente do ECMAScript. Para selecionar o elemento pelo ***'querySelector',*** dentro dos *parêntesis* e entre *aspas* duplas ou simples, vai o texto ***‘div#’*** mais o nome se for uma ‘***div name’***, ou o texto ***‘div.’*** mais o nome se for uma ***‘div class’***

***- var d2 = window.document.querySelector('div#msg');***

***- d2.style.color = 'black';*** - Este comando especifica uma nova cor para o texto da ***div.***

***- var d3 = window.document.querySelector('div.msg3');***

***- d3.style.fontStyle = "italic";*** - Este comando especifica que o estilo do texto da ***div*** deve ser em itálico.

***- d3.style.fontSize = "xx-large";*** - Este comando especifica aumenta o tamanho da fonte do texto da ***div***.

**- Atenção:**

- Quando a div é um ***'name'*** usa-se ***'#' hashtag***.

- Quando a div é uma ***'class'*** usa-se ***'.' ponto.***

**- Código dos itens acima:**

***<!DOCTYPE html>***

***<html lang="pt-br">***

***<head>***

***<meta http-equiv="refresh" content="1;">  <!--Tag para recarregar a tela do navegador automaticamente-->***

***<meta charset="UTF-8">***

***<meta http-equiv="X-UA-Compatible" content="IE=edge">***

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

***<title>ex005.html-Primeiros passos com o DOM</title>***

***<style>***

***body {***

***background-color: rgb(61, 129, 184);***

***color: white;***

***font:normal 15pt Arial;***

***}***

***h2{***

***color:rgb(225, 210, 111)***

***}***

***</style>***

***</head>***

***<body>***

***<H2>Iniciando estudos de Dom</H2>***

***<p>Aqui vai o resultado</p>***

***<p>Aprendendo a usar o <strong>DOM</strong> em JavaScript.<hr></p>***

***<div id="msg">Clique em mim</div>***

***<div name="msg2">Clique de novo</div>***

***<div class="msg3">Clique mais uma vez</div>***

***<hr>***

***<script>***

***//Mostra na tela o caminho do arquivo***

***window.document.write(window.document.URL)***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//Selecionando o elemento pela tag***

***//Variáveis que vão armazerar a tag e o respectivo índice***

***var par0 = window.document.getElementsByTagName('p')[0];***

***var par1 = window.document.getElementsByTagName("p")[1];***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//Mostra uma linha na tela com a tag 'horizontal row'***

***document.write("<hr>");***

***//----------------------------------------------------------------------------------------------------------------------------------***

***/\*O innerText imprime o texto sem as configurações***

***de estilo que está dentro da variável par1.\*/***

***window.document.write('Esta escrito assim: ' + par0.innerText);***

***document.write("<hr>");***

***document.write(par1.innerText);***

***document.write("<hr>");***

***//----------------------------------------------------------------------------------------------------------------------------------***

***/\*O innerHTML imprime o texto COM as configurações***

***de estilo que está dentro da variável par2. Neste***

***caso a palavra DOM esta em NEGRITO (strong)\*/***

***document.write(par0.innerHTML);***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//window.alert(par1.innerText) - //Este alert funciona***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//Muda a cor da variável para vermelho***

***par0.style.color = 'red';***

***//Aumenta o tamanho da fonte***

***par0.style.fontSize = "xx-large";***

***//Muda o estilo da fonte para itálico***

***par0.style.fontStyle = "italic";***

***//Muda a cor da tela***

***var corpo = window.document.body;***

***corpo.style.backgroundColor = "purple";***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//Selecionando o elemento pelo 'id'***

***var d0 = window.document.getElementById('msg');***

***d0.style.background = 'green';***

***//O texto 'Clique em mim' da div 'aviso'  pode ser mudado conforme os dois modos abaixo.***

***//d0.innerText = 'Estou aguardando...';***

***//window.document.getElementById('msg').innerText = 'Estou aguardando...';***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//Selecionando o elemento pelo 'ByName'.***

***var d1 = window.document.getElementsByName('msg2')[0]***

***d1.style.background = 'red';***

***d1.innerText = 'Você clicou de novo...';***

***//----------------------------------------------------------------------------------------------------------------------------------***

***//Selecionando o elemento pelo 'ByClass'.***

***var d1 = window.document.getElementsByClassName('msg3')[0]***

***d1.style.background = 'blue';***

***d1.innerText = 'Você clicou outra vez...';***

***//----------------------------------------------------------------------------------------------------------------------------------***

***/\*Selecionando o elemento pelo 'querySelector'.***

***Quando a div é um 'name' usa-se '#' hashtag.***

***Quando a div é uma 'class' usa-se '.' ponto.\*/***

***var d2 = window.document.querySelector('div#msg');***

***d2.style.color = 'black';***

***var d3 = window.document.querySelector('div.msg3');***

***d3.style.fontStyle = "italic";***

***d3.style.fontSize = "xx-large";;***

***</script>***

***</body>***

***</html>***

**1.5) - Informações complementares obtidas no site abaixo sobre fontsize e fontStyle**

- <https://www.w3schools.com/jsref/prop_style_fontsize.asp>

[❮ Previous](https://www.w3schools.com/jsref/prop_style_fontfamily.asp)[❮ Style Object Reference](https://www.w3schools.com/jsref/dom_obj_style.asp)[Next ❯](https://www.w3schools.com/jsref/prop_style_fontstyle.asp)

Style fontSize Property

Example

Set the font size of an element to "x-large":

document.getElementById("demo").style.fontSize = "x-large";

[Try it Yourself »](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_style_fontsize)

Definition and Usage

The fontSize property sets or returns the font size of the text.

Browser Support

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |
| fontSize | Yes | Yes | Yes | Yes | Yes |

Syntax

Return the fontSize property:

*object*.style.fontSize

Set the fontSize property:

*object*.style.fontSize*= "value*|initial|inherit"

Property Values

|  |  |
| --- | --- |
| **Value** | **Description** |
| xx-small x-small small medium large x-large xx-large | Sets the size of the font to different fixed sizes, from xx-small to xx-large |
| smaller | Decreases the font-size by one relative unit |
| larger | Increases the font-size by one relative unit |
| *length* | Defines the font-size in length units |
| *%* | Sets the font-size to a % of  the parent element's font size |
| initial | Sets this property to its default value. [Read about *initial*](https://www.w3schools.com/cssref/css_initial.asp) |
| inherit | Inherits this property from its parent element. [Read about *inherit*](https://www.w3schools.com/cssref/css_inherit.asp) |

Style fontStyle Property - <https://www.w3schools.com/jsref/prop_style_fontstyle.asp>

[❮ Previous](https://www.w3schools.com/jsref/prop_style_fontsize.asp)[❮ Style Object Reference](https://www.w3schools.com/jsref/dom_obj_style.asp)[Next ❯](https://www.w3schools.com/jsref/prop_style_fontvariant.asp)

Example

Set the font for an element to "italic":

document.getElementById("demo").style.fontStyle = "italic";

[Try it Yourself »](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_style_fontstyle)

Definition and Usage

The fontStyle property sets or returns whether the style of the font is normal, italic or oblique.

Browser Support

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Property |  |  |  |  |  |
| fontStyle | Yes | Yes | Yes | Yes | Yes |

Syntax

Return the fontStyle property:

*object*.style.fontStyle

Set the fontStyle property:

*object*.style.fontStyle = "normal|italic|oblique|initial|inherit"

Property Values

|  |  |
| --- | --- |
| **Value** | **Description** |
| normal | Font is normal. This is default |
| italic | Font is in italic |
| oblique | Font is in oblique |
| initial | Sets this property to its default value. [Read about *initial*](https://www.w3schools.com/cssref/css_initial.asp) |
| inherit | Inherits this property from its parent element. [Read about *inherit*](https://www.w3schools.com/cssref/css_inherit.asp) |

**202.2) - Aula-10 - Eventos DOM - Document Object Model** *(Modelo de Objetos para Documentos)*

- Evento - É tudo do que pode acontecer com qualquer elemento *(div, body, etc.)***.**

- Consulte os sites abaixo para conhecer os eventos:

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

- <https://www.w3schools.com/jsref/dom_obj_event.asp>

**2.1) - O mais comum são eventos de *mouse***

1) ***- mouseenter*** - mover o mouse para dentro do elemento.

2) ***- mousemove*** - mover o mouse no interior do elemento.

3) ***- mousedown*** - clicar e segurar o botão do mouse.

4) ***- mouseup*** - soltar o botão do mouse.

5) ***- click*** - apertar e soltar o bot

6) ***- mouseout*** - mover o mouse para fora do elemento.

**2.2) - Funções**

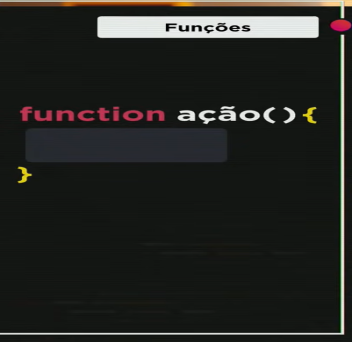
- As *Funções* são um conjunto de linhas de códigos que serão executados somente quando ocorrer um evento.

- Esse conjunto de linhas de código é chamado de bloco o qual é delimitado por um par de chaves {}.

- A sintaxe de uma Função é:

*- function + nome da função + (eventuais parâmetros) { }* **-** O nome da função geralmente é a ação que ela executa.

***- function clique aqui () { };***



- Os eventos podem ser configurados na parte de HTML ou de JavaScritp.

**300) - Siglas**

1) - **Modem** - Serve para **Mod**ular e **dem**odular, ou seja, transformar os ***sinais de*** ***ondas quadradas digitais*** *(zeros e uns)* enviados pelo computador em ***sinais de*** ***onda*** ***senoidal*** *(Modulação),* que são entendidas por uma linha telefônica, antena de celular ou um sistema de TV e vice-versa, ou seja, transformar os sinais de onda senoidal recebidos em sinais de ondas quadradas digitais.

2) - Servidor **DNS** - Domain Name System - Sistema Nome de Domínio - É um servidor que recebe um nome literal tipo *instagram.com* e fornece o código do endereço numérico. Isso acontece toda vês que se acessa um site.

3) - **IP** - Internet Protocol - Para localizar um endereço de IP acesse o site <https://www.iplocation.net/>

4) - **URL** - Uniform Resource Locator - Localizador Uniforme de Recursos

5) - **HTML** - HyperText Markup Language - Linguagem de Marcação para Hipertexto. *(Referida ‘A HTML’ no feminino).* **É uma linguagem focada em conteúdo**: textos, imagens, vídeos, tabelas.

6) - **CSS** - Cascading Style Sheets - Folhas de Estilo em Cascata. *(Referida como ‘As CSS’ no feminino).* Tem o foco em design para tornar os sites mais atrativos: cores, sombras, tamanhos posicionamentos.

7) - **JS - Javascript** - Linguagem de programação que faz as interações do site.

8) - **Front-end** - Client-side - Lado cliente - Desenvolvedor Front-end trabalha com as tecnologias ou liguagens ***HTML***, ***CSS*** e ***Javascript***. São tecnologias que vão rodar do lado do cliente através do navegador *(Chrome, Mozila, Microsoft Edge)*

9) - **Back-end** - Server-side - Lado do servidor - Desenvolvedor Back-end trabalha com as tecnologias ou linguagens tipo:

- ***php*** *(que é uma das mais famosas e aplicadas em servidores web atualmente).*

*-* ***JS-Javascript*** também tem uma versão que roda em servidor e uma das soluções é o ***NODE-JS*** *(não confundir Javascript rodando em servidor com aquele rodando no cliente, pois são tecnologias diferente apesar de ser a mesma linguagem)*

- **C#**

- **Pyton**

- **Ruby**

- **Java**

10) - **NODE-JS** - Node.js é um software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web. A principal característica do Node.js é sua arquitetura assíncrona e orientada por eventos.

11) - **Full stack** - É a união de ***Front-end*** e ***Back-end***. Desenvolvedor ***Full stack*** é aquele que domina as linguagens de ***Front-end*** e o ***Back-end***.

**400) - Dicas e Configurações**

400.1) - Configurar navegador no VS Code

1.1) - **Extensão** **Live Server** - Para executar e atualizar automaticamente os arquivos *HTML*, instale a extensão *Live* *Server*: vá em extensões do *VS* *Code* procure por *Live* *Server* e instale. Em seguida clique com o botão direito no *nome do arquivo* e escolha a *opção* *Open with Live Server*. Se não abrir automaticamente, abra seu navegador preferido e digite a URL [http://127.0.0.1:5500/<nome\_do\_seu\_arquivo](http://127.0.0.1:5500/%3cnome_do_seu_arquivo)>. Por exemplo, se o nome do seu arquivo é index.html, vá para <http://127.0.0.1:5500/index.html>. Enquanto o Live Server estiver em execução, você deverá ver sua página.

1.2) - **E*xtensão*** ***HTML Play*** - Para executar os arquivos ***HTML*** dentro do ***VS Code***. Com o comando ***Ctrl-alt-h*** ela abre um navegador ***output*** dentro do ***VS Code***, dispensando o ***Chrome***.