**CADERNO VIRTUAL**

JavaScript

**MÓDULO A**

AULA 1 – O QUE O JAVASCRIPT É CAPAZ DE FAZER?

***Cliente e Servidor: Conceitos a serem tratados.***

Consideramos os celulares, computadores e demais dispositivos de nosso dia a dia como *clientes*, que utilizam serviços fornecidos por um *servidor* que pode ser acessado, por exemplo, por meio de navegadores (Google Chrome, Microsoft Edge, Mozilla Firefox, Opera etc.). Ao acessarmos um destes navegadores, utiliza-se uma URL (*Uniform Resource Locator* ou *Localizador Uniforme de Recursos*), direcionando o usuário à infraestrutura da internet, que por sua vez, responde para qual *servidor* será desviado. Dentro deste *servidor*, possuidor de um banco de dados requisitados pelo usuário, é feita uma cópia, sendo esta enviada diretamente para o navegador para utilização do *cliente*.

A tecnologia *JavaScript* é considerada *Client-Side*, justamente por atuar, como a própria tradução literal deste conceito, no *Lado do Cliente*, embora possa ser (com ressalvas) utilizada no *Server-Side*.

***Website: Tecnologias.***

Na maioria dos casos, é composto por três tecnologias: a responsável pela estrutura semântica e de conteúdo, o HTML (*Hypertext Markup Language*); a responsável pela estrutura gráfica e de design, as CSS (*Cascading Style Sheets*); por último, a responsável pela estrutura dinâmica e de interações, o JS (*Java Script*).

AULA 2 – COMO CHEGAMOS ATÉ AQUI?

***Linha do Tempo: Evolução do JS.***

Em meados da década de 60, durante a Guerra Fria, o presidente dos EUA, Dwight D. Eisenhower, decretou a criação de uma agência de pesquisa tecnológica, a DARPA (*Defense Advanced Research Projects Agency*), onde uma das tecnologias desenvolvidas por ela foi responsável pela segurança de centros militares. A ideia da agência de pesquisa norte-americana foi criar uma rede de dados que protegesse os dados destes centros militares. Esta rede foi chamada de ARPANET (*Advanced Research Projects Agency Network*). Após o crescimento da ARPANET e a criação de outras pequenas redes ocorreram diferentes modificações, onde em 1970 finalmente chegou-se ao nome que é conhecida atualmente: INTERNET (*International Network*).

Após algumas décadas, em 1993, o britânico Timothy John Berners-Lee foi responsável pela criação de tecnologias que visavam tornar o conteúdo mais interativo através de ligações entre documentos, dando origem a linguagem HTML (*Hypertext Markup Language*), ao protocolo HTTP (*Hypertext Transfer Protocol*) e fundando a WWW (*World Wide Web*) em conjunto com a CERN (*European Organization for Nuclear Research*).

Aula 3 – Dando os primeiros passos:

***Bibliografia: Recomendações de Leitura.***

JavaScript: O Guia Definitivo – David Flanagan, O’Reilly.

JavaScript: Guia do Programador – Maurício Sanny Silva, Novatec.

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

Referência ECMA 262: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/> – ECMA International.

**MÓDULO B**

AULA 5 – VARIÁVEIS E TIPOS PRIMITIVOS:

***Variáveis: atribuições e identificadores.***

Dentro da programação, *variáveis* são áreas dedicadas da memória RAM que guardam valores não constantes. Atribuir um valor a certa variável é chamado atribuição. Exemplo: n1 = 8 ou s1 = “Java Script”. Os nomes das variáveis são chamados de identificadores. Há algumas regras para se nomear identificadores:

* Podem começar com letra, $ ou \_
* Não podem começar com números
* É possível usar letras ou números
* É possível utilizar acentos e símbolos
* Não podem conter espaços
* Não podem ser palavras reservadas

***Tipos Primitivos.***

Number; string; boolean; null; undefined; array; function.

AULA 7 – OPERADORES PT. 1

***Operadores Aritméticos.***

São considerados operadores aritméticos os seguintes: adição (+), subtração (-), multiplicação(\*), divisão (/), resto da divisão inteira (%), exponenciação (\*\*). A precedencia, isto é, a prioridade de “leitura” do programa dos operadores aritméticos se dá da seguinte forma:

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

***Operadores de Atribuição.***

Por outro lado, há os operadores de atribuição, que podem ser por atribuição simples (=) ou por auto-atribuição (var n = 3; n = n + 3; n = 6), onde a última, por sua vez, pode ser simplificada, desta forma: n += 2. A forma mais simplificada de auto-atribuição é o operador de incremento, onde, quando se deseja utilizar uma operação com um valor de 1 para uma operação de adição ou subtração, escreve-se a instrução desta forma: x++, x -- etc.

AULA 8 – OPERADORES PT. 2

***Operadores Relacionais.***

Esse tipo de operador compreende uma relação entre valores, como por exemplo: n1 > n2. São utilizados dentro de uma condição, para criar uma verificação. Exemplo: if n1 >= n2 { fazerAlgo(); }. São operadores relacionais os seguintes: maior que (>), menor que (<), maior ou igual que (>=), menor ou igual que (<=), igual a (==), valor e tipo igual a (===), diferente de (!=). O resultado de uma operação relacional pode ser *true* ou *false*, dependendo da condição. Operadores aritméticos são realizados antes dos relacionais. Há também operadores relacionais de identidade, que correspondem a ‘==’ e ‘===’. Exemplo: 5 == ‘5’ (true); 5 === ‘5’ (false).

***Operadores Lógicos.***

São operadores lógicos: negação (!), conjunção (&&), disjunção (||). Exemplo: !true (false); true && false (false); true || false (true). A ordem de precedência dos operadores, até o momento, se dá da seguinte forma:

* Aritméticos...
* Relacionais...
* !
* &&
* ||

***Operador Ternário.***

Composto por dois símbolos, serve para expressar uma condição seguida de duas ações, dependendo do resultado obtido após a verificação. Exemplo: média>=7 ? “Aprovado” : “Reprovado”.

**MÓDULO C**

AULA 9 – INTRODUÇÃO AO DOM:

***Elementos e Árvore DOM.***

O DOM (document object model) é uma hierarquia de elementos dentro do documento. Qualquer elemento que fique acima de outro na árvore DOM é considerado um *Parent*, ao passo que os elementos que ficam abaixo são considerados *Child*.Exemplo:

* Window
* Body
* h1
* ...

Esses elementos podem ser selecionados por: marca –getElementsByTagName(tag)[0], ID – getElementById(identificação), nome – getElementsByName(nome)[0], classe – getElementsByClassName(classe) e/ou seletor – querySelector(seletor).

AULA 10 – EVENTOS DOM:

***Título.***

Texto.