# ANOTAÇÕES AULA 1

HTML (Hyper Text Markup Language - Linguagem de Marcação de Hipertexto)

HTML é uma linguagem de marcação, pois ela possibilita marcar conteúdos por meio de tags

* O HTML é estático, isto é, não muda. Não podemos fazer nada muito sofisticado com ele.
* Algumas tags não possuem marcação, como o <br>, que serve para pular uma linha

**<br>** = pular linha (br é a abreviação de "break")

**<h1>** = título/destaque/cabeçalho (h de "heading")

**"Ctrl + R"** = recarregar a página

**<a>** = tag HTML de "âncora" para criar links na web

- A tag <a> nos permite sair de uma página e mudar para outra

**href** = h"hiperlink" ref "referência"

linguagem **JavaScript** deve ser colocado dentro de **<script> </script>**

**<meta charset="UTF-8">** = como o navegador deve interpretar as cadeias de caracteres

* meta = elemento capaz de definir qualquer tipo de informação
* charset = indicar o formato de codificação de alguns caracteres que são utilizados dentro do documento
* UTF-8 = é uma maneira de viabilizar a programação e a leitura de documentos em qualquer tipo de idioma

Depurador que diz onde houve um problema no código: **F12** ou **Ctrl+Shift+C**

**alert** = instrução de linguagem que indica que um alerta deverá ser exibido

- texto deve estar entre parênteses e aspas duplas " ", e com um ; (ponto e vírgula) no fim

**Ctrl+S** = salvar

**"String"** = nome comum para "texto"

A extensão de arquivo confere ao Sistema Operacional uma pista de qual programa deve ser usado para abrir o arquivo em questão. Neste caso, em todo arquivo criado deve ser utilizada a extensão .html

* criar arquivos com letras minúsculas
* se houver mais de uma plavra, separar com "\_"
* não utilizar acentos
* ex.: primeiro\_sistema\_que\_fiz.html

**Convenções:** tags HTML devem ser digitadas com letras minúsculas (no JavaScript não funcionará se for digitado em maiúsculo)

Para abrir o arquivo no navegador: **"Ctrl + O"**

# ANOTAÇÕES AULA 2

“alert” recebe o texto dentro de "()"

O mínimo que precisamos no programa para que ele funcione é o seguinte código:

1 <meta charset="UTF-8">

2

3 <script>

4 </script>

A primeira, **<meta charset="UTF-8">** resolve o problema de acentuação e a segunda define o mundo JavaScript, pois é entre a abertura da tag **<script>** e seu fechamento **</script>** que escreveremos os códigos dinâmicos!

Para verificar *onde está ocorrendo o erro no código*: Na barra de menu superior, selecionaremos as opções "Visualizar > Desenvolvedor > Console JavaScript ou se preferir a tecla **F12**".

Para escrever, a partir do mundo JS, um HTML:

document.write: "escrever no documento" - recebe o texto dentro de "("...")"

Utilizamos o document.write porque o JavaScript é uma linguagem dinâmica. Podemos, eventualmente, imprimir estas informações múltiplas vezes.

- javaScript só considera texto se estiver entre aspas.

- Números podem estar fora de aspas, com número é possível fazer operações matemáticas dentro do código.

ex.: document.write(18 + 20)

Se os número estiverem dentro de aspas, não irá somar.

document.write("18" + "20") = "concatenação" (operação de soma envolvendo texto = junta um texto com outro)

o resultado impresso seria: 1820

A operação que deve ocorrer primeiro deve estar entre parênteses (matemática).

ex.:

. document.write((20 + 10 + 30) / 3);

. document.write("A soma das idades é " + (20 + 10 + 30));

Para o resultado da soma fique separado do texto, precisamos dar um espaço dentro do texto entre aspas, depois de "é ".

# ANOTAÇÕES AULA 3

Criação de variáveis:

"var" = variável

"=" = recebe

ex.: var ano = 2016 (variável "ano" recebe o valor "2016")

Automaticamente, todos os lugares em que a palavra "ano" está presente serão interpretados como 2016.

O processo de "raciocínio" do JavaScript será questionar se determinado trecho está entre aspas, e se estiver, é uma string, caso negativo, será feita uma próxima pergunta: trata-se de um número? Caso a resposta seja "não", significa que se trata de uma variável.

A variável começa com letras minúsculas. Se houver duas palavras a segunda inicia com letra maiúscula (camelCase).

Em qualquer momento, no próprio código, podemos indicar um novo valor para esta mesma variável e tudo que estiver abaixo dela deverá considerar a variável desta forma. O prefixo var só é utilizado ao declararmos a variável pela primeira vez, então não é necessário utilizá-lo ao definirmos um novo valor

Para arredondar resultados trabalharemos com a matemática (Math). Em seguida, chamaremos a função round (do inglês, "arredondar"): = Math.round(variável)

Com o conteúdo entre parênteses de document.write(), definimos o que será exibido na tela, e com Math.round definiremos qual valor será arredondado

# Anotações Aula 4

Para pular linha:

document.write("<br>")

document.write("<br><br><br><br>")

Todas as funções que criarmos estarão nas primeiras instruções da tag <script>, ou seja, declararemos no início do código.

Toda função criada se inicia com o termo "function", em seguida temos o nome da função, terminando com parêntese:

ex.: function pulaLinha(){

document.write("<br>")

}

Sempre dentro de um bloco utilizaremos a tecla "Tab" para criar um espaçamento na linha e, assim, o código ficará indentado e teremos uma melhor noção de hierarquia.

Toda função pode englobar uma ou mais instruções.

Para executar a função, é necessária a utilização dos parênteses: ex. pulaLinha()

Função é sempre um verbo

O nome de uma função é tão importante quanto o código que ela executa, para deixar claro o que ela faz.

Colocar trechos em comentários, por meio de barras duplas (//). O programa ignorará a referida linha, que continuará presente para fins didáticos, para lembrarmos o que havia antes.

Função com parâmetro:

function mostra(frase) {

document.write(frase);

}

O parâmetro da função é considerado como se fosse uma variável.

Uma função pode ter como instrução a chamada de outra função.

ex.: function mostra(frase) {

document.write(frase);

pulaLinha();

}

Para "chamar" a função, precisamos definir o parametro que deve ser exibido.7

ex.: mostra("Calopsita!");

<hr> = imprime uma linha na tela