**Livros e Manuais:**

1. Javascript: O Guia Definitivo – David Flanagan
2. JavaScripct - Guia do Programador – Maujor – *Pouco desatualizado*
3. Guia de Referência da Mozilla –
   1. <https://developer.mozilla.org/pt-BR/>
   2. https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference
4. Guia de Referência da ECMA
   1. https://www.ecma-international.org/publications-and-standards/standards/ecma-262/

**Projetos Próprios:**

**Aula 4 – Criando o seu primeiro script**

*- HLTM: Corpo Texto*

*- CSS: Design*

*- JS: Integração*

- Pelo Visual Studio Code é possível criar pastas e arquivos, porém é necessário selecionar uma pasta inicial anteriormente.

- Colocar *.html* no final do arquivo para funcionar no navegador.

- Abrir o arquivo de navegador e atualizar para ver o código funcionando.

Código

- Digitar *html* no corpo sugere algumas opções

- Digitar *html:5*  traz o código de html 5 já todo escrito

- Bolinha no título do programa quer dizer que falta atualizar

*Html*

***<head>*** *:*

***<title>*** *: aparece no título da página*

***<body>*** *: Corpo da página*

***<h1> :*** *Título de nível 1*

***<p> :*** *Parágrafo*

*CSS*

*Todo final de marcação tem ;*

***<style> :*** *feito na área de cabeça*

***Body{}*** *: Estilo do body*

***Background-color:*** *muda cor do fundo*

***Color:*** *muda cor da letra*

***Font:*** *muda fonte conforme regra “normal”, ”tamanho” e “tipo da fonte”*

*Javascript*

***<script>:*** *Começa a criar o script*

***Window.alert(‘’):*** *Alerta com escrito entre aspas*

**Aula 5 – Variáveis e Tipos Primitivos**

- Sempre quando temos 1 “=”, o devemos chamar de recebe, portanto se ***vaga a1 = carro*** *então* ***vaga a1 recebe carro.***

- Podemos tirar a informação da variável utilizando **vaga a1 = null.**

- Exemplos de variáveis

var n1 = 5 var s1 = “Javascript”

var n2 = 8.5 var s2 = ‘Curso em Vídeo’

var n3 = 15 var s3 = `Guanabar`

Obs: “”,’’ e `` são diferentes

- O nome de cada variável possuem alguns identificadores (regras):

a) Podem começar com letra, $ ou \_

b) Não podem começar com números

c) É possível usar letras ou números

d) É possível usar acentos e símbolos

e) Não podem conter espaços

f) Não podem ser palavras reservadas

- Dicas para identificadores

a) Maiúsculas e minúsculas fazem diferença

b) Tente escolher nomes coerentes para as variáveis

c) evite se tornar um ‘programador alfabeto’ ou um ‘programador contador’

**VARIÁVEIS SÃO PARA GUARDAR DADOS**

Tipos primitivos primordiais

- Number (5, 18, -12, 0.5, -15.9, 3.14, 8.0): Não tem diferenciação de Reais e Inteiros

- String (“Google”, ‘Javascript’, `Maria`): Cadeia de Caracteres, o CPF é uma string

- Boolean: true, false

Outros tipos primitivos

1. Number:

a) Infinity

b) NaN (Not a Number)

2. Null

3. Undefined

4. Object

a) Array

5. Function

* Quando definímos as variáveis, isso não quer dizer que definimos o tipo primitivo da variável para sempre.
* Para saber o tipo primitivo da variável basta usar o comando *typeoff* ‘**variável’**
* Para o JS o ***null*** é um objeto

**Aula 6 – Tratamento de Dados**

- É possível utilizar copiar e colar no visual basic studio para copiar arquivos e pastas

- Para receber o valor escrito pelo prompt, basta direcionar o comando para a variável

- Para chamar as variáveis em um alert, sempre precisamos coloca-las fora das aspas

- + junto com as variáveis, tem função de concatenar

- O + pode ser para concatenação e para somar

- Para o JS entender, precisamos explicar que é a variável

- No prompt o JS recebe um valor de STRING e por isso precisamos converter

- Para converter podemos fazer de algumas formas:

a) Number.parseint(n) – conversão para inteiro

b) Number.parseFloat(n) – conversão para real

c) Number (n) – conversão para número, entende a necessidade do número

- Sugestão é converter depois do JS ler a resposta

- Para abrir e fechar parênteses de forma rápida, basta selecionar tudo e abrir parênteses

- Podemos converter de número para string também:

a) String (n) – gerar uma string

b) n.toString() – jogar para string

- Para escrever de uma forma mais fácil e organizada, podemos utilizar a template string utilizando as crases (``) e o ${} para chamar as variáveis

Outros comandos para formatar strings:

1. s.length – quantos caracteres a string tem
2. s.toUpperCase() – tudo para ‘maiúsculas
3. s.toLowerCase() – tudo para ‘minúsculas’

- Em html <Strong> deixa em negrito e <br/< quebra linhas

Outros comandos para formatar números

1. n1.toFixed(2) – deixa o número com 2 casas decimais
2. n1.toFixed(2).replace(‘.’,’,’) – substitui o . com por virgula
3. n1.toLocaleString('pt-BR', {style: 'currency', currency:'BRL'}) – muda para Reais

**Aula 7 – Operadores (Parte 1)**

Tipo de operadores:

1. aritméticos;
2. atribuição;
3. relacionais;
4. lógicos;
5. ternário.
6. Operadores aritméticos

Soma (+), subtração (-), multiplicação (\*), divisão real (/), resto da divisão inteira (%), potência (\*\*).

Ordem de precedência

()

\*\*

\*/%

+-

1. Auto-atribuição
2. Var n = 3;
3. n = n + 4 ou n += 4
4. n = n – 5 ou n -= 5
5. n = n \* 4 ou n \*= 4
6. n = n / 2 ou n/= 2
7. n = n \*\* 2 ou n \*\*= 2
8. n = n % 5 ou n %= 5
9. Operadores de incremento
10. Var x = 5
11. x = x + 1 ou x += 1 ou x – (mostra e depois incrementa)
12. x = x – 1 ou x -= 1 x++ - (mostra e depois incrementa)
13. – x (incrementa e depois mostra)
14. ++ (incrementa e depois mostra

**Aula 8 – Operadores (Parte 2)**

1. Relacionais – trazem valores bouleanos (true,false)
   1. > (maior)
   2. < (menor)
   3. >= (maior ou igual)
   4. <= (menor ou igual)
   5. == (igual)
   6. != (diferente)
2. Identidade

Quando colocamos:

* 1. 5 == 5 (true)
  2. 5 == ‘5’ (true)

Para comparar se são do mesmo tipo:

* 1. 5 === ‘5’ (false)
  2. 5 === 5 (true)

3. Lógicos

1. ! (negação) – NÃO
2. && (conjunção) - E
3. || (disjunção) – OU

Precedência

() \*\* /

< > >=

!

&&

||

1. Ternário – é o SE(IF)

**Aula 9 – Introdução ao DOM**

*DOM – Document Object Model*

Conjunto de objetos dentro navegador que da acesso aos componentes internos do website

*Árvore DOM*

window

1. location – qual é a localização do site (url)
   1. html
      1. head
         1. meta
         2. title
      2. body
         1. h1
         2. p
            1. strong
         3. p
         4. div
2. document – documento atual
3. history – de onde veio, para onde vai

Selecionando elementos

* 1. por Marca – getElementsByTagName() – body, p, body
  2. por ID – getElementByID() – preenchido dentro do elemento (#)
  3. por Nome – getElementsByName() - preenchido dentro do elemento (.)
  4. por Classe – getElementsByClassName() - preenchido dentro do elemento
  5. por Seletor – querySelector() ou querySelectorAll()