# Identificadores (variáveis)

Podem começar com letra, $ ou \_

Não podem começar com números

É possível usar letras e números

É possível usar acentos e símbolos

Não podem conter espaços

Não podem ser palavras reservadas

# Dicas

Maiúsculas e minúsculas fazem diferença

Usar nomes coerentes para as variáveis

# Conversões de tipos

parseInt( var ) 🡪 converte para número inteiro

parseFloat( var ) 🡪 converte para número com ponto flutuante

String( var ) 🡪 converte para String

var.toString 🡪 converte para String

Number( var ) 🡪 converte para tipo numérico, o JS que irá definir o seu tipo primitivo de acordo com a necessidade.

# Formando String (template string)

Criamos a nossa “frase” entre crases.

Exemplo:

Var s = ‘JavaScript’

`Eu estou aprendendo ${ s }`

Dessa forma é mais simples criar as frases sem ficar realizando uma série de concatenações.

s.length 🡪 Quantidade de caracteres a string tem

s.toUpperCase() 🡪 tudo para letras maiúsculas

s.toLowerCase() 🡪 tudo para minúsculas

Número com 2 casas decimais

n1.toFixed( 2 )

n1.toLocaleString(‘pt-BR’, { style: ‘currency’, currency: ‘BRL’ })

# Operadores Básicos

## Aritméticos

|  |  |
| --- | --- |
| + | Adição |
| - | Subtração |
| \* | Multiplicação |
| / | Divisão |
| % | Resto da Divisão |
| \*\* | Potenciação |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | + | 2 | 7 |
| 5 | - | 2 | 3 |
| 5 | \* | 2 | 10 |
| 5 | / | 2 | 2.5 |
| 5 | % | 2 | 1 |
| 5 | \*\* | 2 | 25 |

## Atribuição

= 🡪 atribuição simples

n += 4 🡪 auto atribuição

n ++ 🡪 incremento

## Relacionais

|  |  |
| --- | --- |
| > | Maior |
| < | Menor |
| >= | Maior ou Igual |
| <= | Menor ou Igual |
| == | Igual |
| != | Diferente |
| === | Idêntico |

Sempre retorna um valor booleano, ou seja, *true* ou *false*.

Idêntico também verifica se os tipos das variáveis são iguais.

## Lógicos

|  |  |
| --- | --- |
| ! | Negação |
| && | Conjunção ( e ) |
| || | Disjunção ( ou ) |

## Ternário

Teste ? true : false

Exemplo:

Média >= 7.0 ? “Aprovado” : “Reprovado”

# Document Object Model ( DOM )

* Realizar a instalação do Watch in Chrome no VSCode. Realizar a instalação da Extensão no Google Chrome.
  + Agora, quando alterarmos o arquivo em HTML, automaticamente as alterações já serão realizadas no navegador, sem a necessidade de salvar o arquivo e depois atualizar a página.
  + CTRL + SHIFT + P 🡪 digitar watch in chrome.
* Instalar a extensão do Node Exec no VSCode
  + para conseguirmos rodar os comandos do node com a tecla F8.

DOM 🡪 um conjunto de objetos dentro do navegador. Só funciona dentro do navegador.

## Árvore DOM

window (janela)

locaton – localização do site ( URL )

document – documento atual

history - de onde veio e para onde vai (caminhos de URL)

Dentro do document, existe o arquivo HTML.

Dentro de HTML, temos head e body.

Logo, HTML é parente de body. E HTML é child de document.

Por ter essa “árvore genealógica” é que chamamos de Árvore DOM

Por esse motivo, quando vamos usar alguma funcionalidade do JavaScript, usamos por exemplo:

window.document.

Estamos acessando a janela ( window ) e em seguida o seu filho document.

## Selecionando

Existem diversas formas de navegarmos entre os elementos da nossa página

Por marca

getElementsByTagName( )

Selecionamos mais de um objeto por vez

Por ID

getElementById( )

Por nome

getElementsByName( )

Por classe

getElementsByClassName( )

Por Seletor

querySelector( )

querySelectorAll( )

Colocamos o elemento e em seguida a classe ou id

ex:

document.querySelector(‘div#msg’) 🡪 nesse caso, pegaremos a DIV e tem o ID msg

document.querySelector(‘div.msg’) 🡪 nesse caso, pegaremos a DIV e tem a CLASS msg

Esse é o método mais recente no momento.

# Eventos DOM

Existem diversos eventos que podem acontecer em nossa página, como por exemplo, o evento de passar o mouse sobre uma div.

<https://developer.mozilla.org/pt-BR/docs/Web/Events>

A parte de eventos podem ser criados diretamente no HTML ou no script do JavaScript.

Dentro do HTML:

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

Dentro do JS, criamos um escutador e informamos o evento que desejamos ouvir e qual a função que deverá ser chamada.

    <script>

        var area = document.getElementById('area')

        area.addEventListener('click', clicar)

        area.addEventListener('mouseenter', entrar)

        area.addEventListener('mouseout', sair)