**Roteiro de aula – Javascript**

* Introdução
* Variáveis e operações aritméticas
* Funções
* Operadores lógicos e relacionais
* Condicionais e condicionais ternários
* Estrutura de repetição
* Complementos
* Exercícios

**Introdução**

* Executar scripts do lado cliente (navegador). A requisição não precisa chegar no servidor para devolver algumas respostas.
* O JS permite fazer alterações dinâmicas na árvore de elementos em tela (DOM).
* Popularidade do JS, versões e frameworks.
* JS pode ser escrito direto no console ou através do HTML. Vamos começar usando o HTML.
* Tag <script> no final do <body> para garantir que todos os elementos do DOM já tenham sido renderizados quando a gente precisar usá-los dentro do JS.

**aula01/index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Curso de Javascript</title>

</head>

<body>

    <script>

        console.log("Hello, World!");

    </script>

</body>

</html>

**Variáveis**

* Não há tipagem estática, ou seja, não precisa definir o tipo da variável. Além disso, eu sempre posso mudar o valor da variável sem precisar converter o tipo do dado (tipagem dinâmica).

<script>

        var nome = "Luana";

        nome = 30;

        console.log(nome);

 </script>

* Tipos de dados  em Javascript
  + String
  + Number
  + Boolean
  + Array
  + Object
  + Function (veremos mais a frente)

var nome = "Luana"; // string

        var idade = 31; // number

        var peso = 75.5; // number

        var matriculado = true; // boolen

// var nome = “Luana”, idade = 31, peso = 75.5; 🡪 Pode omitir o *var*

        var alunos = ["Luana", "Fernando", "Paula"]; // array

        var aluno = { // object

            nome: "Luana",

            idade: 31,

            peso: 75.5,

            matriculado: true

        }

* Exibição de arrays e objetos

console.log(alunos);

        console.log(alunos[0]);

        console.log(alunos[1]);

        console.log(aluno);

        console.log(aluno.nome);

        console.log(aluno["peso"]);

**Operações aritméticas**

* O JS sempre retorna a divisão exata considerando as casas decimais, pois o tipo numérico dele é number e não int, double, float etc.

var x = 10, y = 5;

        var soma = x + y;

        var subtracao = x - y;

        var multiplicacao = x \* y;

        var divisao = x / y;

        var modulo = x % y;

// incrementar (o mesmo vale para todos os operadores matemáticos).

x = x + 1;

x += 1;

x++; //apenas quando for incrementar 1.

**Funções**

function soma(n1, n2){

            var resultado = n1 + n2; //variável resultado em escopo local

            return resultado;

        }

        var resultado = soma(1,2); // outra variável resultado

        console.log(resultado); // console.log(soma(1,2);

**Operadores lógicos e relacionais**

* 5 > 7 🡪 false
* 1 == “1” 🡪 true
* 1 === “1” 🡪 false
* 5 > 7 && 1 == “1” 🡪 false
* 5 < 7 || 1 === “1” 🡪 true
* ! 5 > 7 🡪 true

**Condicionais e condicionais ternários**

* If e else

var sexo = ‘F’

        if (sexo == ‘M’) {

            console.log("Masculino");

        } else {

            console.log("Feminino");

        }

* Elseif

var sexo = ‘F’

        if (sexo == ‘F’){

            console.log("Feminino");

        } else if (sexo == ‘M’) {

            console.log("Masculino");

        } else {

            console.log("Outro");

        }

* If ternário

  function retornaSexo(sexo){

           return (sexo == 'M') ? 'Masculino' : 'Feminino';

        }

        var resultado = retornaSexo('F');

        console.log(resultado);

* Switch

function retornaSexo(sexo){

            switch (sexo) {

                case 'M':

                    return 'Masculino';

                case 'F':

                    return 'Ferminino';

                default:

                    return 'Outro';

            }

        }

        var resultado = retornaSexo('F');

        console.log(resultado);

**Estruturas de repetição**

* While

var i = 1;

       while(i <= 10){

           console.log(i);

           i++;

       }

* For

for (var i = 1; i <=10; i++) {

console.log(i);

}

**Complementos**

* Interlavo : Define um intervalo de tempo para uma função ser executada.
  + exibeAlgo() 🡪 Com parênteses, executa a função.
  + exibeAlgo 🡪 Sem parênteses, referencia a função, ou seja, é um valor passado como parâmetro.
  + Tempo é medido em milissegundos.

function exibeAlgo(){  console.log("Hello, World!"); }

       setInterval(exibeAlgo, 1000);

* Timeout: Inicia uma função a partir de um determinado tempo.

function exibeAlgo(){  console.log("Hello, World!");  }

       setTimeout(exibeAlgo, 5000);