

Relembrando Programação!!!

1 - O que é uma Linguagem de Programação?

Linguagem de programação é um conjunto de códigos e regras que transmitem instruções aos computadores, de maneira que eles executem determinadas funções.

As linguagens de programação servem para possibilitar a comunicação entre máquinas e humanos. Elas definem as regras sintáticas e semânticas que padronizam a comunicação e permitem trocar informações com computadores.

Seu principal objetivo é fazer a transformação de ideias humanas em linguagem de máquina, ser a interface (ligação) entre homem e máquina.

2 - Quais são os tipos de Linguagem de Programação?

Basicamente temos dois níveis de Linguagens, Linguagem de Baixo Nível e Linguagem de Alto Nível.

Linguagens de Baixo Nível

As linguagens de baixo nível são as mais próximas do hardware do computador. Elas são compostas por instruções que são facilmente compreendidas pelo computador, mas que são difíceis para os humanos entenderem. Isso ocorre porque as linguagens de baixo nível são muito específicas e descrevem detalhes técnicos do hardware.

As linguagens de baixo nível são usadas principalmente em aplicações que exigem alto desempenho e controle preciso do hardware. Exemplos de linguagens de baixo nível incluem:

- Linguagem de máquina: é a linguagem mais básica que um computador pode entender. Ela é composta por sequências de números binários que representam instruções de hardware.
- Linguagem de montagem: é uma linguagem que é mais fácil de entender do que a linguagem de máquina. Ela usa mnemônicos, que são palavras ou abreviações para as instruções de hardware.

Linguagens de alto nível

As linguagens de alto nível são mais abstratas e fáceis de entender para os humanos. Elas se concentram em conceitos de alto nível, como variáveis, loops e funções.

As linguagens de alto nível são usadas para a maioria das aplicações de programação. Elas são mais fáceis de aprender e usar do que as linguagens de baixo nível, e permitem que os programadores se concentrem no algoritmo do programa, deixando os detalhes técnicos para a linguagem de programação.

Exemplos de linguagens de alto nível incluem:

- Python
- Java
- PHP
- JavaScript

E quanto ao funcionamento das linguagens?

Linguagens de programação podem ser divididas em duas categorias principais:

- ✓ Linguagens compiladas.
- ✓ Linguagens interpretadas

A principal diferença entre elas está no processo de tradução do código fonte para código executável.

A escolha entre linguagens interpretadas e compiladas dependerá das necessidades do projeto e das preferências do desenvolvedor.

Linguagens interpretadas são mais fáceis de aprender e de prototipar, enquanto linguagens compiladas oferecem maior performance e segurança.

<p>Prototipar é uma técnica que consiste em desenvolver o protótipo de algum produto. Ou seja, uma forma de visualizar a sua ideia antes de ela ser entregue definitivamente ao mercado.</p>

Linguagens Compiladas

Já as linguagens compiladas são aquelas em que o código fonte é compilado antes da execução, gerando um arquivo executável que pode ser executado diretamente pelo sistema operacional.

Exemplos de linguagens compiladas são C, C++, Java e Rust.

O processo de compilação pode levar mais tempo, mas o código resultante tende a ser mais rápido e eficiente em termos de recursos

Linguagens Interpretadas

Linguagens interpretadas são aquelas em que o código fonte é executado diretamente por um interpretador, que analisa o código linha por linha e executa as instruções em tempo real.

Exemplos de linguagens interpretadas são Python, Ruby, JavaScript e PHP.

Como o código fonte é executado diretamente, não é necessário um processo de compilação prévio e o código pode ser modificado e testado com mais rapidez e facilidade.

JAVASCRIPT

O JavaScript é uma linguagem que está praticamente em todos os lugares por causa de sua flexibilidade e adaptabilidade à diversos ambientes. Podemos utilizar o JavaScript no front-end através de frameworks e bibliotecas como o Angular e o React; no back-end, através do Node.js e do Deno; no desenvolvimento mobile, através de frameworks como Ionic e React Native; e até mesmo para o desenvolvimento de games. Esse já é um bom motivo para dedicar um tempo de estudo a ele.

Mercado de trabalho

O JavaScript é uma das linguagens mais versáteis na atualidade, o que possibilita a sua utilização em uma série de ambientes diferentes, como vimos na introdução. Isso se reflete na quantidade de vagas disponíveis para desenvolvedores JavaScript no mercado atualmente

Multiplataforma

Uma das características mais interessantes do JavaScript é a sua versatilidade, o que o torna uma linguagem que pode ser executada em uma infinidade de ambientes, passando pelo front-end e indo até o desenvolvimento de jogos e aplicações para IoT.

Facilidade para quem está começando

O JavaScript é uma linguagem extremamente amigável para quem está começando com desenvolvimento de software: não há a necessidade de ferramentas muito específicas ou pesadas para que seja possível desenvolver algo com JavaScript, bem como não há a necessidade de um computador com hardware muito avançado.

Muitas opções de frameworks/bibliotecas

A quantidade de soluções, frameworks e bibliotecas existentes para JavaScript é muito grande. Tanto que até brincam que a cada dia, um framework ou biblioteca JavaScript é criado.

Comunidade

A quantidade de pacotes, bibliotecas e frameworks open source para JavaScript existentes já mostra o tamanho da comunidade JavaScript e o quão ativa ela é: trata-se de uma das comunidades de desenvolvimento mais ativas.

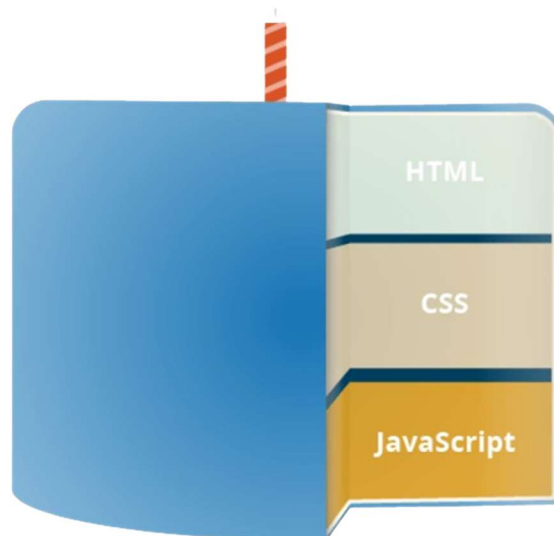
JavaScript é uma linguagem de programação que permite a você implementar itens complexos em páginas web – toda vez que uma página da web faz mais do que simplesmente mostrar a você informação estática – mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados, etc. – você pode apostar que o JavaScript provavelmente está envolvido.

JavaScript é uma linguagem de programação de alto nível que foi criada em 1995 por Brendan Eich, então um engenheiro da Netscape. A ideia inicial era criar uma linguagem de programação que pudesse ser executada no navegador e que permitisse aos desenvolvedores adicionar interatividade e dinamismo às páginas da web.

A primeira versão do JavaScript foi lançada junto com o navegador Netscape Navigator 2.0 em setembro de 1995. A linguagem foi rapidamente adotada por outros navegadores, como o Internet Explorer da Microsoft, e se tornou uma das principais tecnologias utilizadas na criação de páginas web dinâmicas.

Ao longo dos anos, o JavaScript evoluiu e se tornou uma linguagem cada vez mais poderosa, capaz de realizar tarefas complexas tanto no navegador quanto no servidor.

O JavaScript é considerada a terceira camada do bolo das tecnologias padrões da web, duas primeiras são (HTML e CSS).

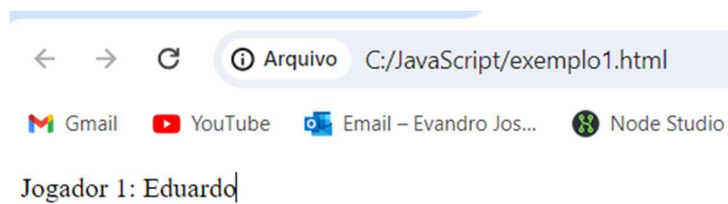


- HTML é a linguagem de marcação que nós usamos para estruturar e dar significado para o nosso conteúdo web. Por exemplo, definindo parágrafos, cabeçalhos, tabelas de conteúdo, ou inserindo imagens e vídeos na página.
- CSS é uma linguagem de regras de estilo que nós usamos para aplicar estilo ao nosso conteúdo HTML. Por exemplo, definindo cores de fundo e fontes, e posicionando nosso conteúdo em múltiplas colunas.
- JavaScript é uma linguagem de programação que permite a você criar conteúdo que se atualiza dinamicamente, controlar multimídias, imagens animadas, e tudo o mais que há de interessante. Ok, não tudo, mas é maravilhoso o que você pode efetuar com algumas linhas de código JavaScript.

Vamos ver um exemplo prático.

exemplo1.html → Somente HTML → Escreva HTML:5

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Exemplo 1</title>
</head>
<body>
  <p> Jogador 1: Eduardo
</body>
</html>
```

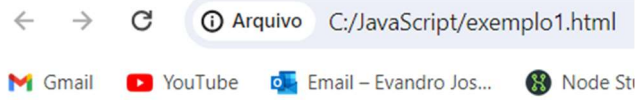


estilo.css → Criando o arquivo CSS

```
p {
  font-family: "helvetica neue", helvetica, sans-serif;
  font-weight: bold;
  letter-spacing: 1px;
  text-transform: uppercase;
  text-align: center;
  border: 2px solid rgba(0, 0, 200, 0.6);
  background: rgba(0, 0, 200, 0.3);
  color: rgba(0, 0, 200, 0.6);
  box-shadow: 1px 1px 2px rgba(0, 0, 200, 0.4);
  border-radius: 20px;
  padding: 5px 10px;
  display: inline-block;
  cursor: pointer;
}
```

Importando o arquivo CSS

```
<html>
  <head>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <p>Jogador número 01:</p>
  </body>
</html>
```

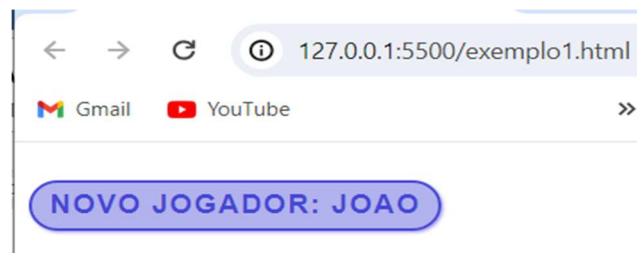
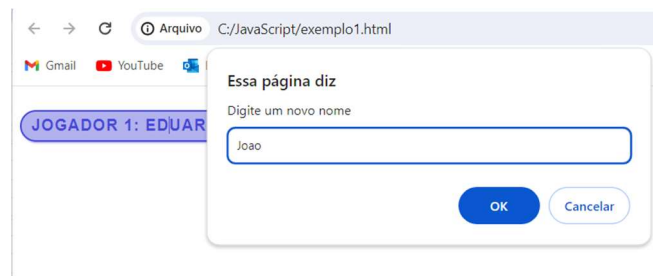


JOGADOR 1: EDUARDO

script.js → Criando JavaScript

```
mudartexto = document.querySelector("p");
mudartexto.addEventListener("click", atualizarTexto);
function atualizarTexto() {
    var texto = prompt("Digite um novo nome ");
    mudartexto.textContent = "Novo Jogador: " + texto;
}
```

```
<html>
  <head>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <p>Jogador número 01:</p>
  </body>
  <script src="script.js"></script>
</html>
```



Configurações Necessárias.

- 1 - VSCODE → www.evandrojv.com.br
- 2 - Instalar a extensão Live Server.

MÃOS NA MASSA

Comando de saída → São os comando que vamos usar para mostrar a saída do que programamos. Temos dois básicos.
console.log() → E que vamos observar, que sua impressão de saída não sai diretamente no navegador, temos que depurar o código, para ver os resultados. E temos também o comando **window.alert()** → este vai gerar um caixa de texto com a mensagem desejada.
Vamos fazer um exemplo, vamos colocar nosso Java script diretamente nas páginas nestes primeiros exemplos.
document.write() → Este método vai gravar o resultado na página.

1 - **Crie uma pasta para colocar seus exemplos, ATENÇÃO, sempre salve num PENDRIVE, as maquinas geralmente serão limpas ou até mesmo formatadas.**

aula01.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula 01</title>
  <script>
    console.log("Ola Mundo");
    console.log("Nova Linha");
    window.alert("Ola Mundo");
  </script>
</head>
<body>
</body>
</html>
```

2 - **Vamos agora criar entrada de dados, vamos usar 1 - Usando a função prompt**

aula02.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula 02</title>
  <script>
    texto = prompt("Digite um texto");
    window.alert("Você' digitou: " + texto);
  </script>
</head>
</body>
</html>
```


Variáveis

aula03.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula 03</title>
</head>
<script>
var a = 5;
var b = 10;

if (a === 5) {
  let a = 4; // O escopo é dentro do bloco if
  var b = 1; // O escopo é dentro da função

  console.log(a); // 4
  console.log(b); // 1
}

console.log(a); // 5
console.log(b); // 1
</script>
<body>

</body>
</html>
```

OPERADORES

= → Operador de Atribuição

Operadores de Aritméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
**	Exponencial
%	Módulo
++	Incrementar
--	Decrementar

```

aula04.html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula 04</title>
  <script>
    let num1 =10;
    let num2 = 5;
    soma = num1 + num2;
    divisao = num1 / num2;
    restoDivisao = num1 % num2;
    document.write("Soma "+ soma+ "<br>");
    document.write("divisão "+ divisao+"<br>");
    document.write("Resto da divisão "+ restoDivisao+"<br>");
  </script>
</head>
<body>
</html>

```

Operadores Relacionais/Comparação

Operador	Descrição
==	Igual a
===	Mesmo valor e tipo
!=	Diferente
!==	Diferente em valor e tipo
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a

```

aula05.html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula 05</title>
  <script>
    num = 5;
    num1 = 10;
    num2 = "10";
    document.write(num > 10 ); //false
    document.write("<br>");
    document.write(num1 == num2); //true
    document.write("<br>");
    document.write(num1 === num2); //false
  </script>;
</head>
</body>
</html>

```

OPERADORES LÓGICOS

Operador	Descrição
&&	'e' lógico
	'ou' lógico
!	'não' lógico

```
aula06.html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script>
    num = 5;
    num1 = 10;
    soma = num + num1;
    if(num > num1 && soma >= 15){
      document.write("As duas condições são verdadeiras");
    }else{
      document.write("Uma condição é falsa");
    }
  </script>;
</head>
<body>
</body>
</html>
```

Aula06.html → Usando not !

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <script>
    num = 5;
    num1 = 10;
    soma = num + num1;
    if (!(num > num1 && soma >= 15)){
      document.write("As duas condições são verdadeiras");
    }else{
      document.write("Uma condição é falsa");
    }
  </script>;
</head>
<body>
</body>
</html>
```

OPERADOR TERNARIO

(condição) ? retorno_se_verdadeiro : retorno_se_falso

aula07.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script>
    num = 5;
    num1 = 10;
    (num > num1)?document.write("Primeiro maior"):document.write("Segundo
maior")
  </script>
</head>
<body>
</body>
</html>
```

Exercícios

- 1 - Crie um script em Javascript onde será pedido via "prompt" nome e ano de nascimento o programa através de **window.alert** deverá retornar nome e idade desta pessoa.
- 2 - Crie um script em Javascript onde será pedido via "prompt" que usuário digite um número qualquer, usando "Operador Ternário" deverá avisar se o mesmo é par ou ímpar. **Use o console.log para apresentar o resultado**
- 3 - Crie um script Javascript onde está solicitado a digitação de 3 números, o programa deverá mostrar a soma destes, e se o mesmo é múltiplo de 3, use **document.write para apresentar o resultado**