# **Curso Javascript completo Es6 – 37h**

Professor André Rafael

## **Definições**

Linguagem **interpretada**, ou seja, tudo que escrever num arquivo .js e abrir num browser já vai funcionar, diferente de uma compilada, que primeiro precisa passar num programa pra depois ela funcionar.

Aplicações:

**-** **Front-end:** manipulação do DOM (a tela em si, rolagem de telas), comunicação assíncrona com o back end (a página da Origamid por exemplo, ao trocar de curso, ele não carrega tudo de novo, apenas troca as telas) - e mais;

**- Jogos** – Dá para fazer jogos simples com JS;

**- Back-end:** linguagem back end, se comunicar com o servidor, banco de dados, arquivos etc.

## **Programas a serem utilizados**

Node.JS, VS Code, Git.

Dentro do VSCode, instalar extensões:

**- Origamid:** tema criado pelo professor para facilitar a visualização do código, destacando funções, palavras-chaves, etc.

**- Live Server;**

**- Color Highlight;**

**- Prettier – Code formatter:** arruma o código para que fique os espaços do lado esquerdo ao salvar no VSCode;

**- Fonte IBM Plex Mono (boa para códigos).**

## **Comentários no JS**

Basta utilizar // texto em 1 linha só, ou /\* texto \*/ para várias linhas.

## **Variáveis**

Para chamar o código JS no html, adicionar a tag **script src** e colocar onde está o arquivo (normalmente na mesma pasta):

<body>

  <script *src*="script.js"></script>

</body>

Depois de digitar qualquer coisa, ir em **Open with Live Server,** já irá abrir direto no navegador, depois é só ir em Inspecionar e qualquer alteração feita no código no VSCode, conseguiremos ver lá no mesmo momento:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Bom, **variáveis** são responsáveis por guardar dados na memória. Sempre que eu quiser trazer meu nome várias vezes na tela, basta que eu o coloque dentro de uma variável, e depois só trazer o nome daquela variável. Elas evitam repetições. 3 formas de se trazer uma variável: **var, let, const,** cada uma com suas características que veremos mais para a frente.

### **Tipo de variável: Var**

Vamos começar com **var.** A variável abaixo atribui a palavra **nome** à Nayara. Sempre que digitar **nome** no console (inspecionar do navegador), ele vai trazer “Nayara”:

**Código:**

*var* nome = 'Nayara';

**Console:**



Se criar o código abaixo, ele também vai trazer o nome lá no console, sem precisar digitar nada:

**Código:**

console.log (nome)

**Console:**



Várias variáveis juntas, chamando todas ao mesmo tempo separadas por vírgula, dá certo também:

**Código:**

*var* nome = 'Nayara';

*var* idade = 26;

*var* possuifaculdade = true;

console.log (nome, idade, possuifaculdade, nome)

**Console:**



Outras variáveis:

**Código:**

*var* preco = 50;

*var* totalcomprado = 20

*var* precototal = preco \* totalcomprado;

console.log (precototal);

**Console:**

**Uma imagem contendo Tabela

Descrição gerada automaticamente**

### **Variáveis na mesma linha**

Dá para declarar várias variáveis e informar só 1 var, basta separar entre vírgula e no final **sempre** fechar com ponto e vírgula:

**Código:**

*var* cidade = 'Barueri',

habitantes = '250000', quantidadecachorros = '3800';

console.log (cidade, habitantes, quantidadecachorros)

Outra forma **mais utilizada (um debaixo do outro):**

*var* cidade = 'Barueri',

    habitantes = '250000',

    quantidadecachorros = '3800';

console.log (cidade, habitantes, quantidadecachorros)

**Console:**

****

### **Variáveis sem valor**

Existem também as variáveis sem valor, ou seja, podemos dar um nome para ela, mas sem definir algo a se retornar. No console, irá retornar **undefined:**

**Código:**

*var* datafinal;

console.log(datafinal)

**Console:**

****

### **Nome de variáveis**

Devem seguir as seguintes regras:

* Podem iniciar com letras;
* Podem iniciar com símbolos \_ e $;
* Não podem ter acentos e espaços;
* Podem conter números, mas não iniciar com eles;
* Sensíveis ao case (letras maiúsculas/minúsculas). Ex: nome é diferente de Nome;
* Não utilizar palavras reservadas. Ex.: var, delete, in, if, try, new (consultar lista em <https://www.w3schools.com/js/js_reserved.asp>);
* É comum utilizar o camel modal na hora de escrever uma variável com mais de uma palavra. Ex: clicarBotao (segunda palavra com letra maiúscula para mostrar que é outra palavra, já que não pode espaço):

Texto

Descrição gerada automaticamente

### **Hoisting**

O JS “joga” todas as informações para cima.. contudo, se uma variável é definida na linha 27, só a partir da linha 28 que o console começará a lê-la. No exemplo abaixo, lançamos um console antes de definir a variável que ele pede, então ele traz **undefined**, pois é como se a variável tivesse sido definida sem valor:

**Código:**

console.log(datafinal)

*var* dataFinal = '29/12/2022';

**Console:**



### **Mudar o valor atribuído**