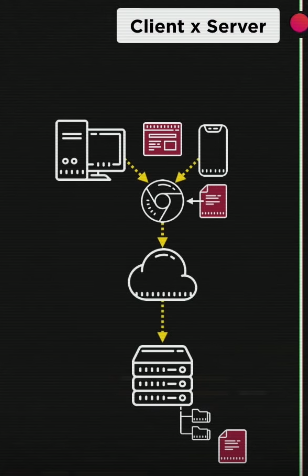
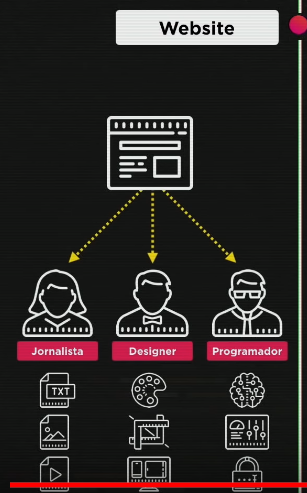
**JavaScript – ECS 6**

# Introdução

JavaScript é uma linguagem multi-paradigma, funciona com o paradigma funcional.

## Para que serve

Client-side (foco), também é utilizada no lado do servidor.

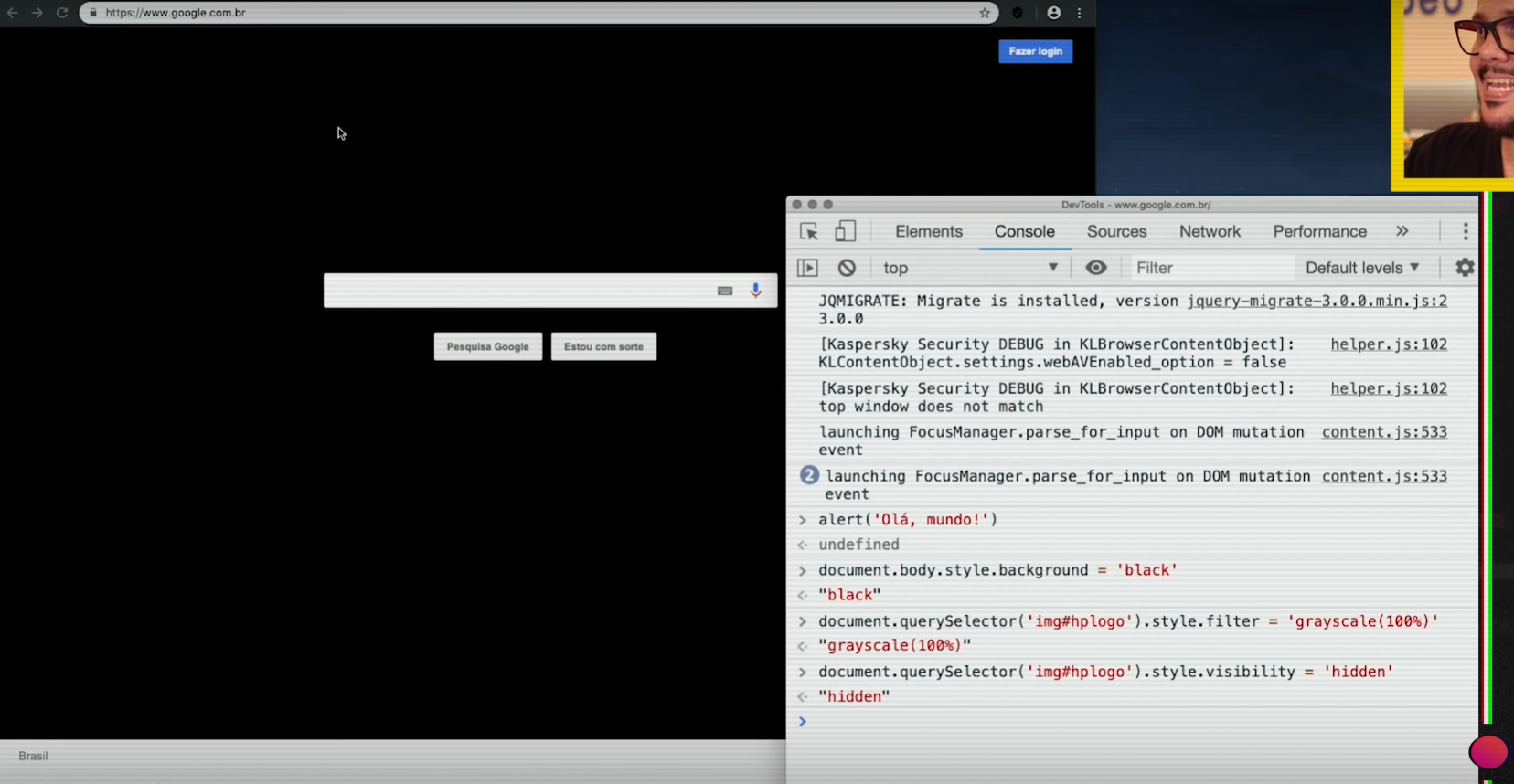
 

Não são linguagens de programação

3 peças essenciais:

* Conteúdo - HTML
* Estilo - CSS
* Interações - JS

Interações com a WEB:



As alterações feitas são realizadas apenas na cópia da página que o servidor mandou para a sua requisição.

## Evolução

1970 – surgimento da internet (arpanet - internet)– década de 60 (Guerra Fria)

JavaScript – não tem relação com Java, na época tava fazendo sucesso falar a linguagem Java (estava nascendo e fazendo sucesso) (o nome, estava sendo muito falada – linguagem do futuro), eles usaram o mesmo nome para atrair atenção, jogada de marketing. As duas (Java e JavaScript) foram baseadas na linguagem C.

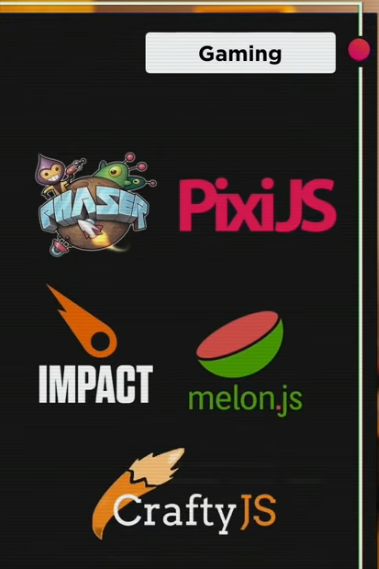
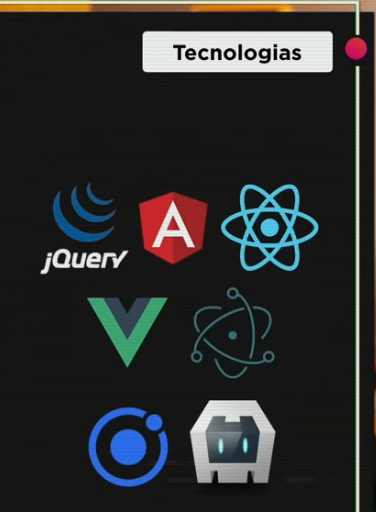
ECMA – empresa europeia de padronização, tipo a ISSO dos EUA -> **ECMAScript (versão padronizada JavaScrip)**

Nodejs -> máquina que roda JS fora do navegador, foi a saída do JS dos navegadores. Atualmente é possível ter JS do lado do servidor também.

Tecnologias

* **JQuery**: conjunto de bibliotecas. Facilita o uso de interatividades em JS.
* **Angular**: criado e mantido pelo Google. Facilita muito a criação de aplicações WEB, linguagem mais declarativa – semelhante ao SQL.
* **React**: semelhante ao Angular, flexibilidades maiores – Reactive Native-> aplicações em celular
* **VUE**: framework.
* **Electron**: mantido pela GitHub, especializado para a criação de interfaces gráficas, programinhas de interface gráfica (VisualStudio Code, chat Discord).
* **Ionic**: SDK para criar aplicações para dispositivos móveis.
* **Cordova**: Apache.

Vários utilizam o NodeJS.

Tecnologias que utilizam o JS

Criação de jogos em JS

# Primeiros Passos

Visual Studio Code – integração com o NodeJS, facilidade de criar ambientes internos. – Cntrl+L – limpar

Para colocar uma coisa dentro de parênteses, você seleciona e coloca tudo e abre o parênteses, o VSC completa automaticamente.

NodeJS – rodar o JS fora do navegador integrando com o VSC.

**JS NÃO é necessário utilizar ;**

**O Script roda antes de tudo**

# Comandos

## Documento

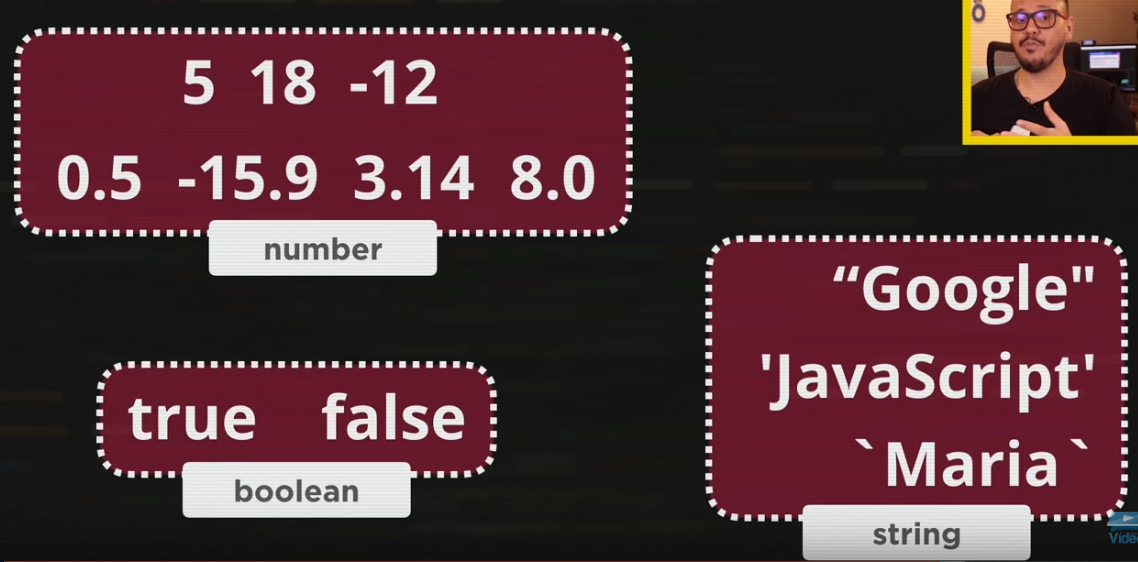
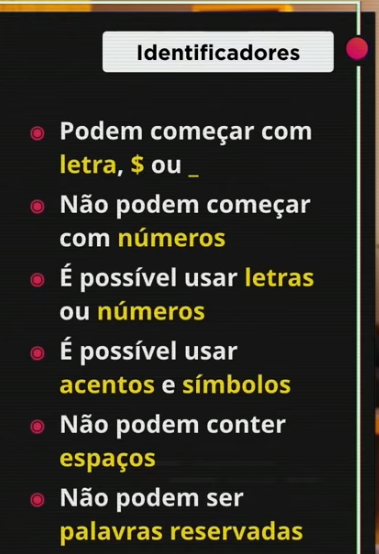
* Document.write() -> escreve no documento

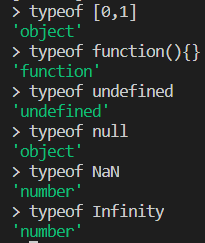
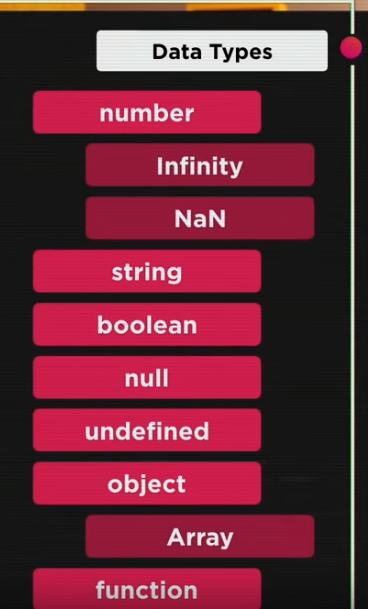
## Window

* **Window.alert(‘’) -> alerta, só possui a opção de OK.**
* **Window.confirm(‘’) -> confirmação, OK Cancel.**
* **Window.prompt(‘’) -> campo de digitação.**

## Variável

**Pode-se usar var ou let (deixa de existir quando sai do bloco).**





typeof

Funções são consideradas tipos primitivos de dados

NaN – not a number

### Conversão de tipos



Number(n) – o JS identifica automaticamente o tipo de dado e faz a conversão



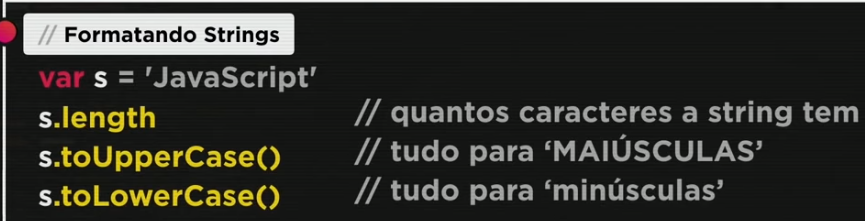
### Template Strings

**Utilizado para simplificar a formatação, utiliza o place holder.**

**Abre e fecha com crase e utiliza o ${} para inserir variáveis**



### Funções



Lenght é um atributo, por isso não possui o parênteses

### Variáveis Compostas

Arrays -> let num = []

Não precisa definir tamanho. Aloca de acordo com o que precisa.

É heterogêneo, aceita valores em um mesmo array de tipos diferentes.

Para acrescentar novos valores -> num.push()

Para saber o tamanho(quantidade de elementos) -> num.length (sem parênteses, pois não é método, é atributo)

Ordenar em ordem crescente -> num.sort()

Acessar posições -> num[indice]

Percorrendo o vetor (objects):

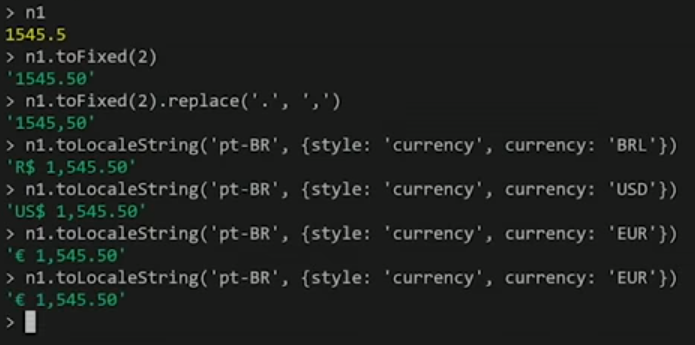
For(let pos in num){ console.log(num[pos]) }

Retorna o índice do elemento -> num.indexOf(7) -> retorna -1 se o valor não for encontrado.

### Formatando Números

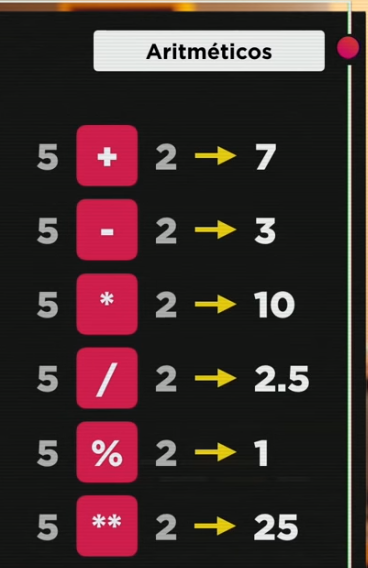
Formata a String – manda localizar a String em português, currency – valor monetário, BRL – real do Brasil. -> USD – dólar EUR – euro.





## Operadores



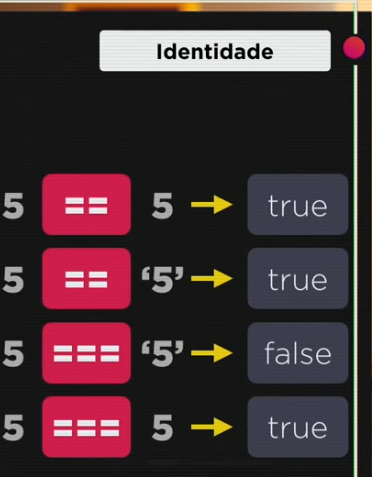


/ -> divisão real

% -> divisão inteira (módulo da divisão)

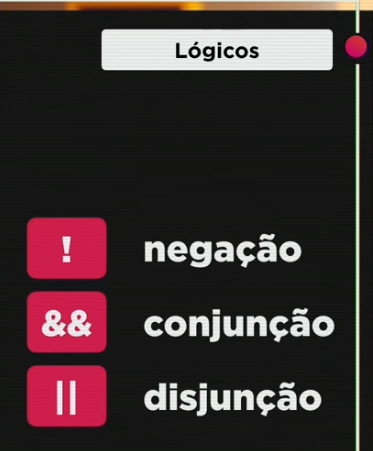
\*\* -> potência

Operador Identidade



== -> analisa o valor, e não o tipo

=== -> operador de igual restrita: analisa o valor e o tipo

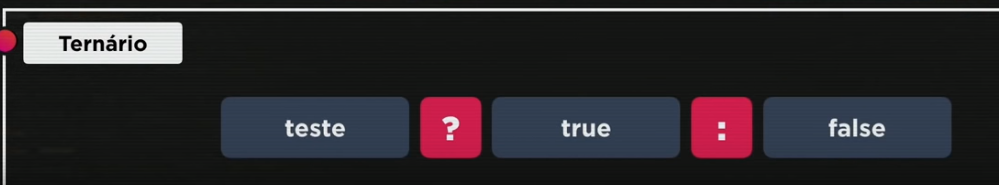


&& -> e

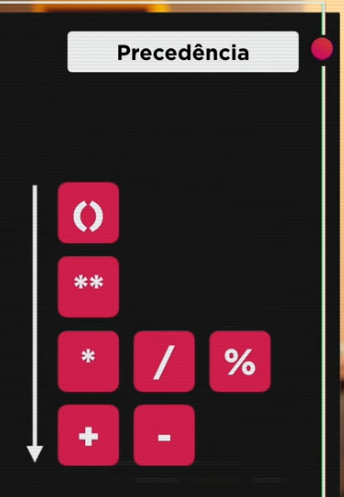
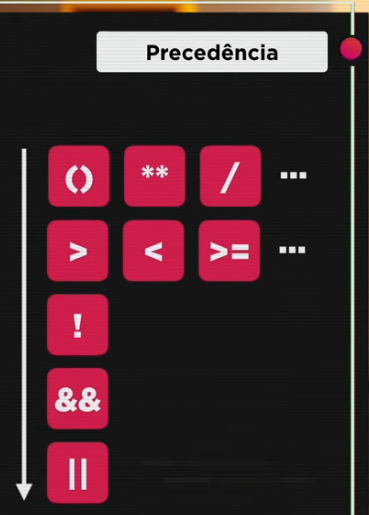
|| -> ou

Operador Ternário

Ternário porque tem três partes. Executa a operação, se for verdade atribui o antes do : se for falso, depois do :



### Ordem de Precedência

## Demais Comandos

### Pegar a hora atual

var hora = new Date()

hora.getHours()

o getDay retorna o dia da semana de 0 a 6 (Dom a Sáb)

### Simbolo Copyright

&copy;

### Formatação Unicode para javaScript:

Ex: [U+1F4A9](https://unicode.org/emoji/charts/full-emoji-list.html" \l "1f4a9) -> `\u{1F4A9}` (o código só funciona entre crases)

Verificar qual string termina em uma variável: nome.endsWith(‘o’) -> retorna verdadeiro ou falso.

# DOM – Document Object Model

Instalar Watch in Chrome

Node Exec – permite usar a tecla f8 para rodar o código somente em JS no node

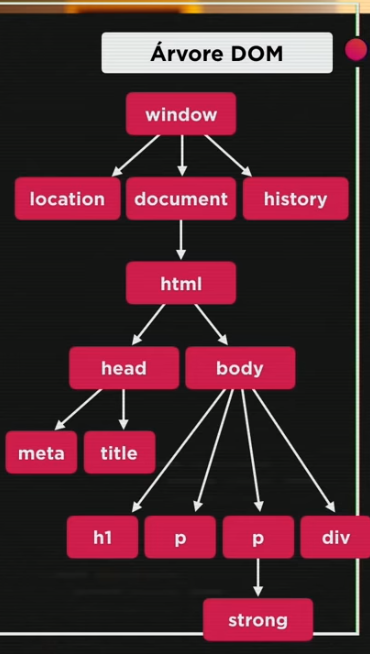
Crtl + shift + P -> watch in chrome

Modelo de objetos para documentos

É um facilitador, conjunto de objetos dentro do seu navegador que dá vai dar acesso aos componentes internos do seu website. Apenas no navegador, não funciona dentro do node.js

## Árvore DOM

Cada página possui a sua própria árvore DOM.



Elementos, objetos DOM:

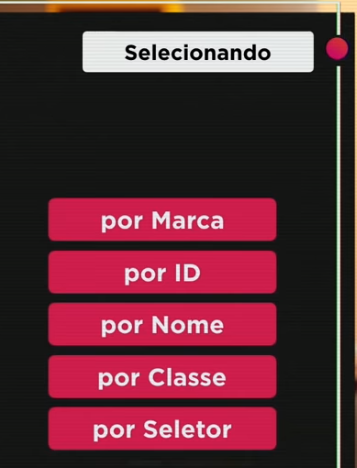
Window (raiz) -> janela

Location -> qual é a localização do seu site, qual a URL, qual a página atual, qual foi a página anterior

Document -> documento atual

History -> vai guardando de onde você veio, pra onde você vai. Isso facilita a navegação dentro do seu site

Html -> parte HTML do seu site



Marca : getElementsByTagName()

ID – CSS : getElementById() - # - id . - class

Nome: getElementsByName()

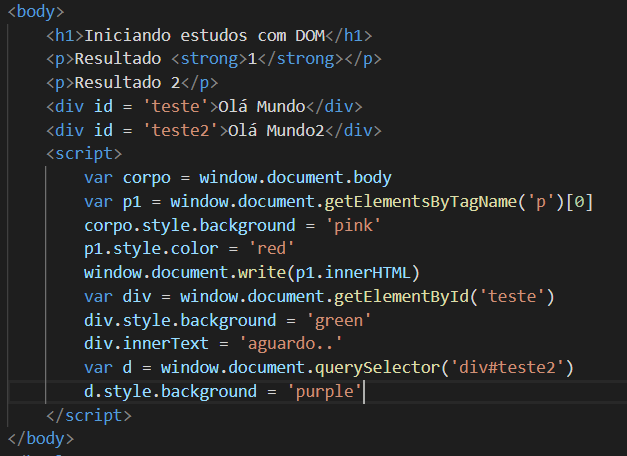
Classe: getElementsByClassName()

Seletor (recomendável) seleção por CSS: querySelector(), queryselectorAll()

Marca : TAG

Seletor: CSS

innerText – texto que está dentro / innerHTML – pega a formatação em HTML, com as tags filhas



## Eventos

Tudo aquilo que possa acontecer com qualquer elemento.

Eventos mais comuns: mouse.



**Mouseenter**: movimento do mouse até chegar dentro do elemento.

**Mousemove**: movimento do mouse dentro do elemento.

**Mousedown**: clicar e segurar.

**Mouseup**: soltar o botão do mouse.

Click: cliques rápidos.

**Mouseout**: movimento do mouse para fora do elemento.

Existem vários elementos que podem acontecer em elementos JS. <<https://developer.mozilla.org/pt-BR/docs/Web/Events>>

Utilizando HTML:

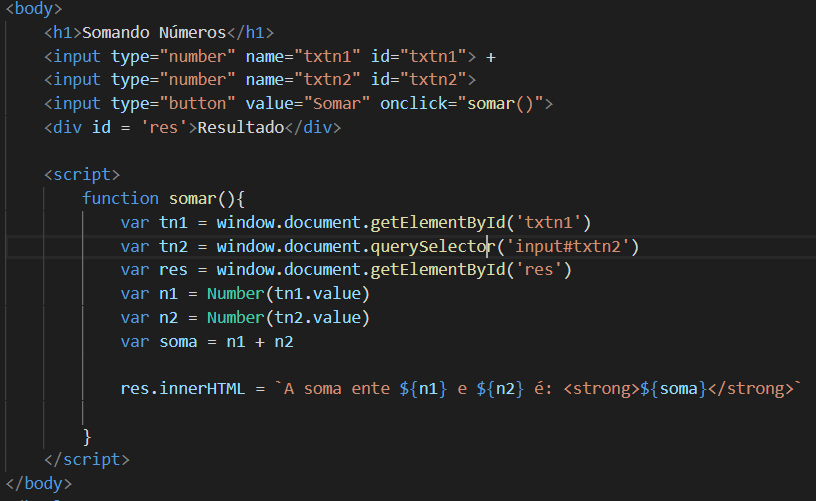


Disparo de eventos no JS utilizando listeners (ouvidores, prestando atenção)



Dessa forma o HTML fica limpo, apenas o JS trata os casos.

Exemplo básico somando dois números:



## Comandos

### Criar imagem dinamicamente

var img = document.createElement('img')

img.setAttribute('id', 'foto')

img.setAttribute('src', 'img/adulta.png')

res.appendChild(img)

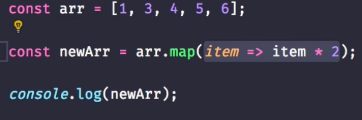
# Condições

Para escrever na linguagem JS você utiliza o comando: console.log(‘’)

Para executar JS utiliza a extensão node exec e executa com o atalho f8

# Arrow Functions

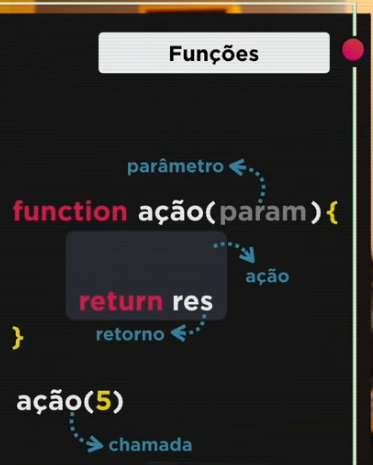
É utilizada em ambiente de função anônima.



# Como detectar erros

Nas IDE’s o erro não é mostrado, você entra no navegador, vai em inspecionar (inspector) e o erro aparece

# Funções

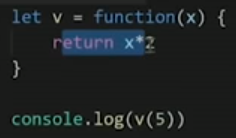


**Funções**: Conjunto de código que serão executadas apenas quando o evento ocorrer.

Os eventos podem ser configurados na parte HTML ou diretamente no JS.

Pode ter as chamadas automatizadas.

É possível colocar funções dentro de variáveis.



Existem a arrow functions: =>

setInterval() -> temporizador que fica chamando a função de acordo com o tempo de intervalo que você passou como parâmetro.

# Exercícios Práticos

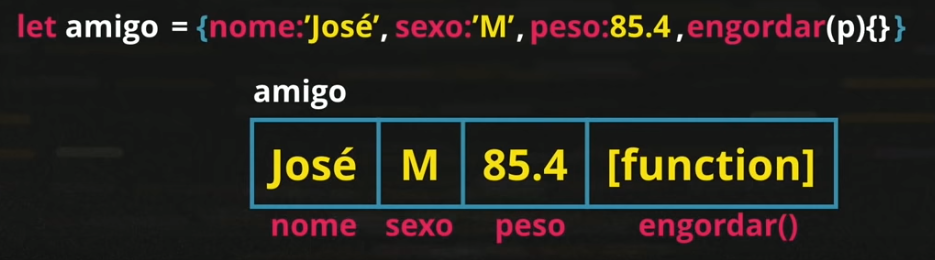
Num.focus() -> foca o cursor do mouse no elemento(faz ficar piscando, seleciona a caixa de texto).

# Estudar

## Funções

* JS funcional - Arrow functions, call-backs, funções anônimas, iife
* Objetos – JS é também uma linguagem orientada a objetos
* Modularização – modularizar o código, deixar o código separado em arquivos separados, reutilizar códigos
* Expressões Regulares (RegEx) – ex: para validação de dados
* JSON
* AJAX – ex: carregamento de conteúdo dinamicamente, sem as teclas de passar para a próxima página, chegou no final do rolamento, carrega mais
* NodeJS – fazer os scripts rodarem do lado do servidor
* JQuery, Angular..

# Objetos

Declaração: 

Chaves definem objetos.

Ao invés do 0,1,2.. dos arrays, passam a ter identificadores (atributos)

Objetos podem guardar funcionalidades, métodos.

# Bibliografia



Guia de Referência – Gratuitos

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

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

<<https://www.ecma-international.org/publications/standards/Ecma-262-arch.htm>>

Guias - Pagos