JAVA SCRIPT

fatec – senai centro | Rua Caetés 741, Belo Horizonte

JAVA SCRIPT E ECMASCRIPT

Rubem Cândido dos Santos

2020

Sumário

[Introdução 3](#_Toc36196140)

[Definição 3](#_Toc36196141)

[O que o Java Script faz? 3](#_Toc36196142)

[Client x Server 3](#_Toc36196143)

[HTML x CSS x JavaScript, “a receita do bolo” 3](#_Toc36196144)

[Prática 1 – Desabilitando e alterando JavaScript no navegador 4](#_Toc36196145)

[Prática 2 – Mensagem de olá Mundo no navegador. 13](#_Toc36196146)

[Prática 3 - Background negrito 16](#_Toc36196147)

[Prática 4. Alterar a cor da logo 18](#_Toc36196148)

[1 Passo: 18](#_Toc36196149)

[Pratica 5 – Esconder a logo 22](#_Toc36196150)

[Prática 6 – Alterar o conteúdo de uma manchete 25](#_Toc36196151)

[Evolução do JavaScript 28](#_Toc36196152)

[Ferramentas necessárias para o JavaScript 29](#_Toc36196153)

[Um Navegador 29](#_Toc36196154)

[Um editor de texto 29](#_Toc36196155)

[Um Interpretador de JavaScript 29](#_Toc36196156)

[window.arlet 30](#_Toc36196157)

[window.confirm 30](#_Toc36196158)

[window.prompt 30](#_Toc36196159)

[Comandos básicos 30](#_Toc36196160)

[Comentário 30](#_Toc36196161)

[Variáveis 30](#_Toc36196162)

[Identificadores 30](#_Toc36196163)

[O que não é permitido 31](#_Toc36196164)

[Prática 7 – Variáveis 31](#_Toc36196165)

[Utilizando o Terminal no Visual Studio Code 33](#_Toc36196166)

[Tipos Primitivos 40](#_Toc36196167)

[Typeof 41](#_Toc36196168)

[Manipulação de Dados 41](#_Toc36196169)

[Atividade 1: 43](#_Toc36196170)

[Convertendo String para Number 43](#_Toc36196171)

[Convertendo String para número inteiro 45](#_Toc36196172)

[Convertendo String para número real, flutuante 47](#_Toc36196173)

[Convertendo String para número indefinido 49](#_Toc36196174)

[Atividade 2 50](#_Toc36196175)

[Convertendo Número para String 51](#_Toc36196176)

[Template Strings (novidade do EcmaScript) 52](#_Toc36196177)

[Formatando Strings 54](#_Toc36196178)

[length 54](#_Toc36196179)

[toUpperCase() 54](#_Toc36196180)

[toLowerCase() 54](#_Toc36196181)

[Formatando Números 56](#_Toc36196182)

[toFixed() 56](#_Toc36196183)

[toFixed().replace() 56](#_Toc36196184)

[toLocaleString() 56](#_Toc36196185)

[Github 58](#_Toc36196186)

[Conclusão 58](#_Toc36196187)

[BIBLIOGRAFIA BÁSICA: 59](#_Toc36196188)

[FLANAGAN, David. JavaScript: o guia definitivo / David Flanagan; tradução: João Eduardo Nóbrega Tortello; revisão técnica: Luciana Nedel. – 6. ed. – Porto Alegre: Bookman, 2013. O Guia Definitivo. ed. O’Reilly. 59](#_Toc36196189)

# Introdução

## Definição

JavaScript é uma linguagem de programação que permite a você criar conteúdo que se atualiza dinamicamente, controlar múltimídias, imagens animadas.

## O que o Java Script faz?

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. Podemos esperar encontrar códigos escritos em JavaScript.

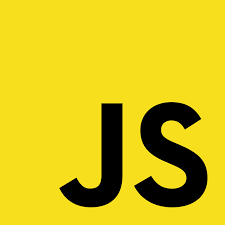


Figura1. Logo da JavaScript

# Client x Server

As máquinas que acessam, consomem serviços, são considerados clientes. Os equipamentos, dispositivos que disponibilizam serviços, são os servidores.

Exemplo:

Quando acesso o Google Class Room estou como cliente.

Os materiais, slides, atividades estão disponibilizados em um servidor.

# HTML x CSS x JavaScript, “a receita do bolo”

Na Web, o JavaScript serve como uma parte de interação, porém, ele trabalha com outras tecnologias de construção Web muito utilizadas.

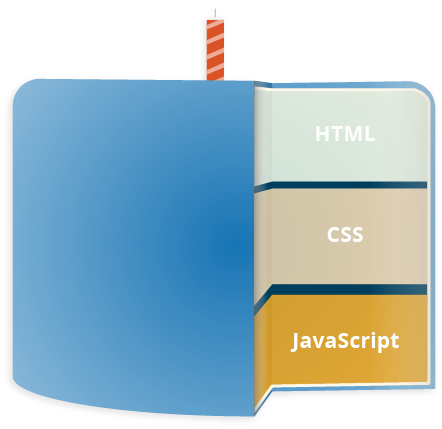


Figura 2. O conjunto de duas tecnologias e uma linguagem de programação

# Prática 1 – Desabilitando e alterando JavaScript no navegador

**1 Passo:** Abra o navegador Google Chrome.

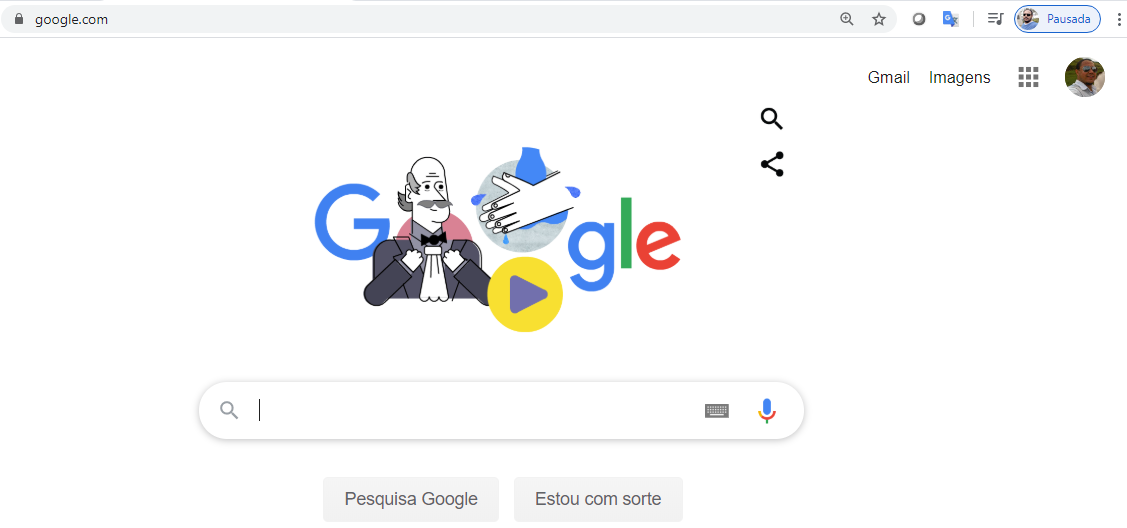


Figura 3. Página do google no dia 20 de março de 2020

**2 Passo:** Faça uma busca por SENAI

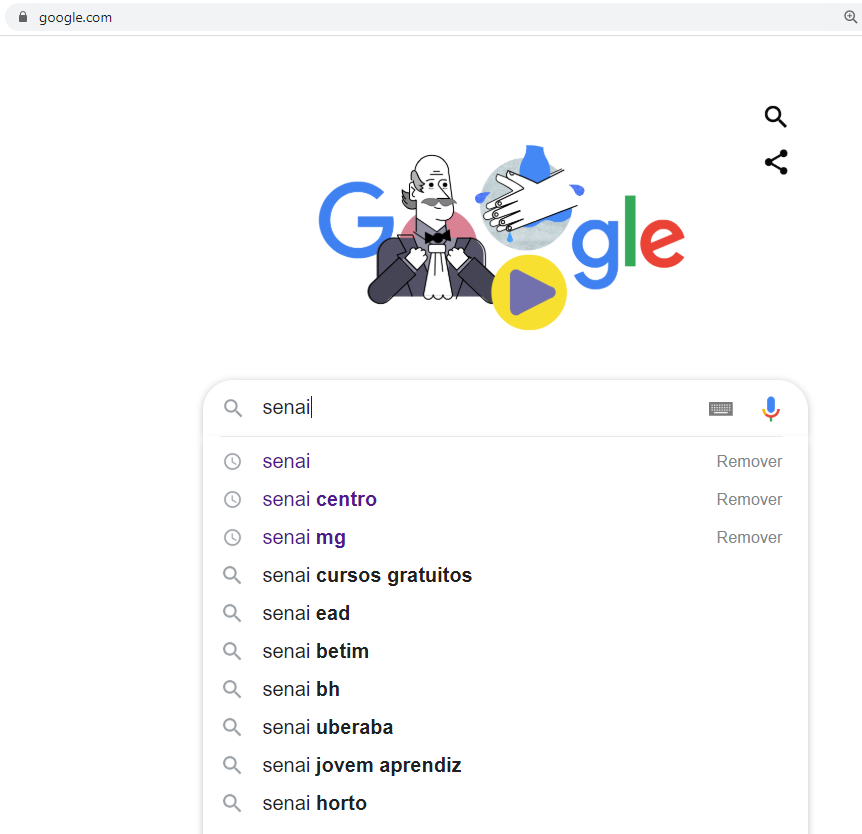


Figura 4. O JavaScript auto completando a busca

**3 Passo:** Clique com o botão direito do mouse sobre a área branca e clique em inspecionar

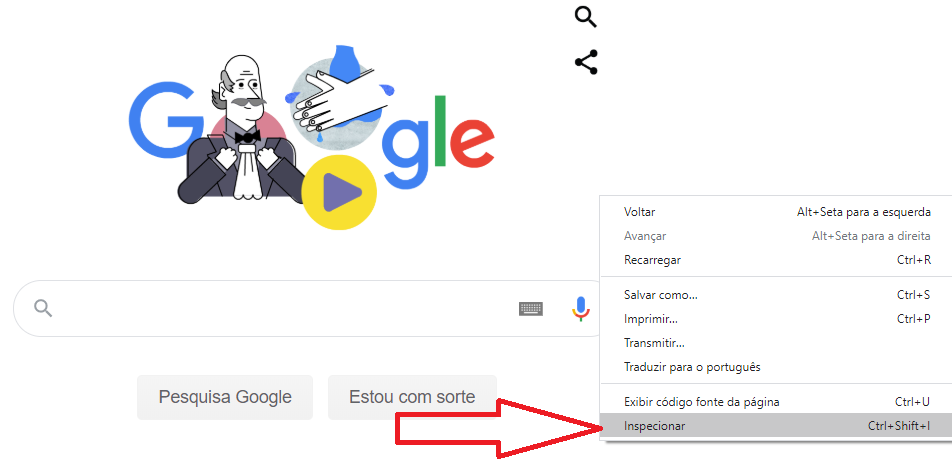


Figura 5. Botão Inspecionar

Podemos também utilizar o atalho Ctrl + Shift + I ou pressionar a tecla F12 do teclado.

4 Passo: Clique no DevTools (os três pontinhos)

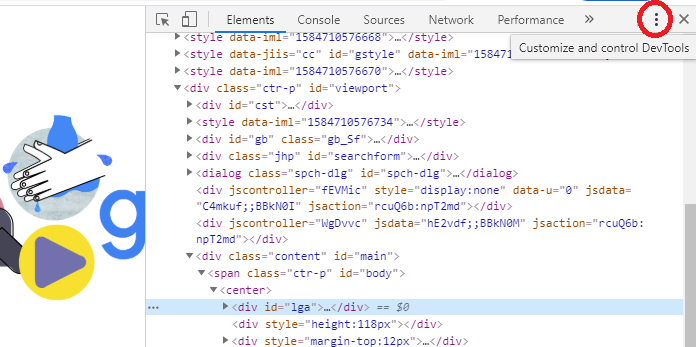


Figura 6. Botão do DevTools

Clique em Undock para destacar a janela

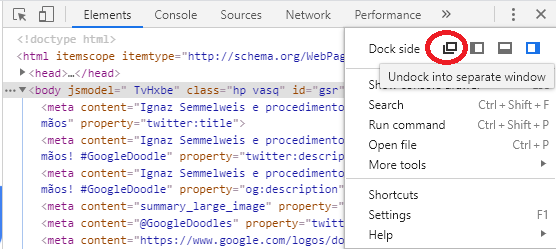


Figura 7. Destacando a janela DevTools

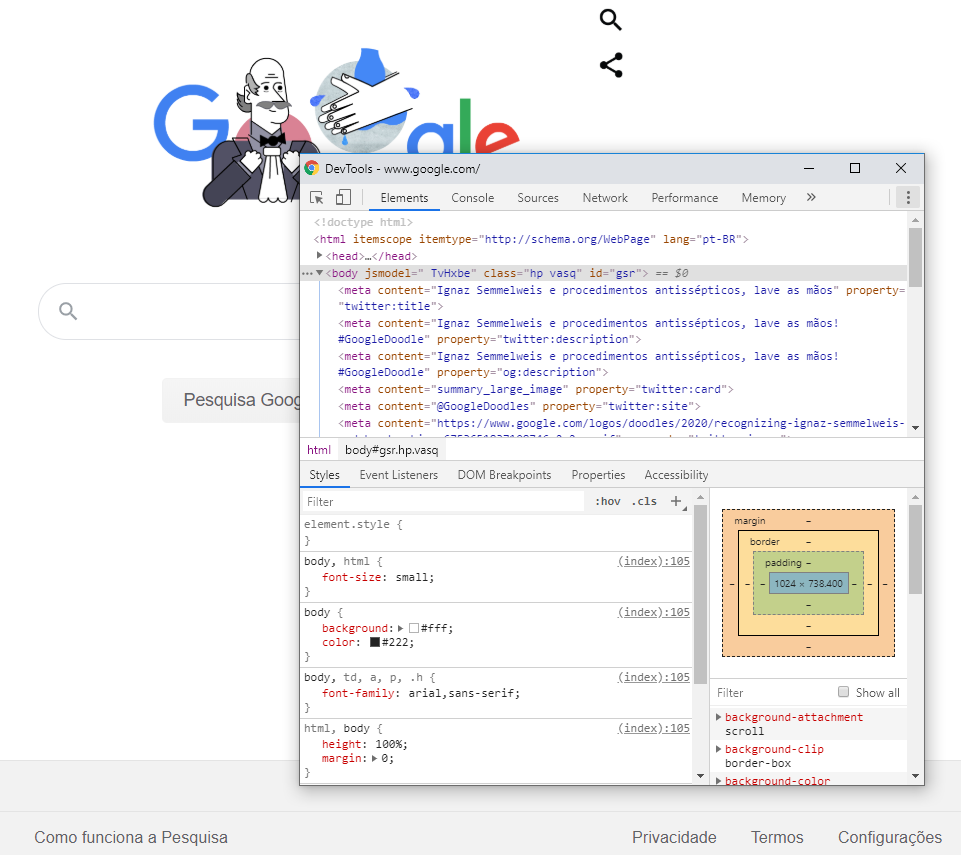


Figura 8. Janela DevTools destacada

Clique novamente nas reticencias e depois em Setiings

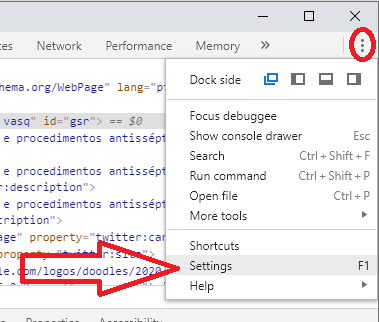


Figura 9. Botão DevTools

Role até encontrar a opção Disable JavaScript e marque para desabilitar JavaScript.

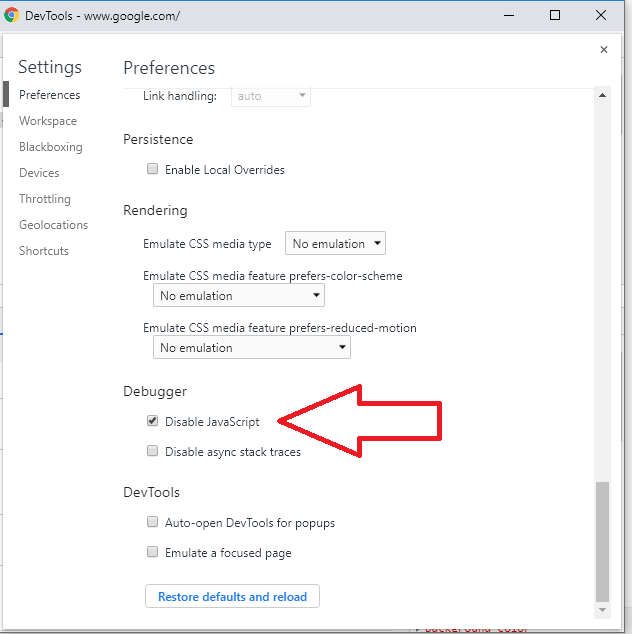


Figura 10. Disable javaScript

Atualize a página e observe as alterações na página.

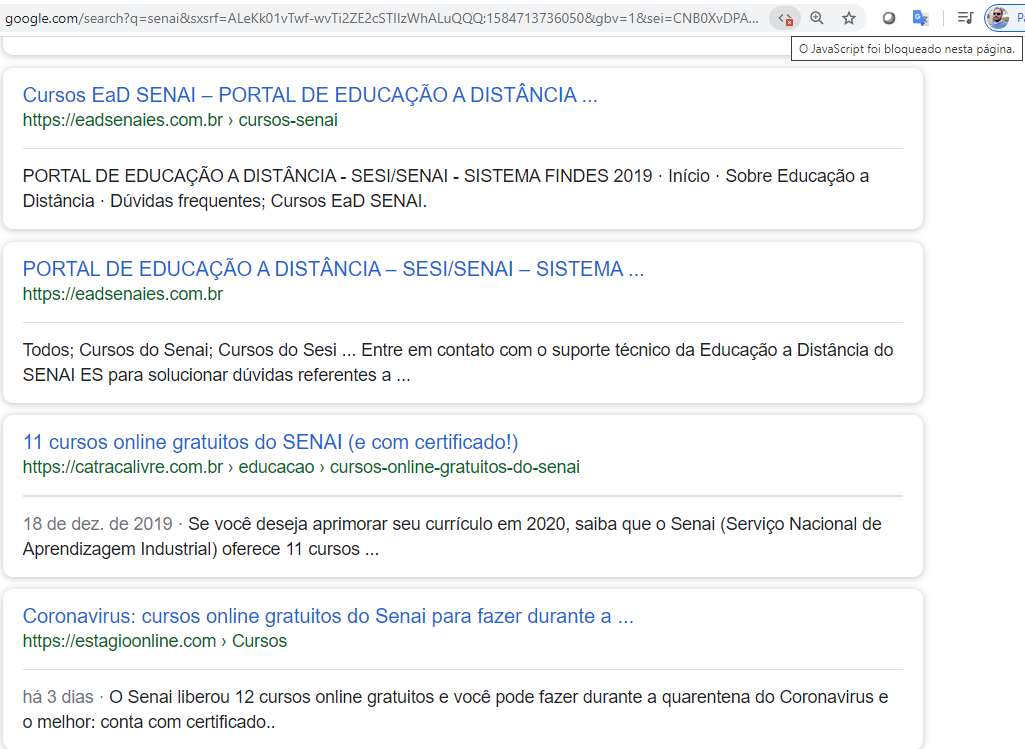


Figura 11. Página do google sem JavaScript

Tente a haba maps. Nem funciona.

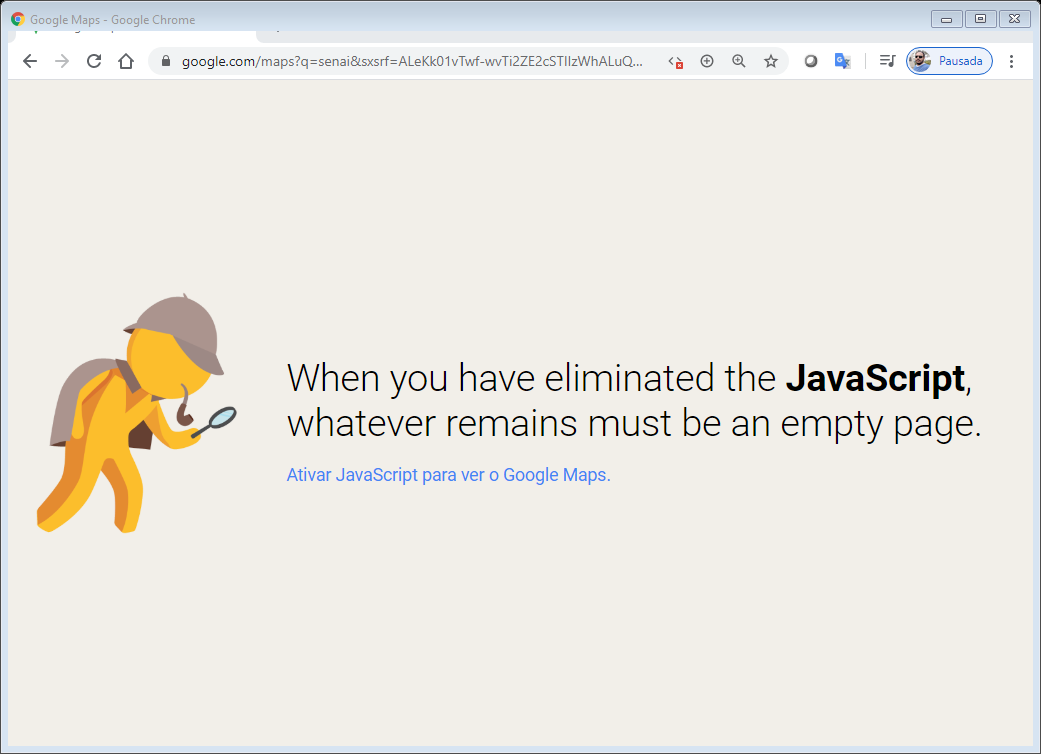


Figura 12. Mensagem da impossibilidade de utilização do Maps

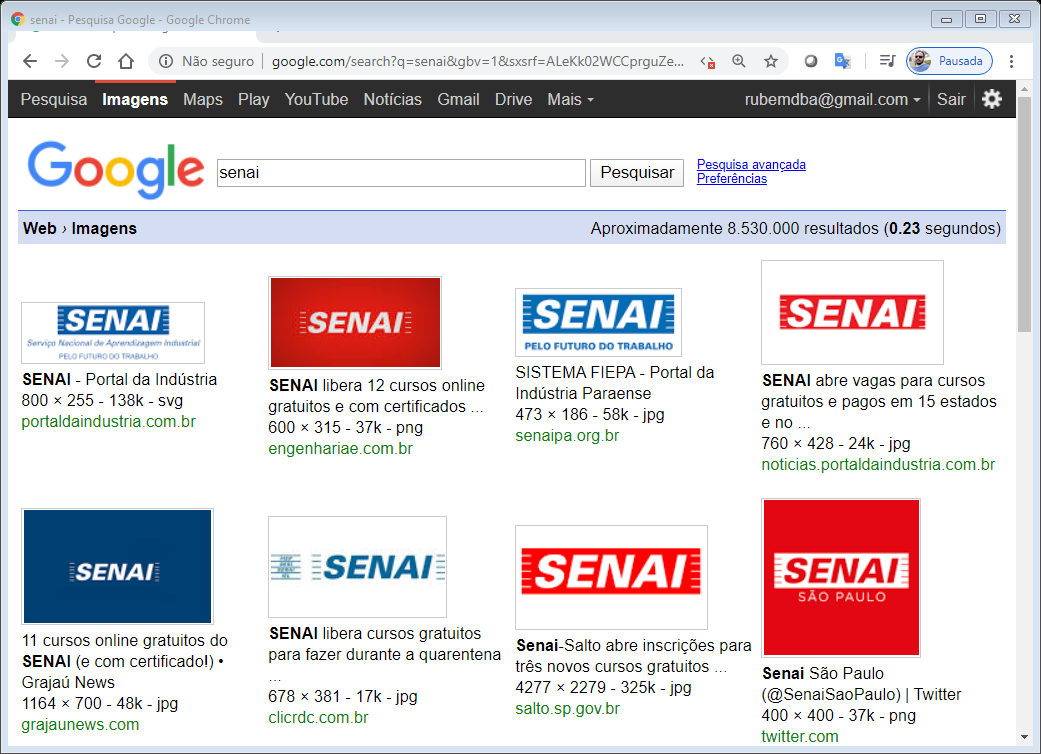


Figura 13. Imagens sem JavaScript

A rolagem infinita que é comum no google não aparece sem o JavaScript

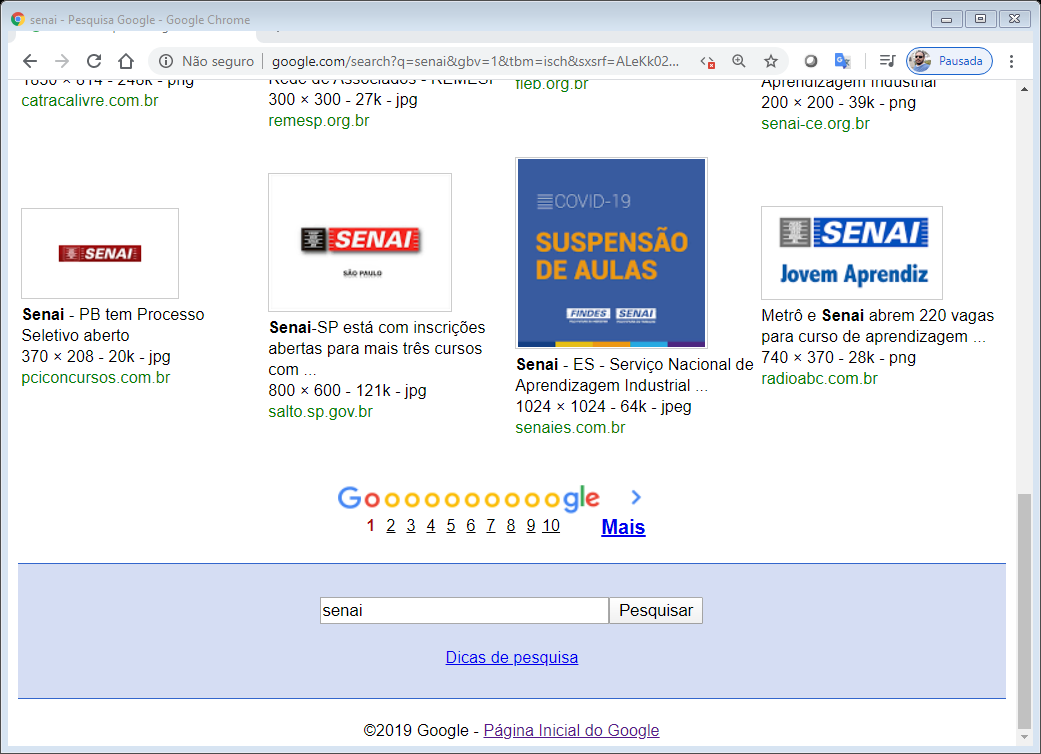


Figura 14. Botões de escolha de página

Com o JavaScript Habilitado

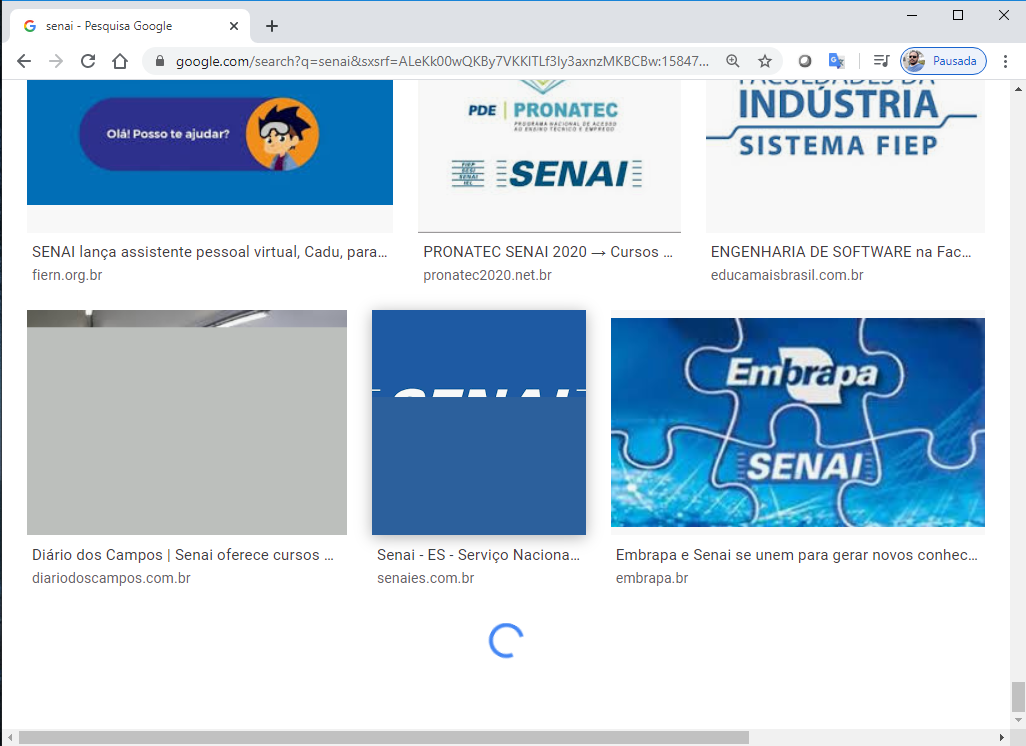


Figura 15. Busca por imagens com o JavaScript habilitado.

O youtube também não funciona sem JavaScript

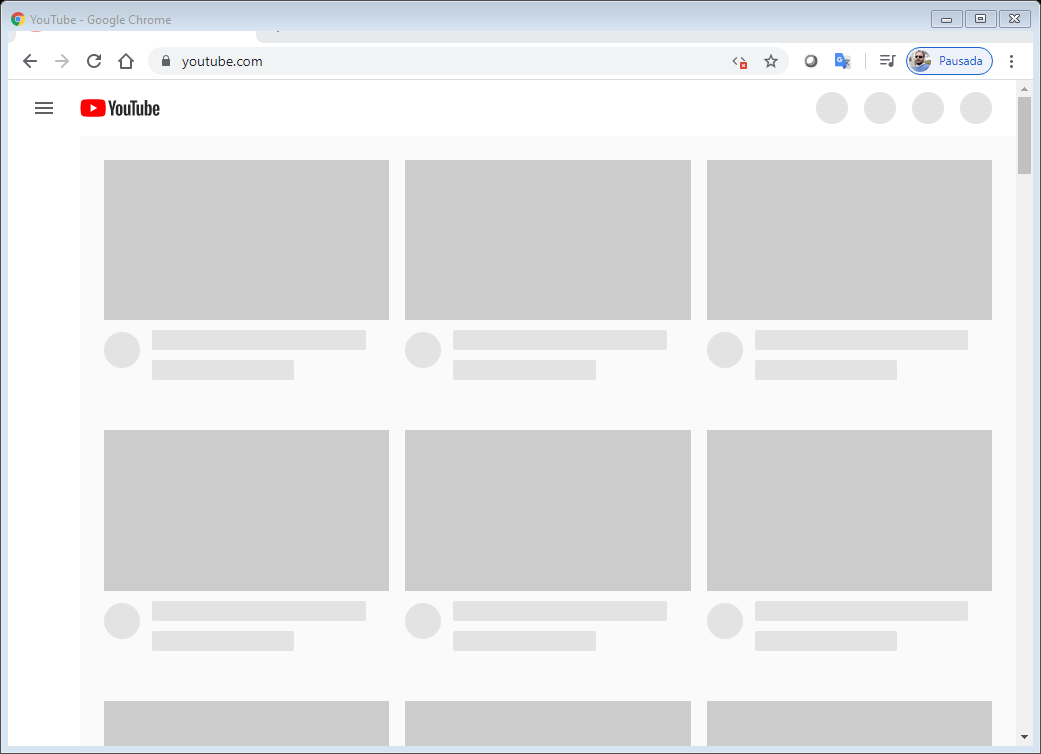


Figura 16. Youtube sem JavaScript

Olha a página do Facebook



Figura 17. Página do Facebook sem JavaScript

# Prática 2 – Mensagem de olá Mundo no navegador.

Com o JavaScript habilitdo no seu navegador.

**1 passo:** Clique na aba Console

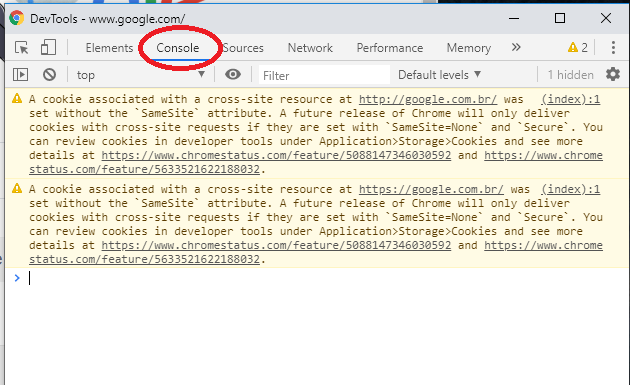


Figura 18. Aba de Console

**2 Passo:** No Console, digite **alert('olá, mundo')** e pressione **Enter**

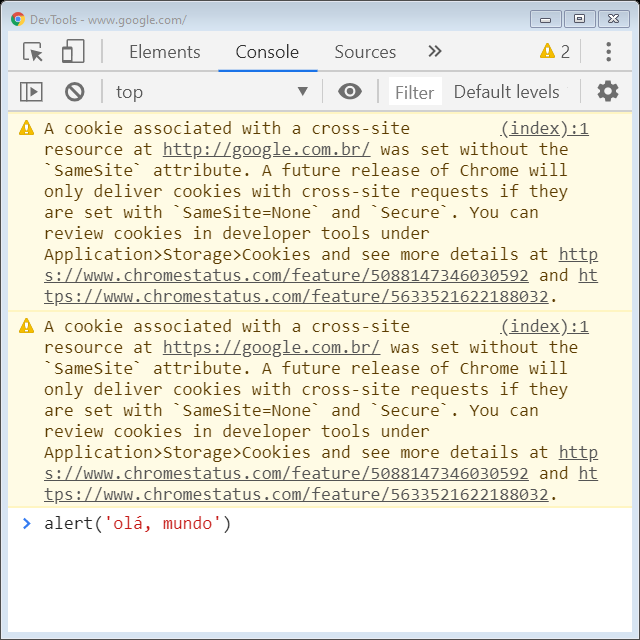


Figura 19. alert('olá, mundo')

Prática do Olá mundo no Console

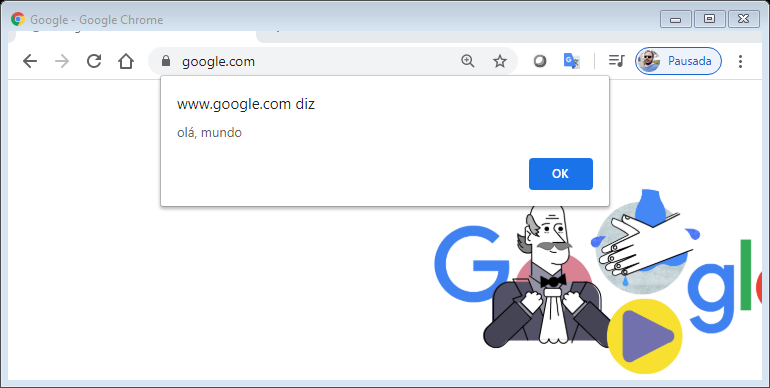


Figura 20. Mensagem de olá mundo

# Prática 3 - Background negrito

Digite o código abaixo na janela do console

document.body.style.background = 'black'

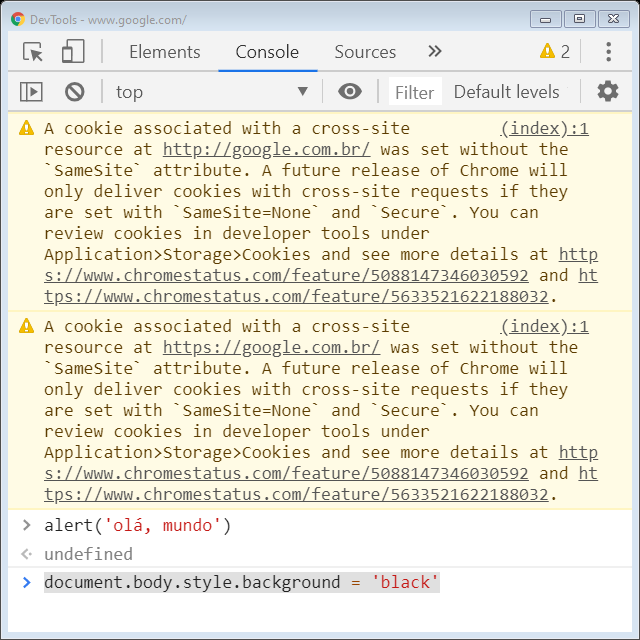


Figura 21. Alterando o fundo do navegador.

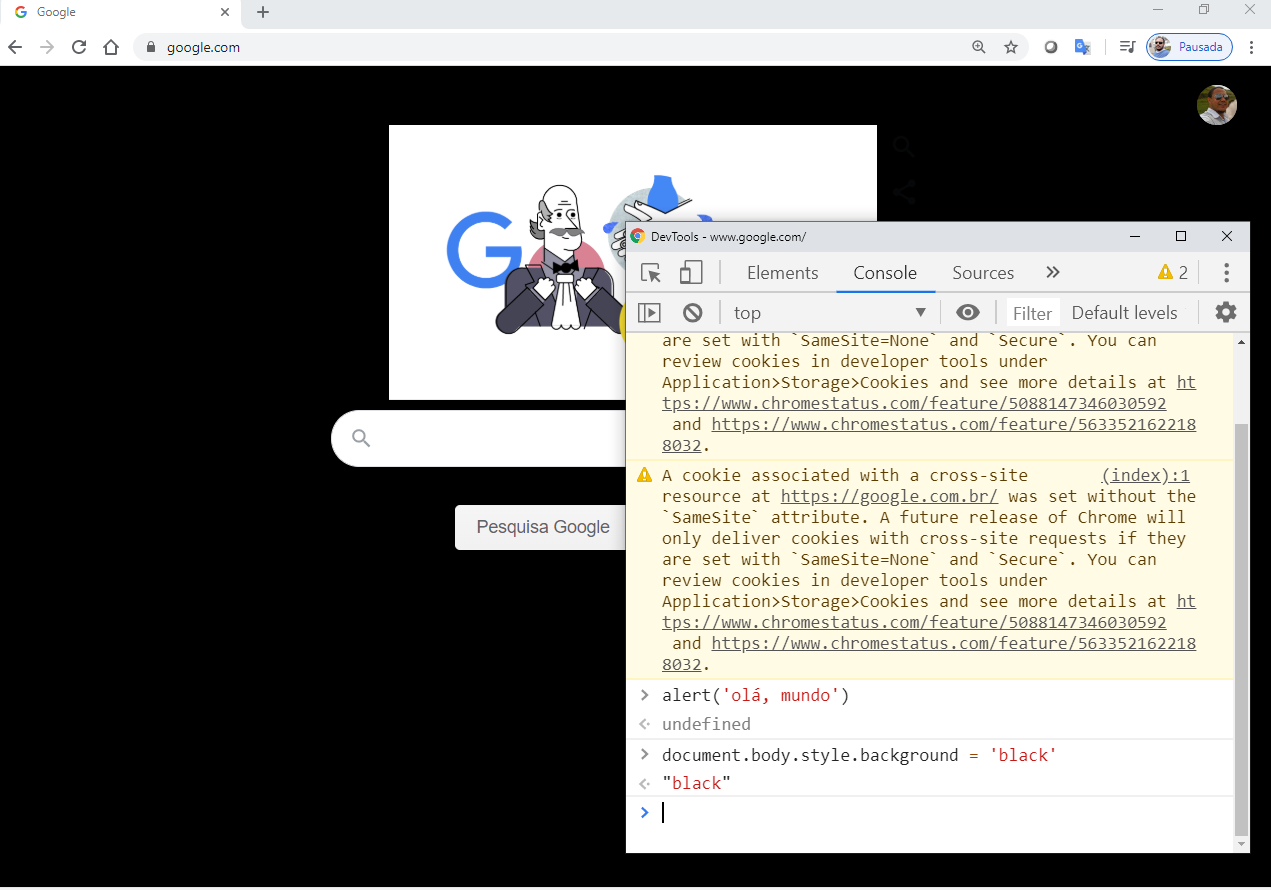


Figura 22. Background em preto

# Prática 4. Alterar a cor da logo

1 Passo: Clique no botão com a setinha (Select na elemento) para ver o nome do elemento.

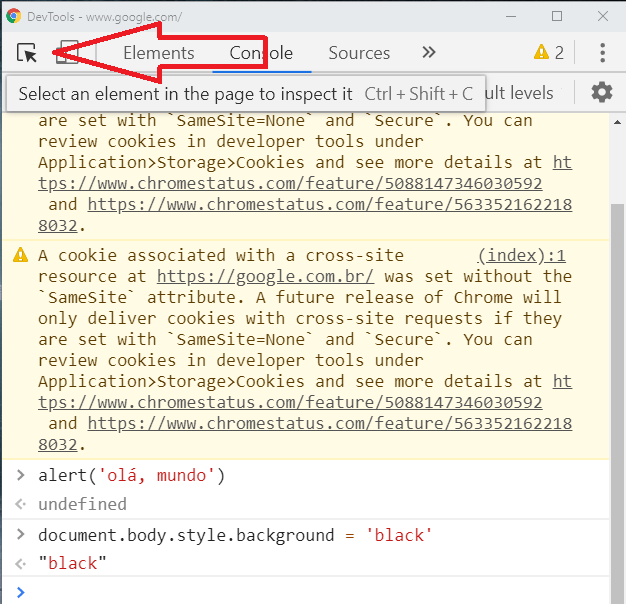


Figura 22. Botão Select na elemento

Passe o cursor do mouse em cima da imagem para que o nome do elemento seja mostrado.

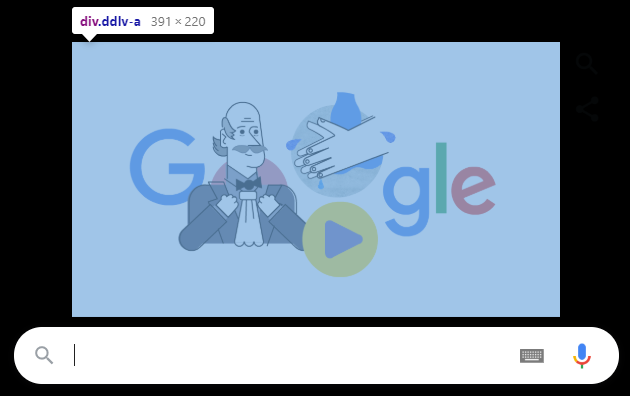


Figura 23. Nome do elemento encontrado

**2 Passo:** Digite o comando abaixo:

document.querySelector(**'div.ddlv-a'**).style.filter = 'grayscale(100%)'

div.ddlv-a é o nome do elemento que apareceu no momento da produção do tutorial.

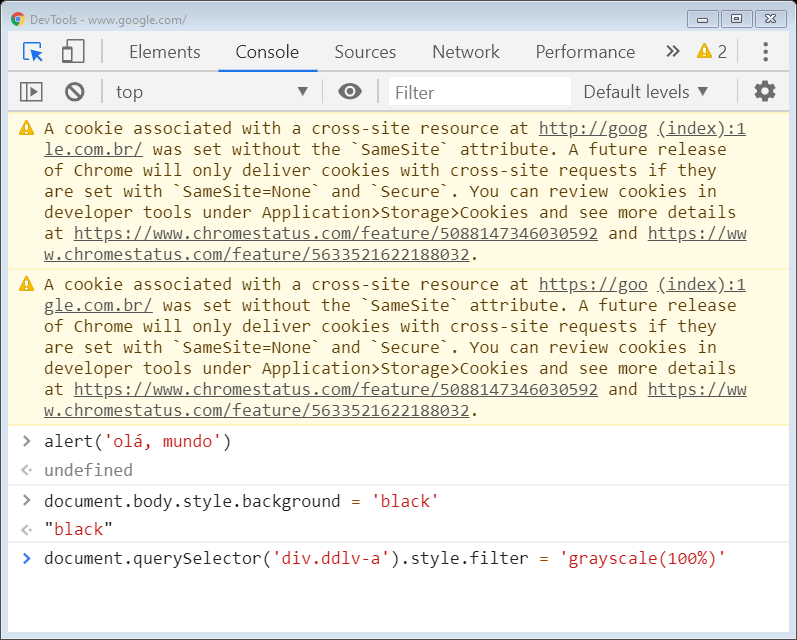


Figura 24. Comando para escurecer a logo

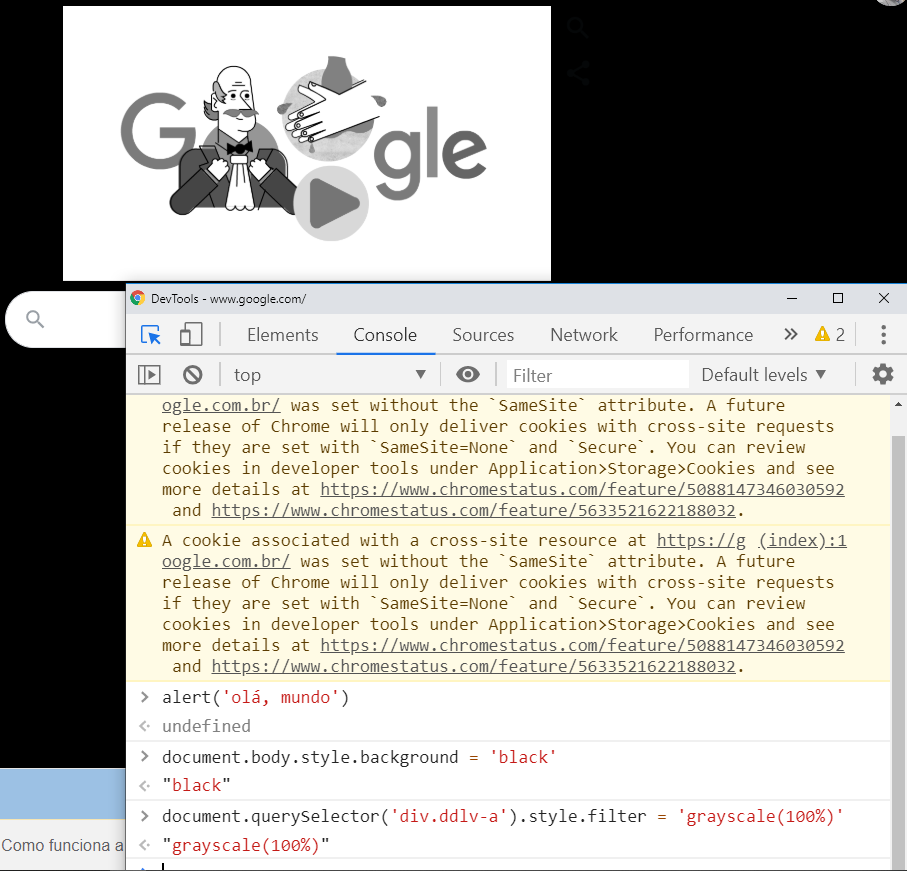


Figura 25. Logo da google com um tom de cinza

# Pratica 5 – Esconder a logo

Também no Console, digite o comando e tecle Enter

document.querySelector('div.ddlv-a').style.visibility = 'hidden'

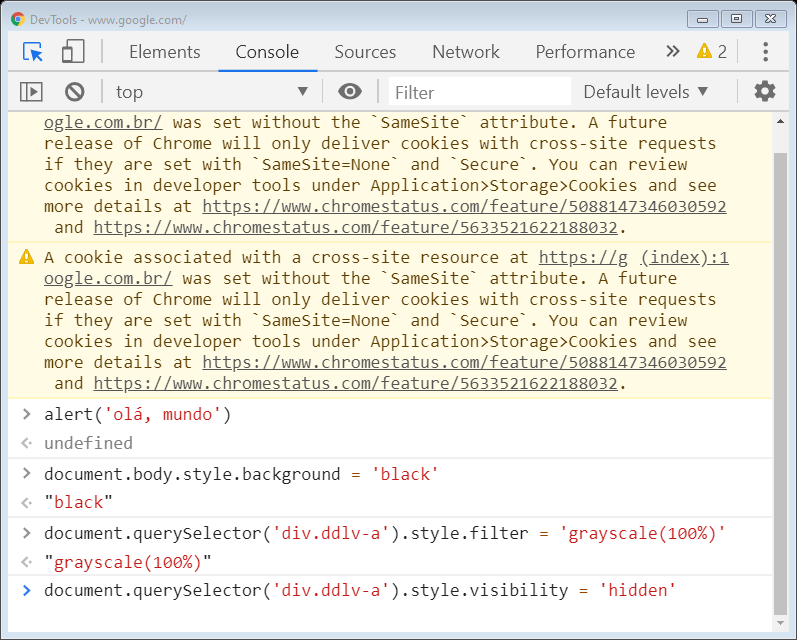


Figura 26. Comando hidden – escuro.

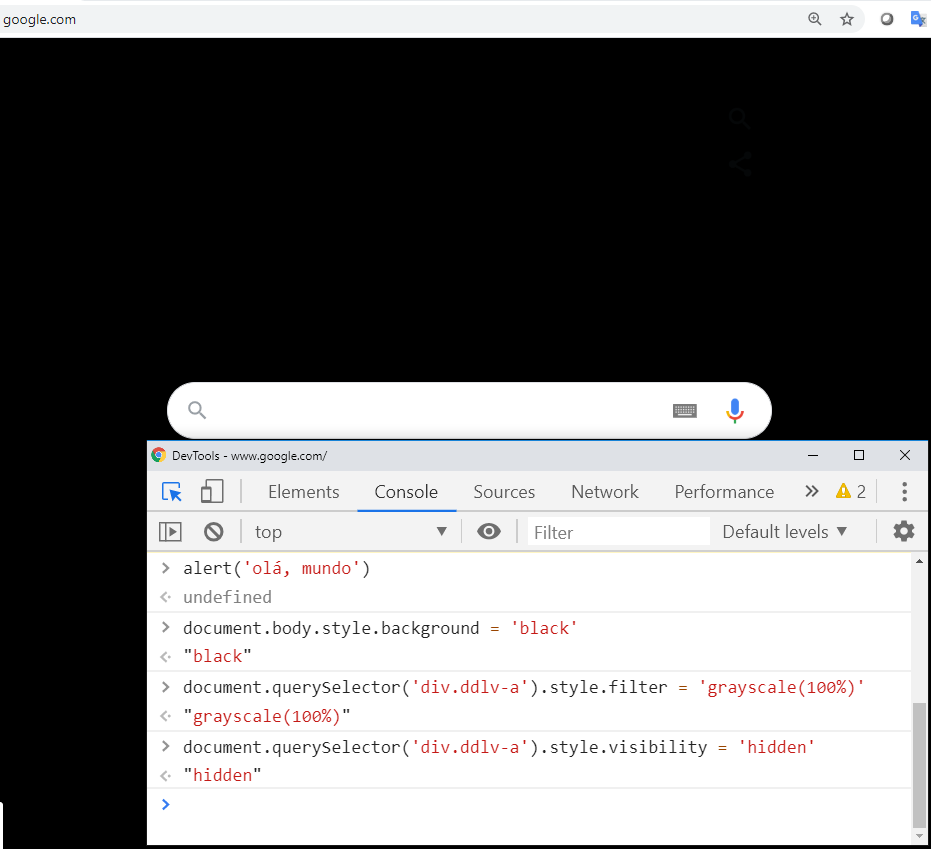


Figura 27. Página do google sem a logo

Atualizando sua página pressionando **Shift + Botão de atualizar**. Fazemos com que o navegador solicite novamente os arquivos disponibilizados no servidor da google, fazendo com que todas as alterações sejam sobescritas.

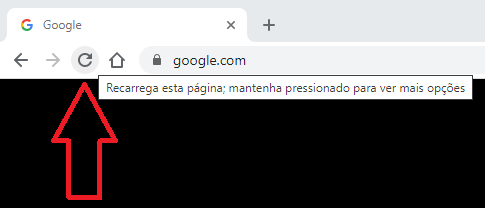


Figura 28. Atualizando a página com a tecla Shift pressionada.



Figura 29. Página do Google atualizada

# Prática 6 – Alterar o conteúdo de uma manchete

Abra uma página de notícias e selecione uma manchete. Fiz um teste utilizando o site em.com.br



Figura 30. Manchetes do Estado de Minas

Selecione o botão da setinha para selecionar elementos

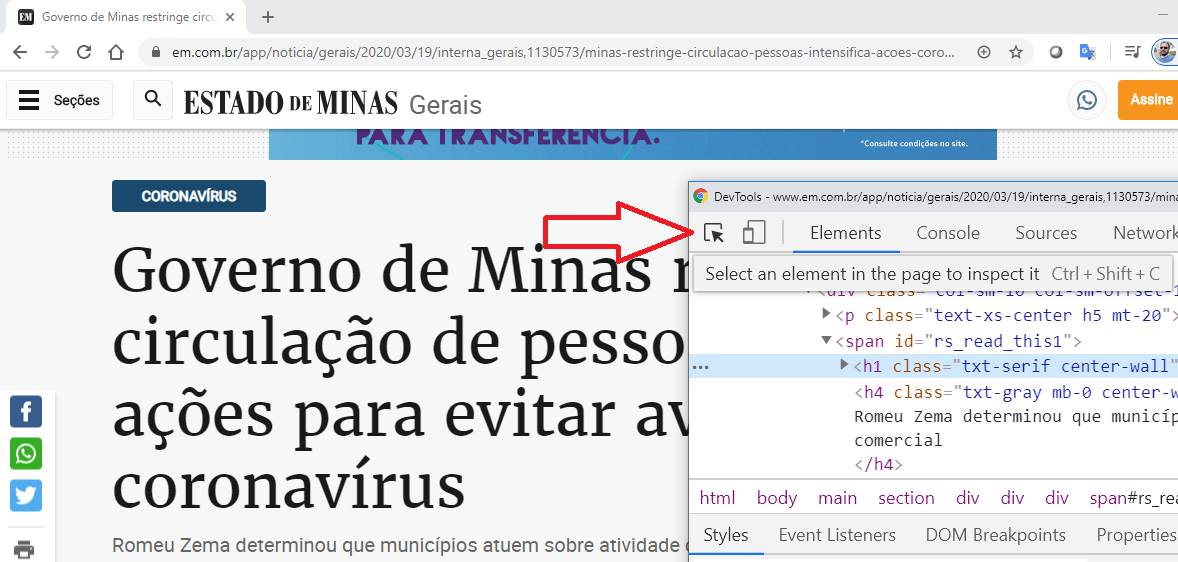


Figura 31. Select an elemenent

Passe o cursor do mouse sobre o texto e observe o nome.



Figura 32. Nome do elemento

Selecione novamente a aba Console, pressione Ctrl + L para limpar e digite o comando:

document.querySelector('h1.txt-serif.center-wall').innerText = 'SENAI Centro arrebenta!'

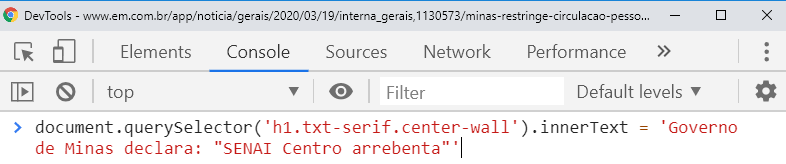


Figura 33. Comando para alterar texto de página

E esse ficou o resultado.



Figura 34. Manchete alterada

# Evolução do JavaScript

Década de 1960 – Guerra fria

Satélite Sputnik

Criação da Darpa pelo General Dwight D. Eisenhower

Arpanet – Internet

1993 – Genebra Tim Berners-Lee - CERN

HTML – HTTP – WWW

Os Co-fundadores Marc Andreessen e James Clark – MOSAIC e NETSCAP cria o em 1995 o (JavaScript)

1995 – Brendan Eich – Linguagem MOCHA – LiveScript – JAVA – JAVASCRIPT

Semelhança apenas na linguagem C que foi a mãe das duas linguagens.

Microsoft compra a Mosaic e cria o Internet Explorer

ECMA (tipo uma ISO) na Europa. ECMA Script é a versão padronizada do JavaScript.

V8 é o motor em JavaScript do Google Chrome.

Node 2010 – Máquina que roda JavaScript fora do navegador.

ECMA 1.0 – 1997

ECMA 2.0 – 1998

ECMA 3.0 – 1999

ECMA 5 – ES5 – 2009

ECMA 6 – ES6 – 2015

ES 2016 – ES 2017 – ES 2018

Tecnologias

jQuery – Conjunto de bibliotecas

Angular – Desenvolvido pela Google

V – Criado por um ex programador da Google

React – Desenvolvido pelo Facebook

# Ferramentas necessárias para o JavaScript

## Um Navegador

Google Chrome, Mozilla Firefox, Safari, Opera, ...

## Um editor de texto

Visual Studio Code, Sublime Text, Notepad++, Bloco de notas, ...

## Um Interpretador de JavaScript

Node.js, Angular.js, Vue.js, React.js

Intelicence – Auto complemento

# window.arlet

# window.confirm

# window.prompt

# Comandos básicos

## Comentário

Podemos comentar uma linha utilizando as duas barras **//** comentário

Podemos comentar um trecho do código utilizando **/\*** comentário **\*/**

## Variáveis

var nome = “Rubem”

**var** => Variável

**nome** => Identificador

**=** => Atribuição

**“Rubem”** => Valor que será atribuído dentro da variável

As **strings** (cadeia de caracteres) precisam vir entre aspas. Podemos utilizar:

**‘** – Apóstrofo ou Aspas Simples

**“** – Aspas Duplas

**`** – Crase

# Identificadores

O identificador é o **nome da variável**. Os identificadores podem começar com **letras**, **$** ou **\_**

**Exemplos:** nome, $valor, conta\_receber

O Node é **case sensitive**, portanto identificadores com letras maiúsculas ou minúsculas são diferentes.

Exemplo: **nome** é diferente de **Nome**

Diferente de muitas linguagens de programação. Os identificadores podem possuir:

Acentos e símbolos

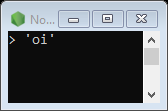
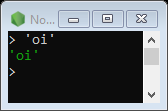
## O que não é permitido

* Não podem começar com números
* Não podem conter espaços
* Não podem ser palavras reservadas. Exemplo: alert, function, var, ...

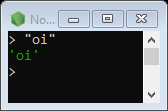
# Prática 7 – Variáveis

Abra o seu interpretador **Node.js**

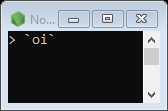
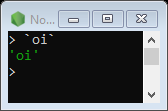
Digite **'oi'** e pressione **Enter**

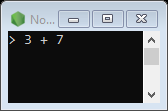
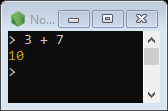
Digite **"oi"** e pressione **Enter**

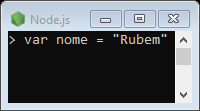
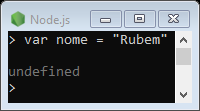
Digite **`oi`** e pressione **Enter**

Digite **3 + 7** e pressione **Enter**

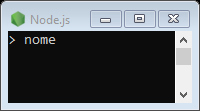
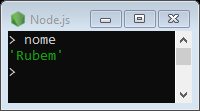
 

Digite var **nome = "Rubem"** e pressione **Enter**

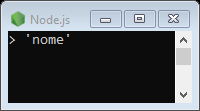
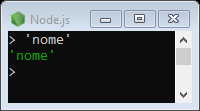
 

O node.js retornou **undefined** porque não fizemos nada com a variável

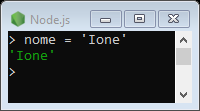
Agora digite **nome** e pressione Enter

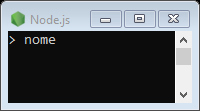
Digite **'nome'** e pressione Enter

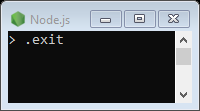
Substitua o conteúdo da variável. Digite: **nome = 'Ione'** e pressione Enter

Digite **nome** e pressione **Enter**

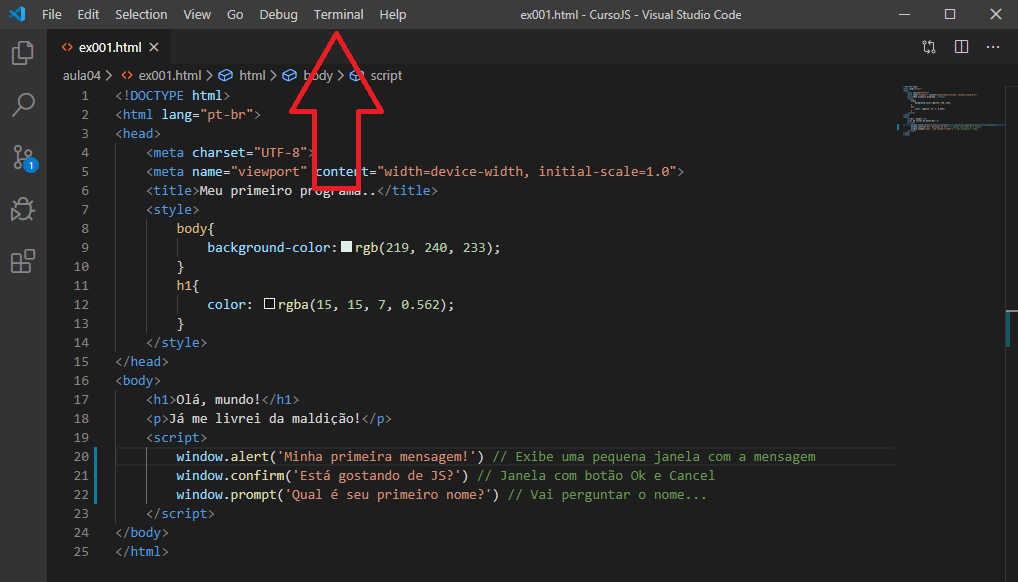
 

Para encerrar o Node.js digite .exit

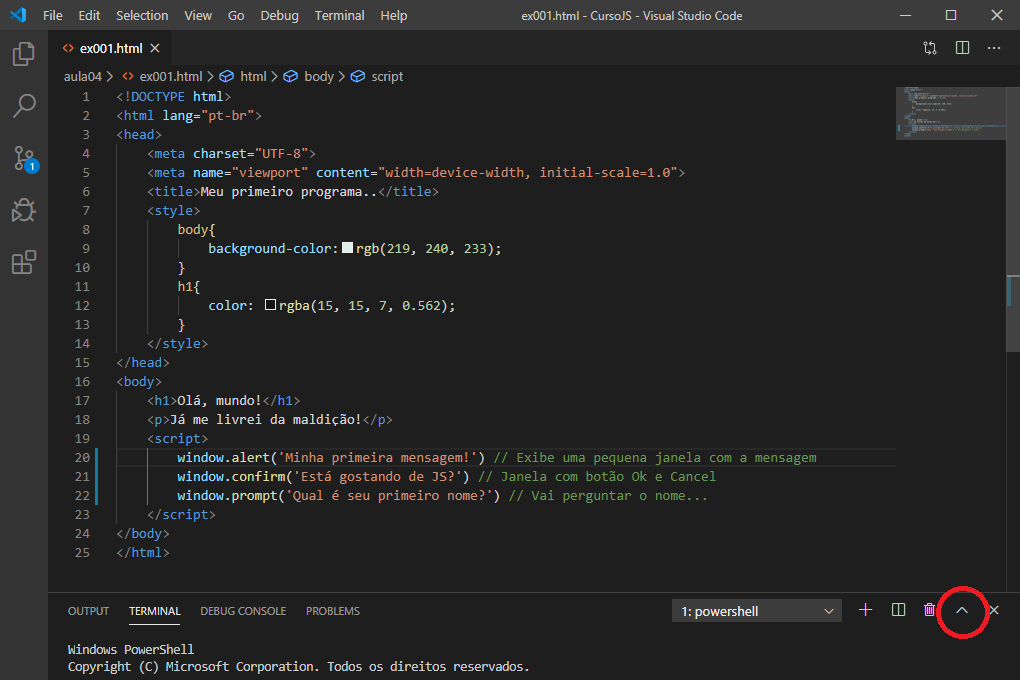


# Utilizando o Terminal no Visual Studio Code

Podemos simplesmente pressionar o botão Terminal no Visual Studio Code

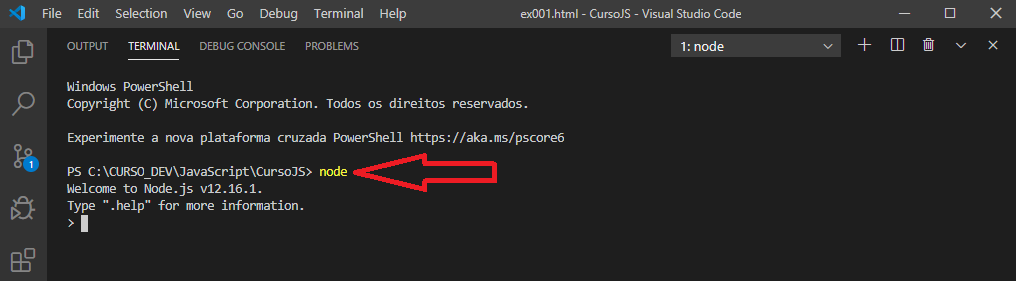


Ou utilizar o atalho **Ctrl** + **Shift** + **‘**



Podemos maximizar clicando na setinha conforme figura acima.

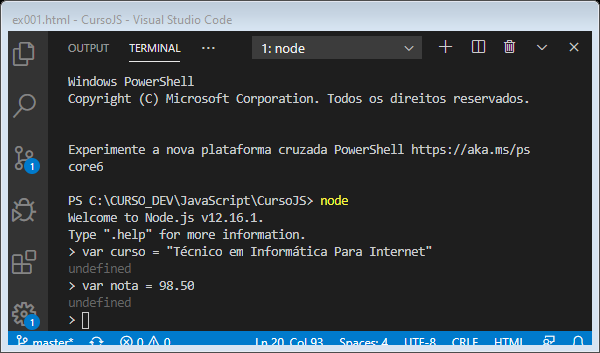
Digite node para acessar o interpretador



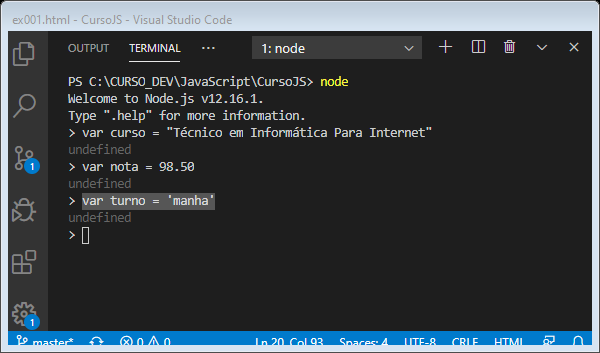
Digite **var curso = "Técnico em Informática Para Internet"** e pressione **Enter**



Digite **var nota = 98.50** e pressione **Enter**

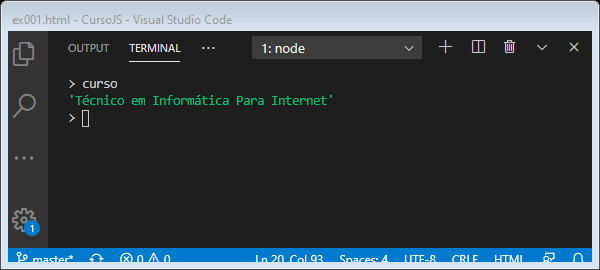


Digite var **turno = 'manha'** e pressione Enter

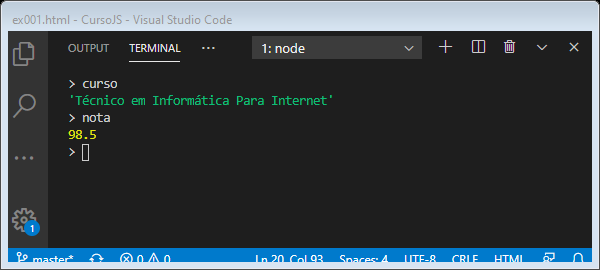


Digite o nome da variável para que o valor seja exibido.

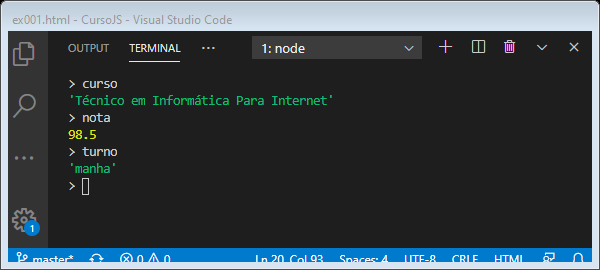
Digite **curso** e pressione **Enter**



Digite **nota** e pressione **Enter**

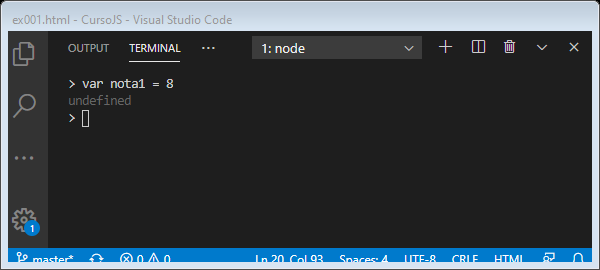


Digite **turno** e pressione **Enter**

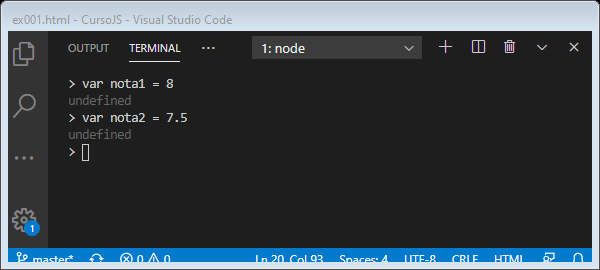


Podemos somar valores de variáveis.

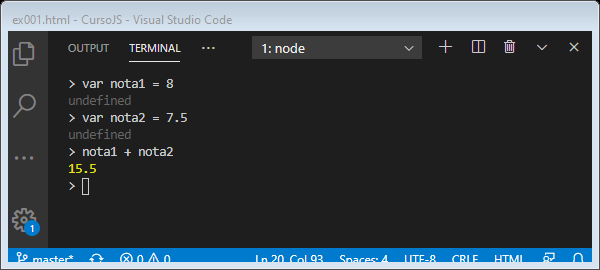
Digite **var nota1 = 8** e pressione Enter



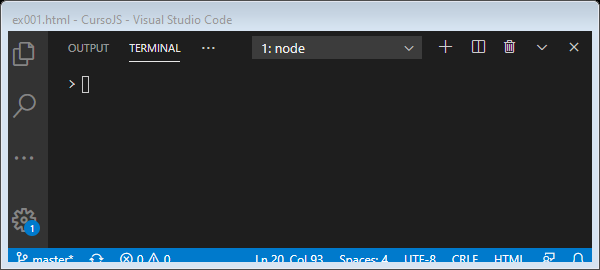
Digite var nota2 = 7.5 e pressione Enter



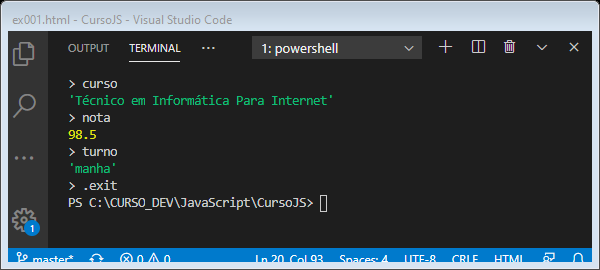
Digite nota1 + nota2 e pressione Enter



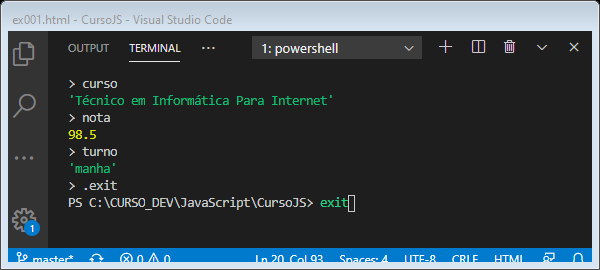
Para limpar a tela podemos digitar Ctrl + L



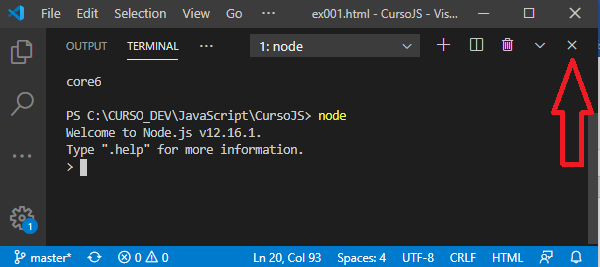
Para sair, digite **.exit** e pressione a tecla Enter



E pode fechar o terminal digitando exit



Ou podemos clicar no X



Obs. Cuidado para não abrir muitos terminais ao mesmo tempo. Isso pode deixar seu PC mais lento desnecessariamente.

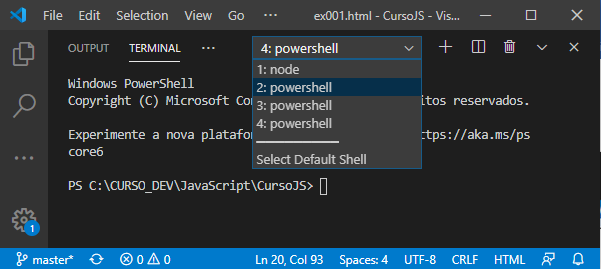
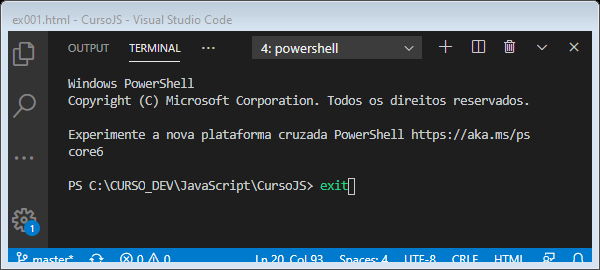


Figura x. Vários terminais abertos desnecessariamente.

Saia do terminal digitando **exit** para cada terminal aberto por engano.



## Tipos Primitivos

No JavaScript temos três tipos primitivos de variáveis.

**Number**, **String** e **Boolean**

Alguns dos tipos de variáveis

O number possui dois tipos

**Infinity**

**NaN** = Not and Number

**Boolean** = Verdadeiro ou falso, alto ou baixo, zero ou um, true ou false.

**Undefined**

**Object**

**Array**

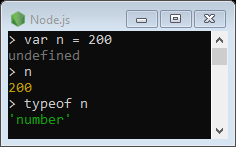
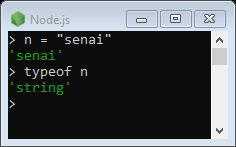
**Null** = Nulo ou vazio

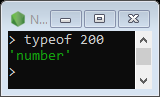
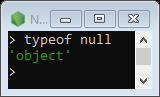
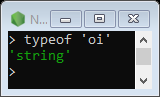
**Function**

## Typeof

Comando para saber o tipo da variável é o Typeof

Exemplo:

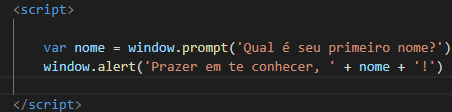
# Manipulação de Dados

Para capturar uma string em uma variável, vamos utilizar o comando **prompt**.

var nome = window.prompt('Qual é seu primeiro nome?')

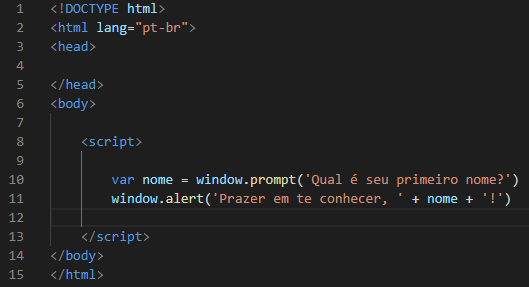
Para exibir o conteúdo de uma variável, vamos utilizar o comando **alert**

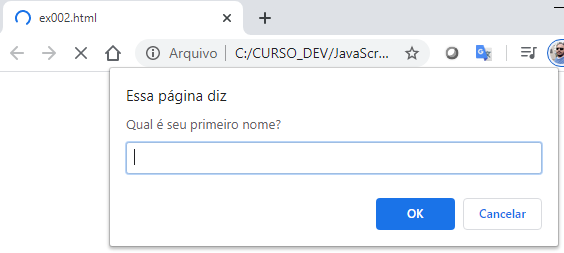
Exemplo:

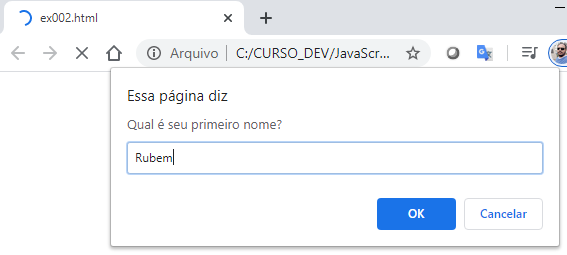


O símbulo de + aqui, serve para concatenar (juntar/ligar) o texto com o conteúdo da variável.

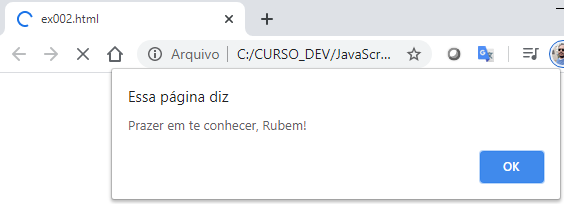
Observe a sequência:







Clicando em OK



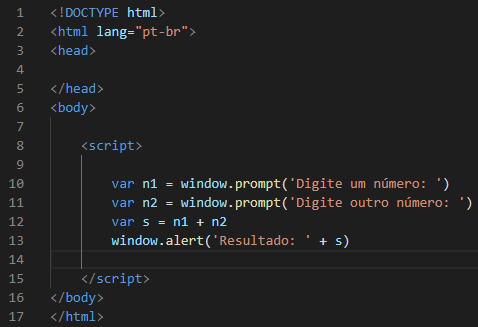
Atividade 1: Utilizando JavaScript. Crie páginas para solicitar, armazenar e exibir os dados dos usuários de acordo com as solicitações abaixo.

1. Nome
2. Sobrenome
3. Idade
4. Sexo
5. Telefone
6. Endereço
7. E-mail

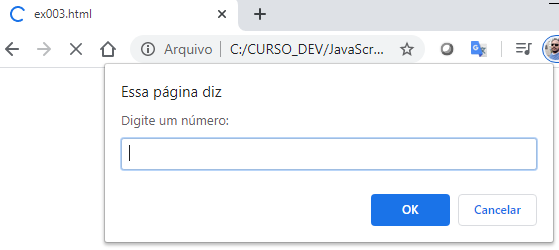
# Convertendo String para Number

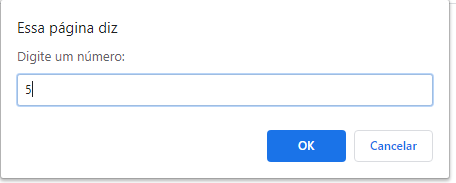
Toda vez que utilizamos o prompt para receber um valor digitado pelo usuário. O JavaScript recebe esses dados como sendo uma string.

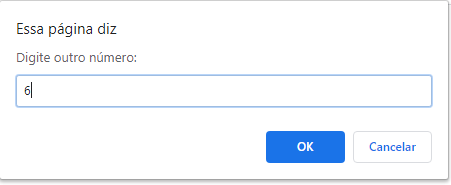
Observe o código de exemplo:



Observe o resultado obtido:









Como podemos visualizar. O resultado foi 56, quando na verdade esperávamos que fosse 11. Pois 5 + 6 são 11.

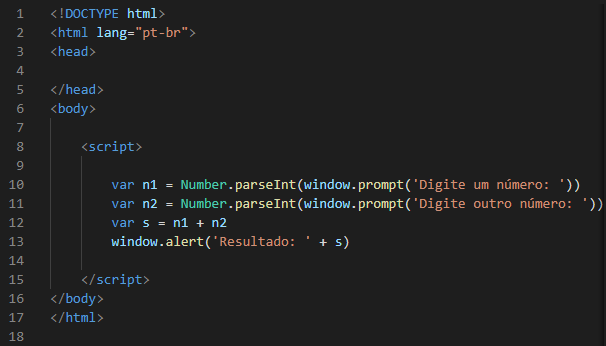
Para realizarmos cálculos, precisamos converte-los para number.

## Convertendo String para número inteiro

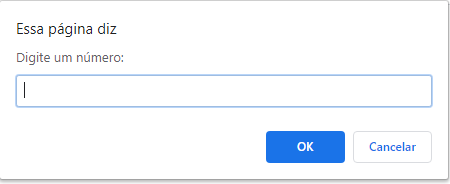
Number.parseInt(n)

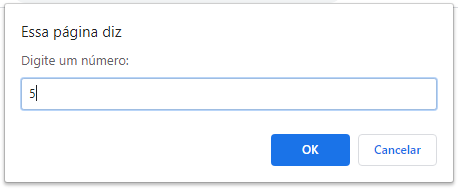
Obs. O **I** precisa ser maiúsculo e o restante minúsculo.

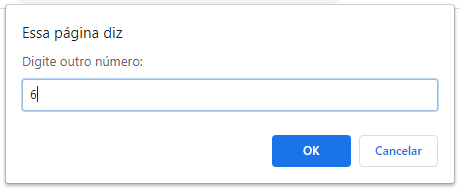
Exemplo: Vamos somar dois números e exibir o resultado.



Agora observe o resultado:







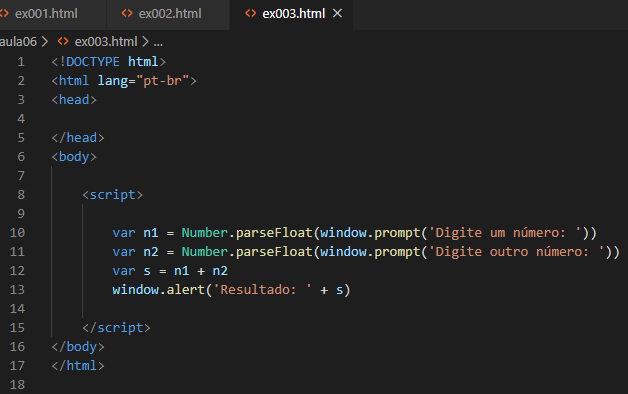


## Convertendo String para número real, flutuante

Number.parseFloat(n)

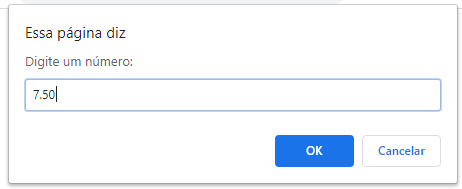
Obs. O **F** precisa ser maiúsculo e o restante minúsculo.

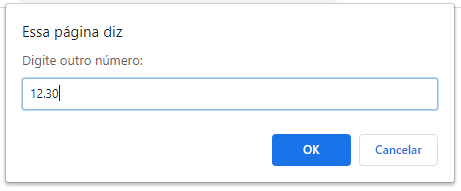
Exemplo: Vamos somar dois números e exibir o resultado.



Agora observe o resultado:







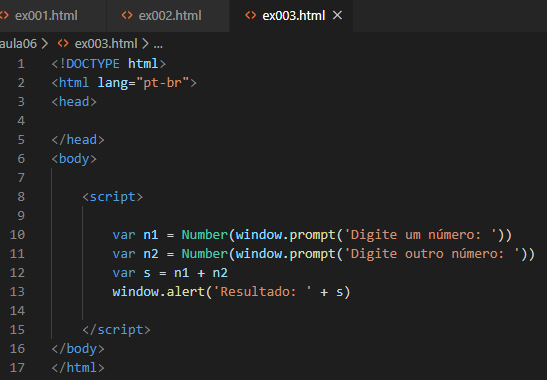


## Convertendo String para número indefinido

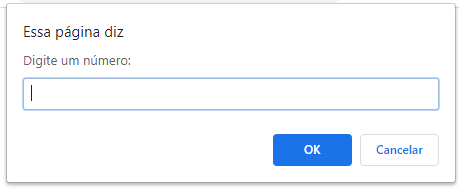
Number (n)

Obs. O JavaScript identifica o tipo, se é float ou integer.

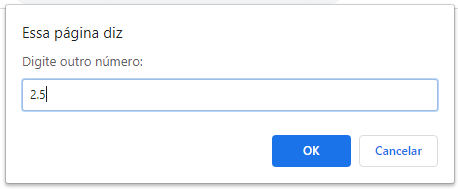
Exemplo: Vamos somar dois números e exibir o resultado.



Agora observe o resultado:









Atividade 2: Utilizando JavaScript. Faça uma página que solicite do usuário dois valores e exiba o resultado para as operações abaixo.

1. Soma
2. Subtração
3. Multiplicação
4. Divisão

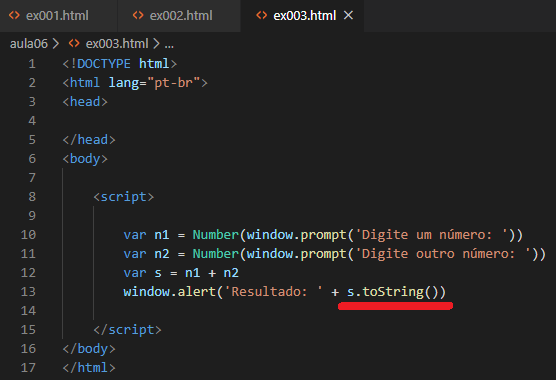
## Convertendo Número para String

Temos duas formas de se converter dados do tipo number para string.

String(n)

n.toString()

Exemplos:





# Template Strings (novidade do EcmaScript)

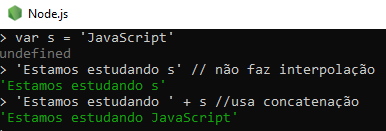
Para encurtar o texto na hora de se trabalhar texto com conteúdo de variável, podemos utilizar o Template String. Para isso precisamos utilizar a **crase `**

**Forma convencional**

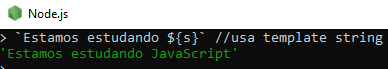
var s = 'JavaScript'

'Estamos estudando s' // não faz interpolação

'Estamos estudando ' + s //usa concatenação

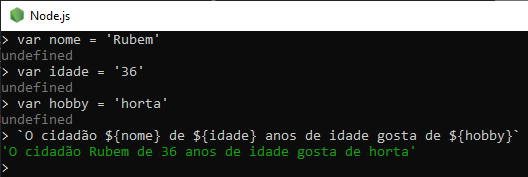


`Estamos estudando ${s}` //usa template string

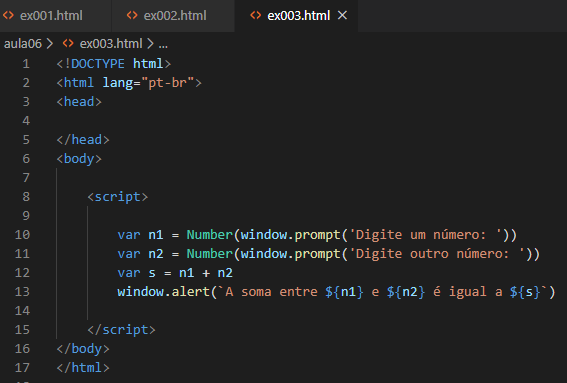


Com a utilização do **placeholder** ( **$** ) economizamos espaço no código.

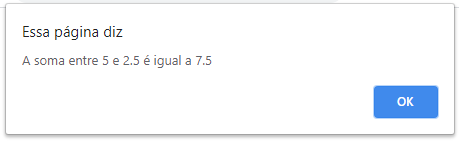
Exemplo:



Melhorando o código anterior:



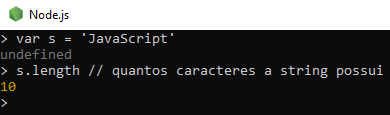
Resultado da soma dos números 5 e 2.5



# Formatando Strings

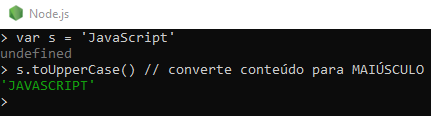
## length

Podemos utilizar o s.length para saber quantos caracteres uma string possui



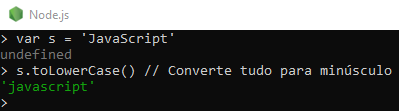
## toUpperCase()

Convertendo a string para caixa alta

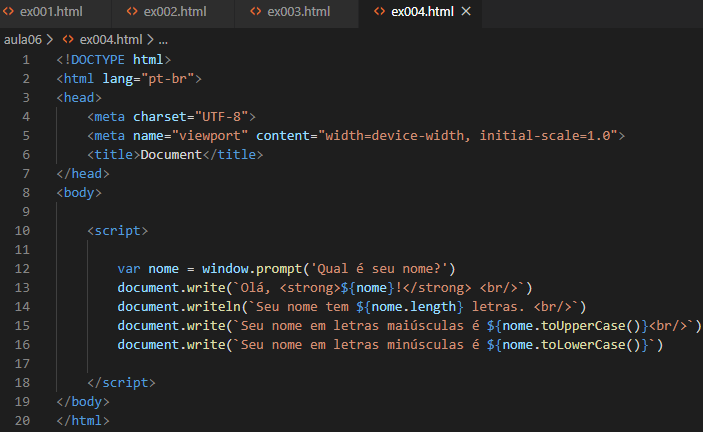


## toLowerCase()

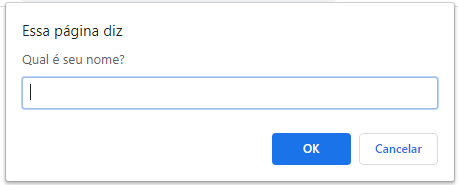
Convertendo a string para caixa baixa

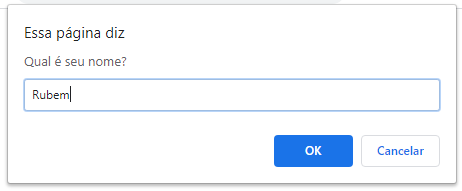


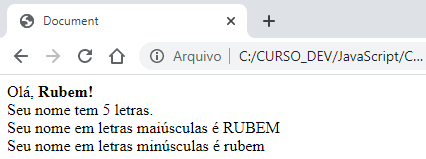
Exemplo:



Observando o resultado







# Formatando Números

## toFixed()

Acrescentando casas decimais

## toFixed().replace()

Altera o ponto para virgula

## toLocaleString()

Para colocar informações de moeda como:

Real:

toLocaleString('pt-BR', {style:'currency', currency: 'BRL'})

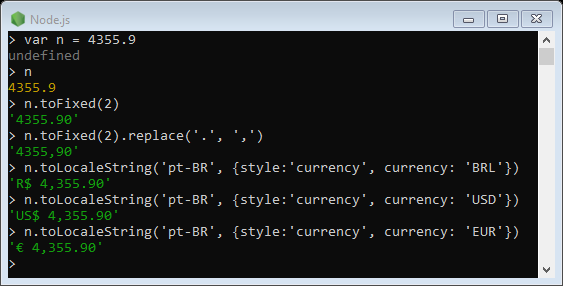
Dolar:

toLocaleString('pt-BR', {style:'currency', currency: 'USD'})

Euro:

toLocaleString('pt-BR', {style:'currency', currency: 'EUR'})

Exemplo:



# Atividade

# Operadores do JS

## Operadores Aritméticos

+ Adição

Exemplo: 5 + 2 = 7

- Subtração

Exemplo: 5 – 2 = 3

\* Multiplicação

Exemplo: 5 \* 2 = 10

/ Divisão

Exemplo: 5 / 2 = 2.5

% Módulo - Resto da divisão

Exemplo: 5 % 2 = 1

\*\* - Potência

Exemplo: 5 \*\* 5 = 25

## Ordem de precedência

No Java Script os operadores aritméticos seguem a ordem de precedência da matemática.

**( )** – Parênteses

**\*\*** – Potências

**\* / %** – Multiplicação, divisão e módulo

**+ -** – Adição e subtração

# Atividade

Utilizando JavaScript, resolva as questões abaixo e observe cada resultado:

var a = 5 + 3

var b = a % 5

var c = 5 \* b \*\* 2

var d = 10 – a / 2

var e = 6 \* 2 / d

var f = b % e + 4 / e

# Operadores de atribuição

# Auto atribuição ou auto referência

É quando uma variável recebe ela mesma e mais alguma outra atribuição.

Exemplo

var n = 3

**Auto atribuição Auto atribuição simplificada**

n **=** n **+** 4 n **+=** 4

n **=** n **–** 5 n **-=** 5

n **=** n **\*** 4 n **\*=** 4

n **=** n **/** 2 n **/=** 2

n **=** n **\*\*** 2 n **\*\*=** 2

n **=** n **%** 5 n **%=** 5

# Incremento e decremento, pré-incremento

Exemplo: var x = 5

x = x + 1 ou x += 1 ou x++ ou ++x

x = x – 1 ou x -= 1 ou x-- ou --x

# Operadores Relacionais

**>** Maior Exemplo: 5 > 2 true

**<** Menor Exemplo: 7 < 4 false

**>=** Maior ou igual Exemplo: 8 >= 8 true

**<=** Menor ou igual Exemplo: 9 <= 7 false

**==** Igual Exemplo: 5 == 5 true

**===** Identidade Exemplo: 5 === ‘5’ false

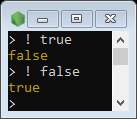
**!=** Diferente Exemplo: 4 != 4 false

**!==** Desigual restrito Exemplo 5 !== ‘5’ true

# Operadores Lógicos

**!** NOT NÃO Negação Operador unário

|  |  |  |
| --- | --- | --- |
| **!** | true | false |
| **!** | false | true |

****

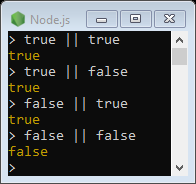
**&&** AND E Conjunção Operador binário

|  |  |  |  |
| --- | --- | --- | --- |
| true | **&&** | true | true |
| true | **&&** | false | false |
| false | **&&** | true | false |
| false | **&&** | false | false |

****

**||** OR OU Disjunção

|  |  |  |  |
| --- | --- | --- | --- |
| true | **||** | true | true |
| true | **||** | false | true |
| false | **||** | true | true |
| false | **||** | false | false |



Ordem de precedência

|  |  |  |
| --- | --- | --- |
| **( )** | **\*\*** | **/** |
| **>** | **<** | **>=** |
| **!** |  |  |
| **&&** |  |  |
| **||** |  |  |

Obs. Operadores relacionais não tem ordem de precedência. Quem aparecer primeiro será realizado primeiro. Esquerda para a Direita.

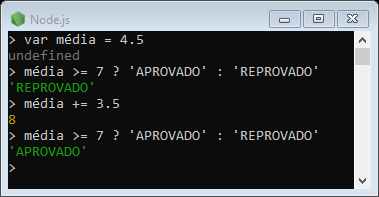
# Operadores Ternário

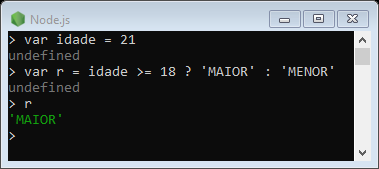
**? :** - Dentro de uma mesma expressão

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **teste** | **?** | **true** | **:** | **false** |

**Exemplo:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **média >= 7** | **?** | **"Aprovado"** | **:** | **"Reprovado"** |





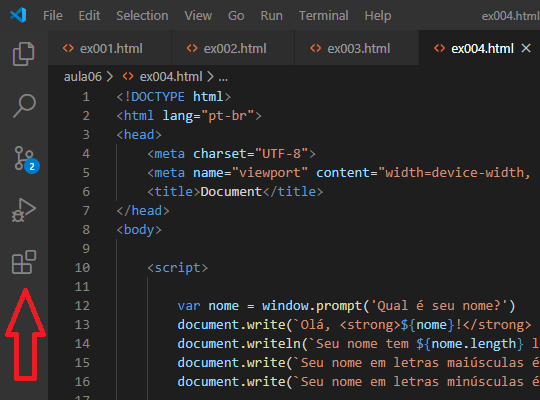
# Entendendo o DOM

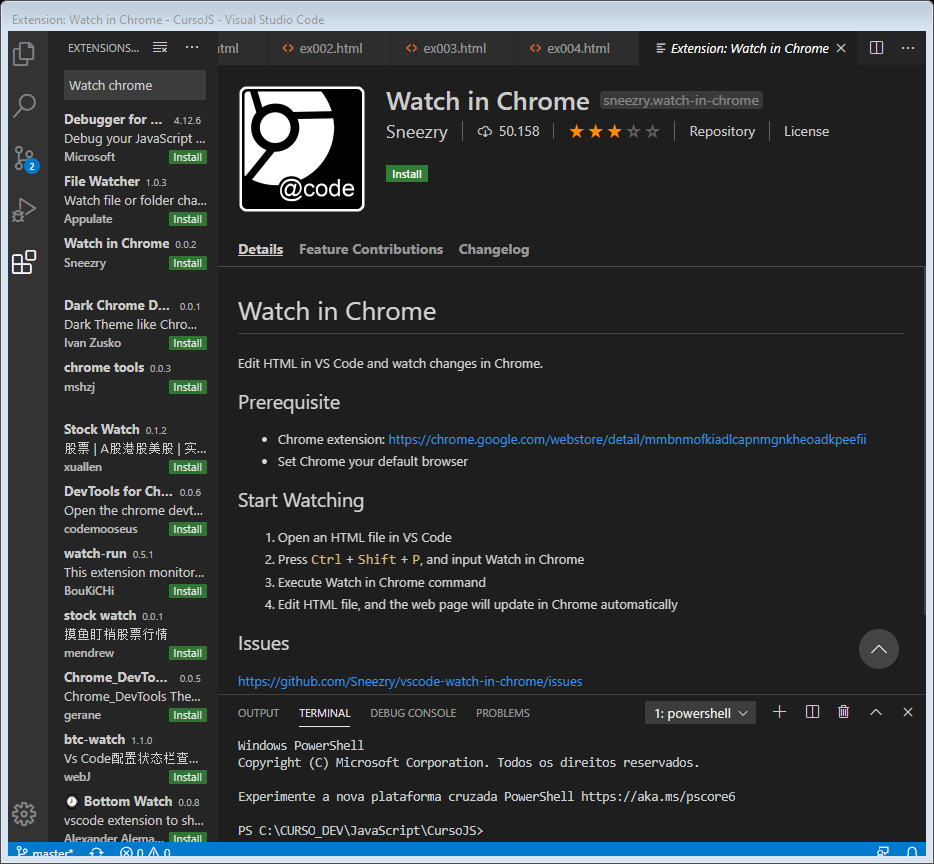
DOM é um acrônimo para Document Object Model, é um modelo de objetos para documentos. É um conjunto de objetos do navegador (cliente) que dão acesso ao servidor de Web Sites (server).

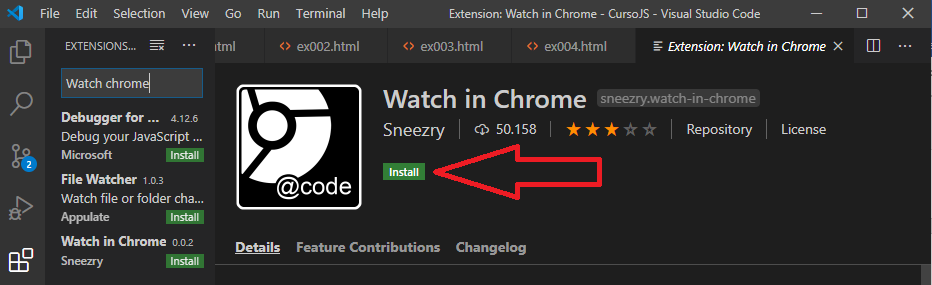
# Extensões necessárias

# Watch in Chorme

Abra o Visual Studio Code, clique no ultimo ícone da esquerda “Extensões” e digite **Watch in Chrome**

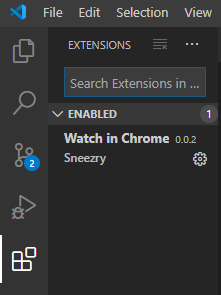




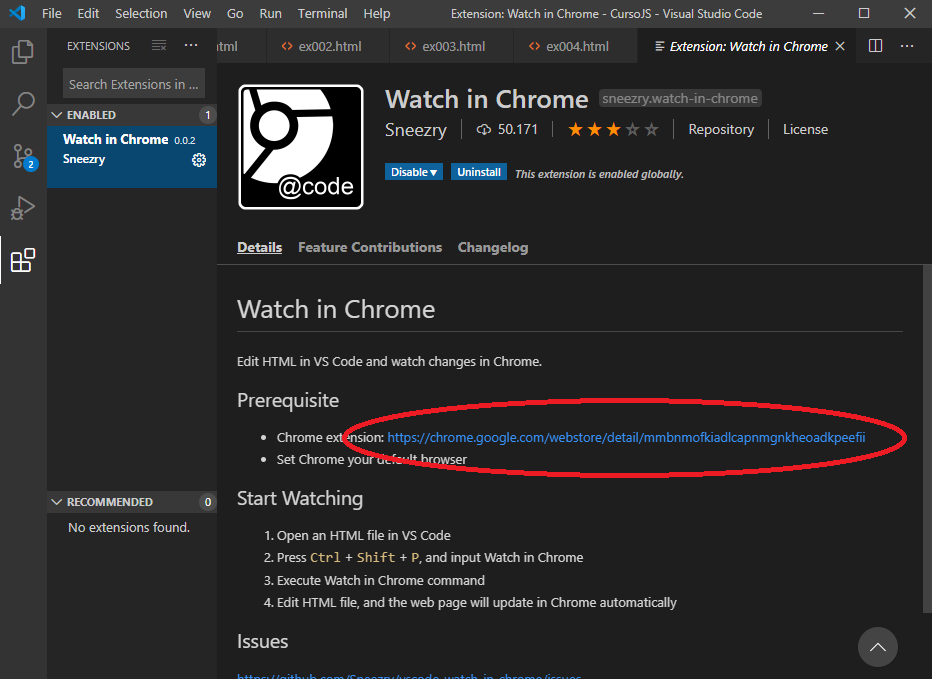


# Instalando Plugin no Chrome

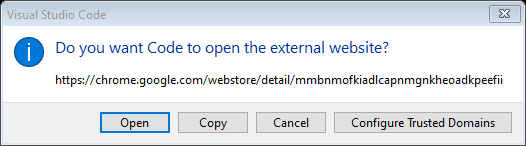
Feche o Visual Studio Code e reabra. Veja que o Watch in Chrome já consta instalado.



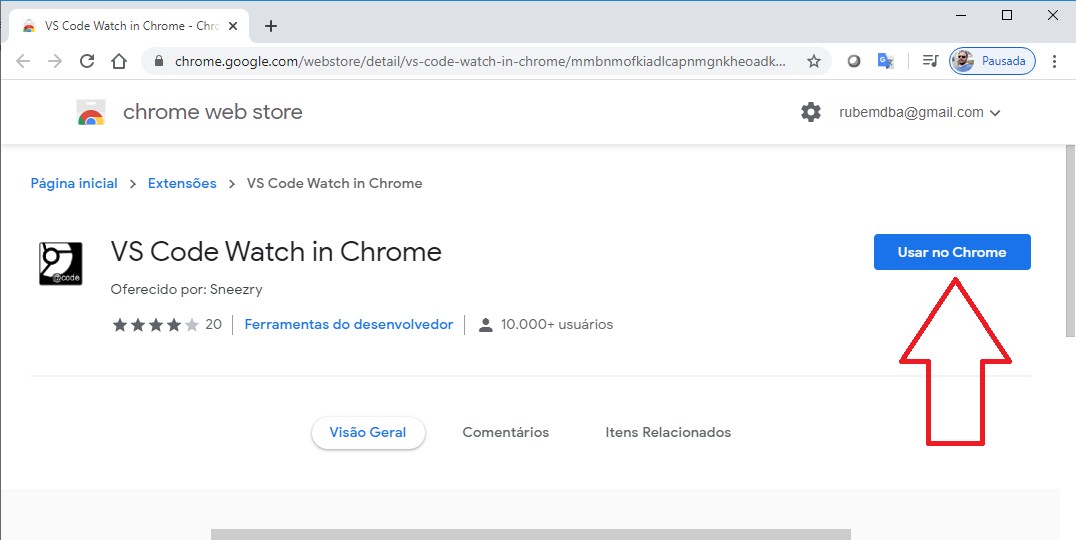
Clique sobre ele e clique no link que aparece abaixo de Prerequisite



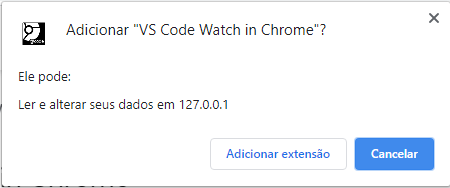
Permita que o Visual Studio Code abra o Browser



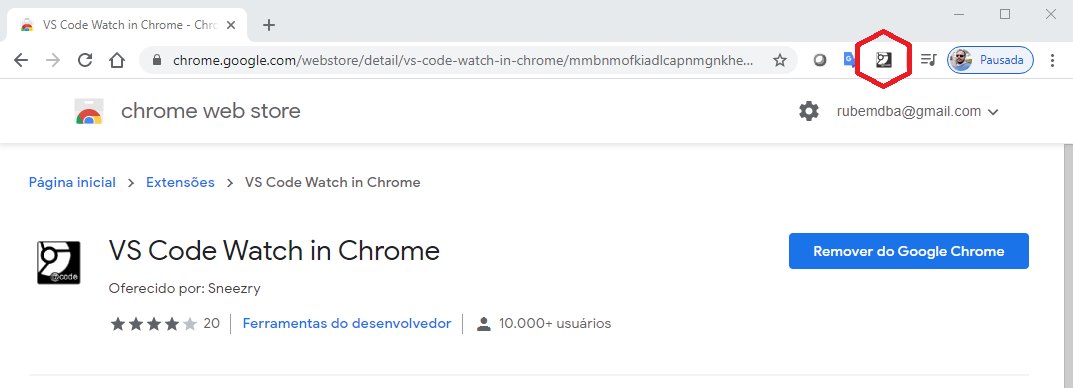
Clique em Add to Chrome



Dê permissão para a instalação clicando em **Adicionar extensão**

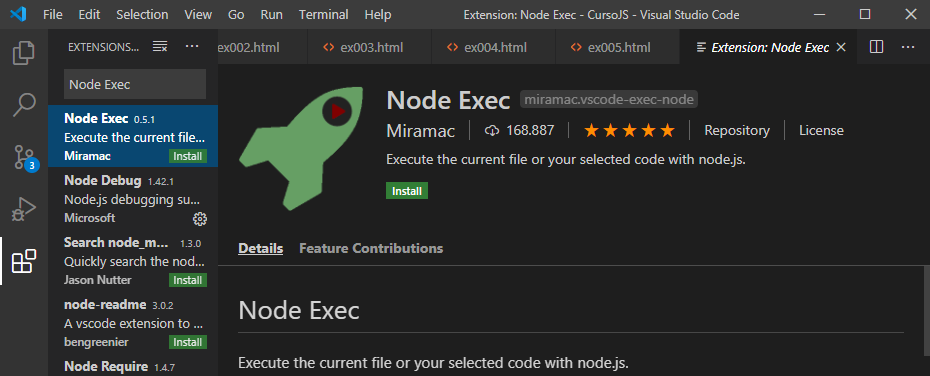


E a extensão irá aparecer no canto do seu navegador



# Extensão Node Exec

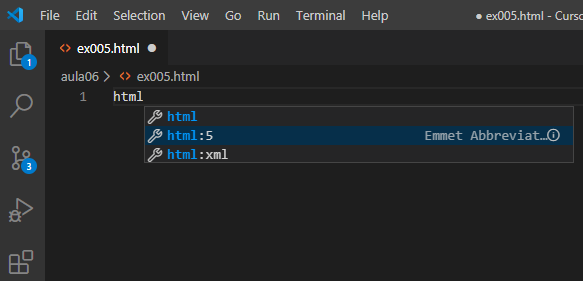
Instale a extensão Node Exec para habilitar a tecla F8 para rodar o JavaScript pelo Node.

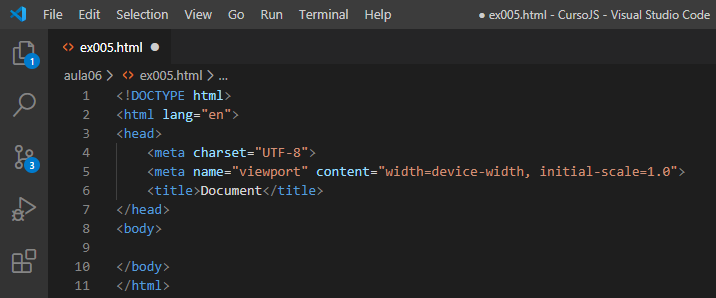


Reinicie o Visual Studio Code para que ele fique pronto para uso.

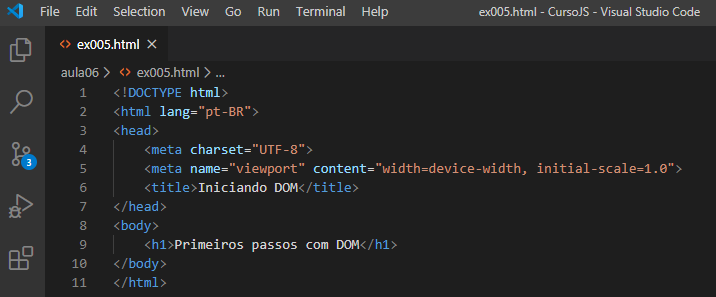
# Praticando

Crie um documento com o nome ex01.html digite html e clique em html5

O Visual Studio Code irá preencher o escopo principal do html.



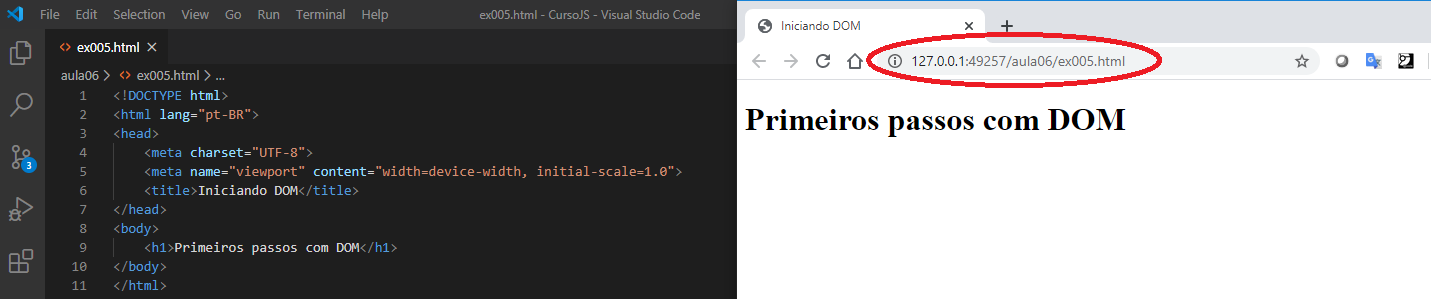
Altere o idioma do lang, o title e insira um texto no corpo da página.



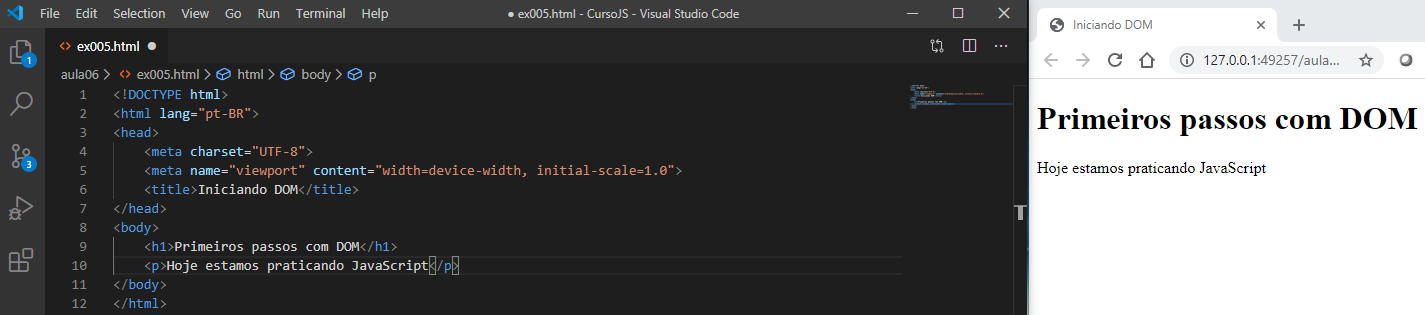
Pressione o atalho Ctrl + Shift + P e digite >whatch in Chrome para que o Visual Studio já exiba o resultado no navegador.



O navegador irá simular um servidor web para exibir seu código.

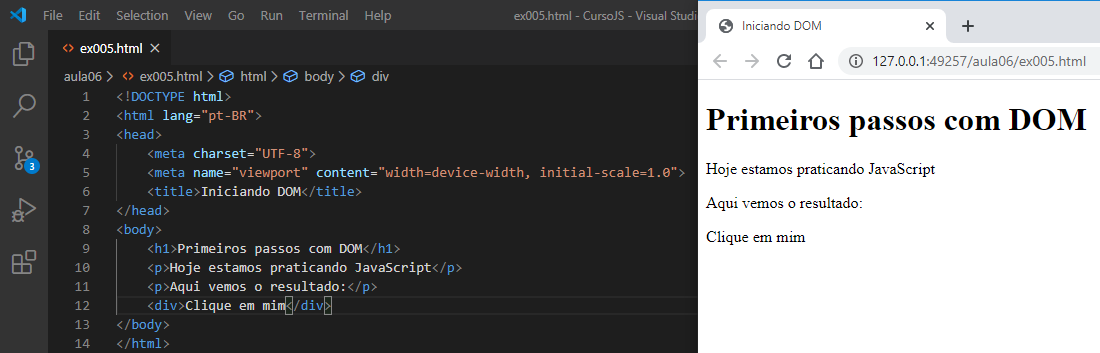


Dessa forma, a medida que formos trabalhando no código. O resultado já é atualizado e executado na hora.

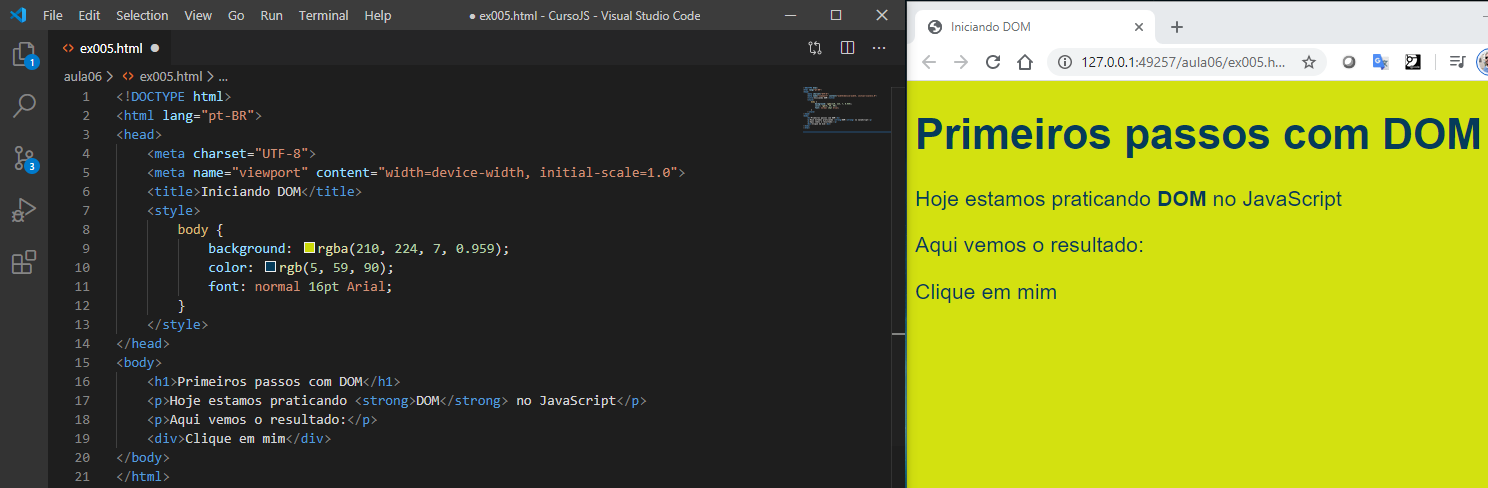


Caso o sincronismo pare de funcionar, basta fechar o navegador e reabri-lo com o atalho Ctrl + Shift + P e digitar >whatch in Chrome.

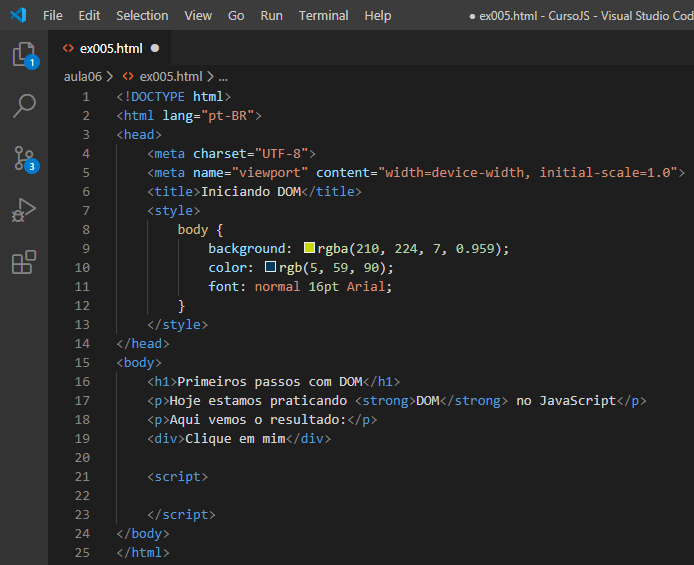
Vamos acrescentar um parágrafo e uma div conforme exemplo abaixo



Vamos colocar um estilo na página só para deixar um pouco mais atraente.



Agora vamos colocar um trecho com JavaScript no final da body



# Arvore DOM

Arvore DOM é uma representação diagramática que demostra o relacionamento hierárquico entre os objetos do web site. São classificados através do relacionamento direto entre o parenty (pai ou mãe) e o chield (filho).

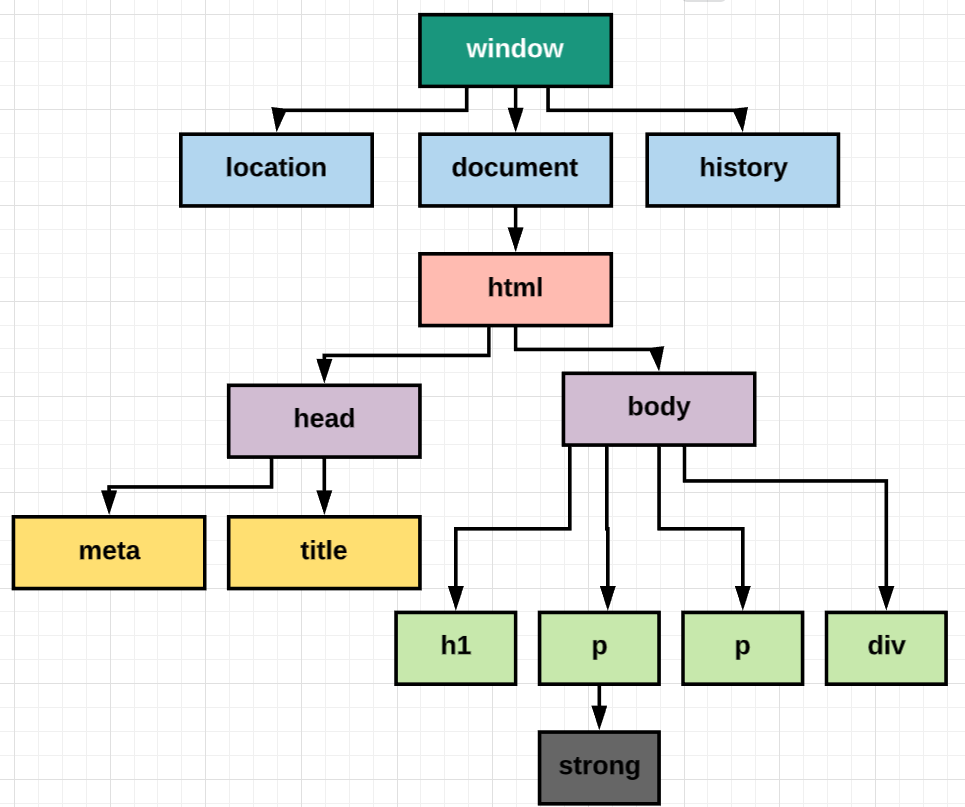
window – Objeto chamado janela.

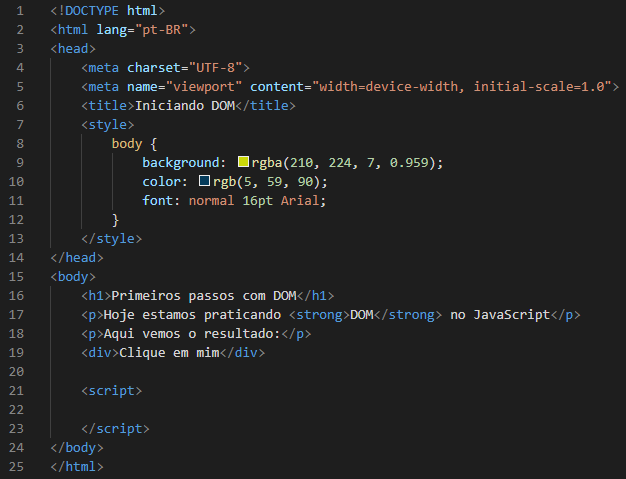
location – Qual é a localização da URL. Página atual, página anterior,...

document – O documento atual. Possui o html que possui o head e o body

history – Armazenamento de histórico de navegação. De onde veio e para onde vai.

Veja abaixo, a arvore DOM do site criado acima.





# Selecionando

Podemos selecionar os objetos existente dentro da arvore DOM de cinco maneiras.

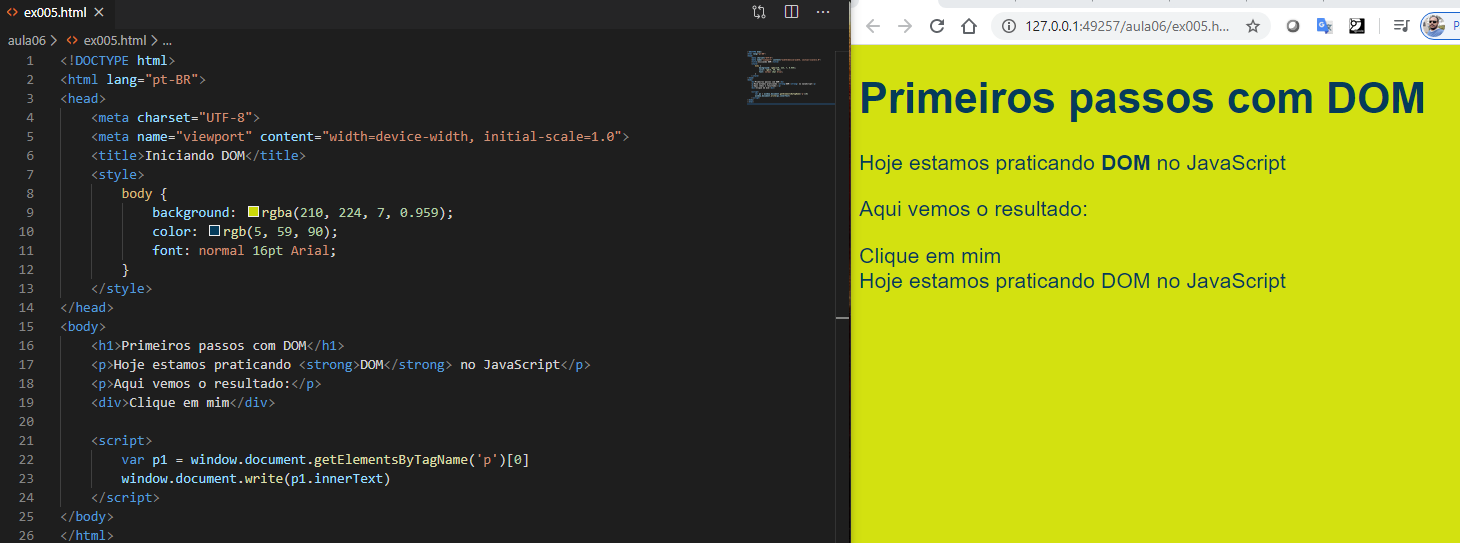
## por Marca (Tag)

getElementsBy TagName()

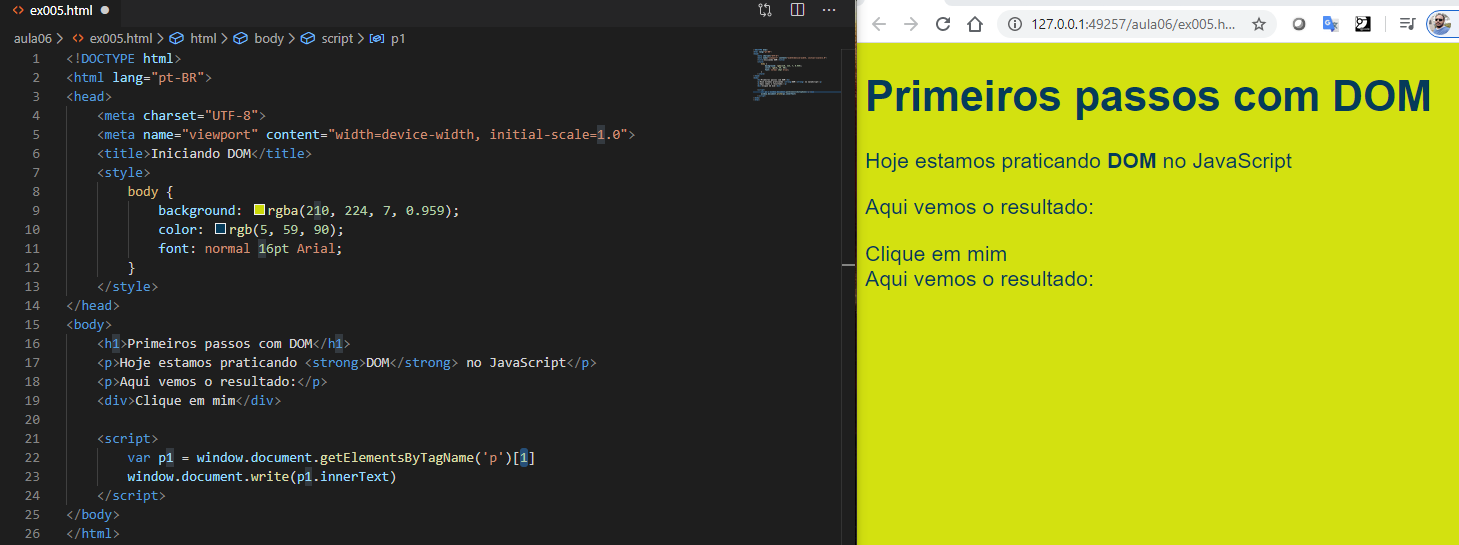


Como o código possui mais de uma marca p, precisamos selecionar qual deles com o colchetes [0] para o primeiro, [1] para o segundo e assim por diante.

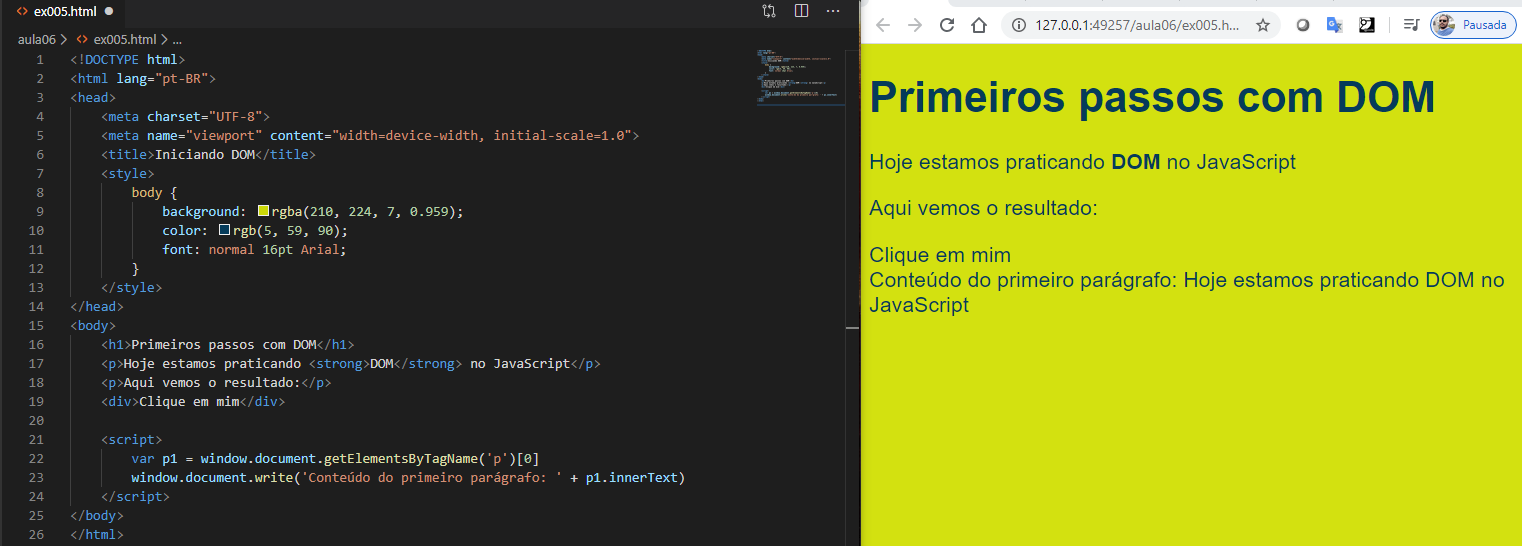
Observe:



Se eu mudo para o elemento [1]



Podemos concatenar um texto com o conteúdo do elemento.



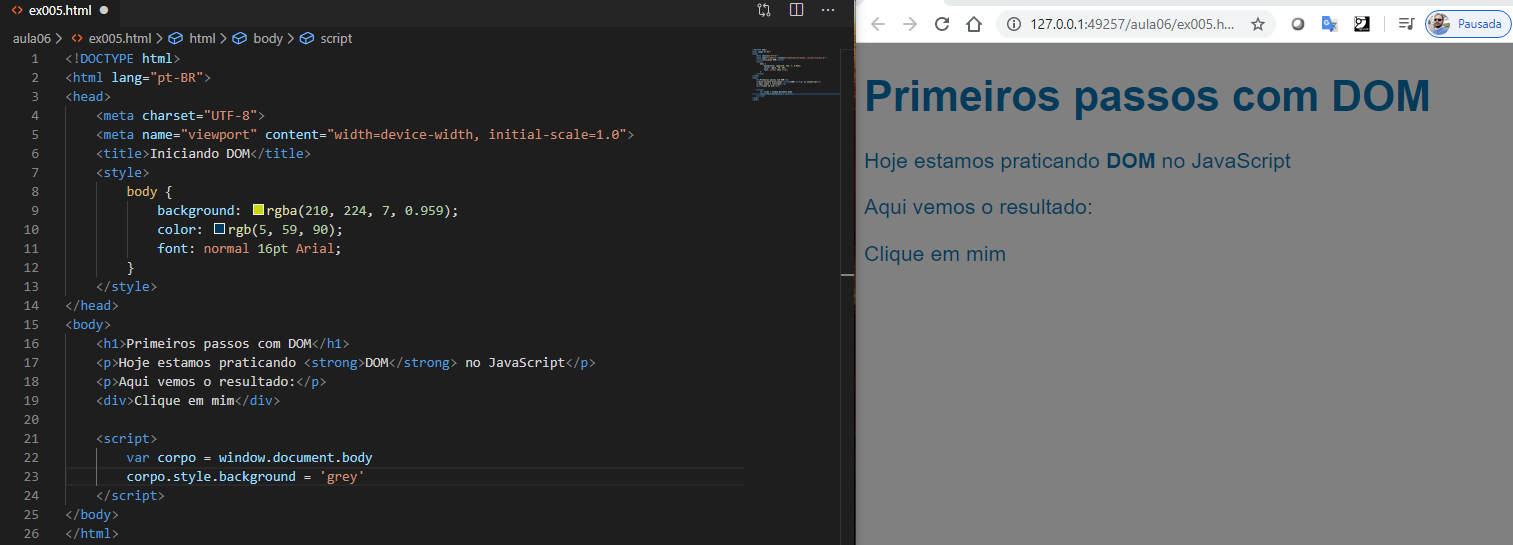
Reparem que o innerText captura apenas o texto, mas o Strong do DOM não foi exibido. Se eu mudar para innerHTML, temos o efeito do Strong sendo exibido.

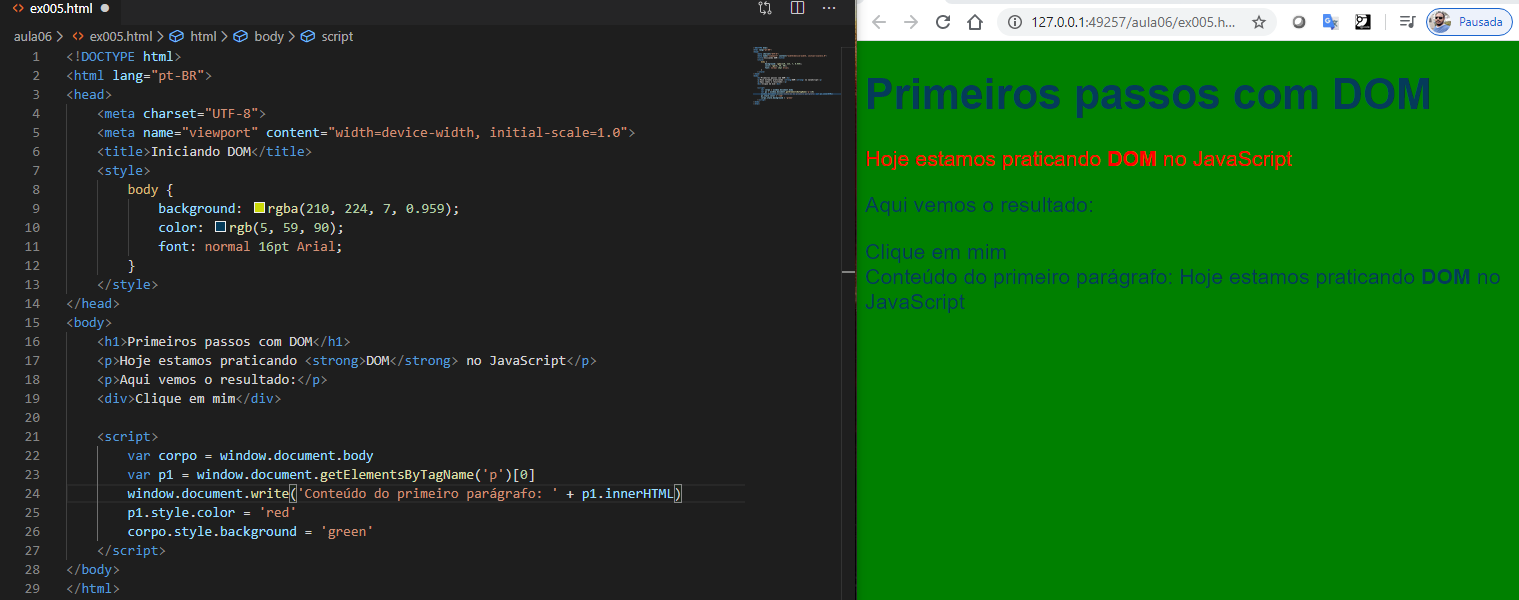


Podemos alterar configurações do elemento também.

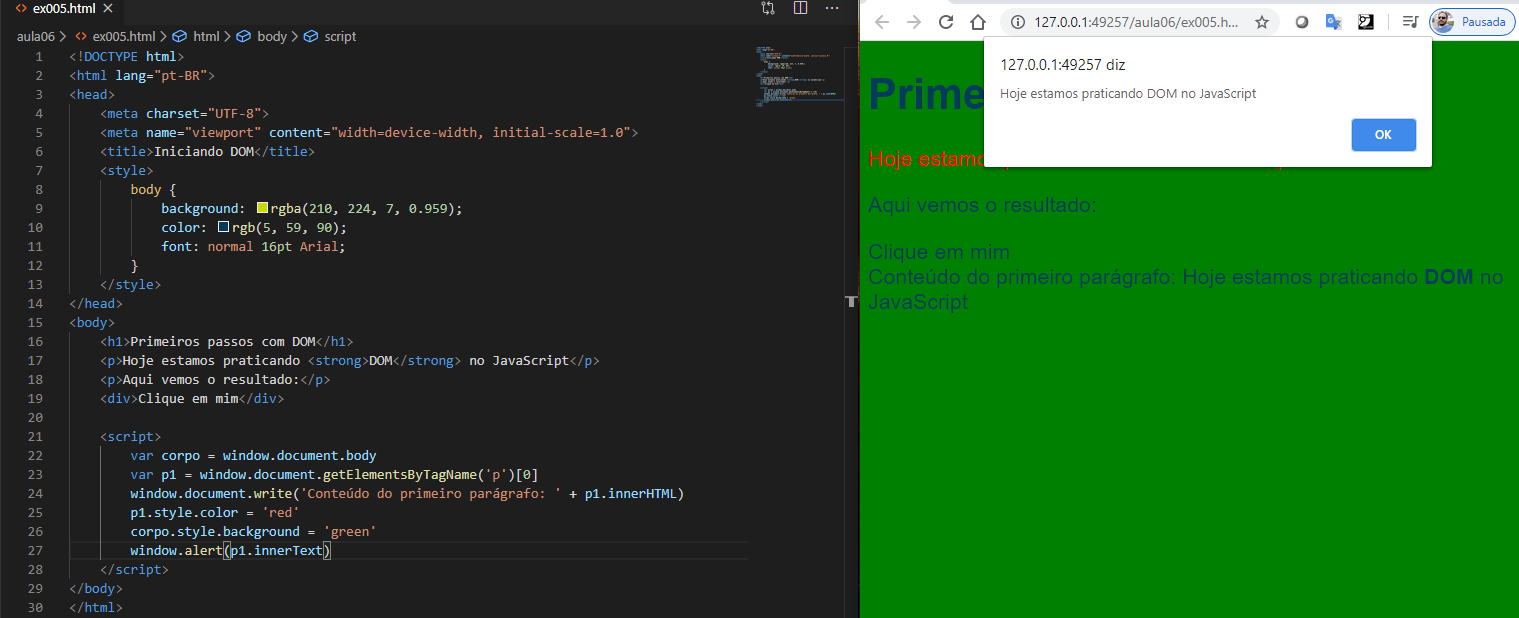


Podemos criar uma variável para o corpo e mandar alterar a cor do fundo:





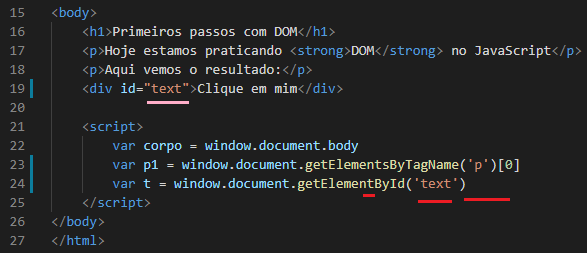
Podemos utilizar o alert também



## por ID

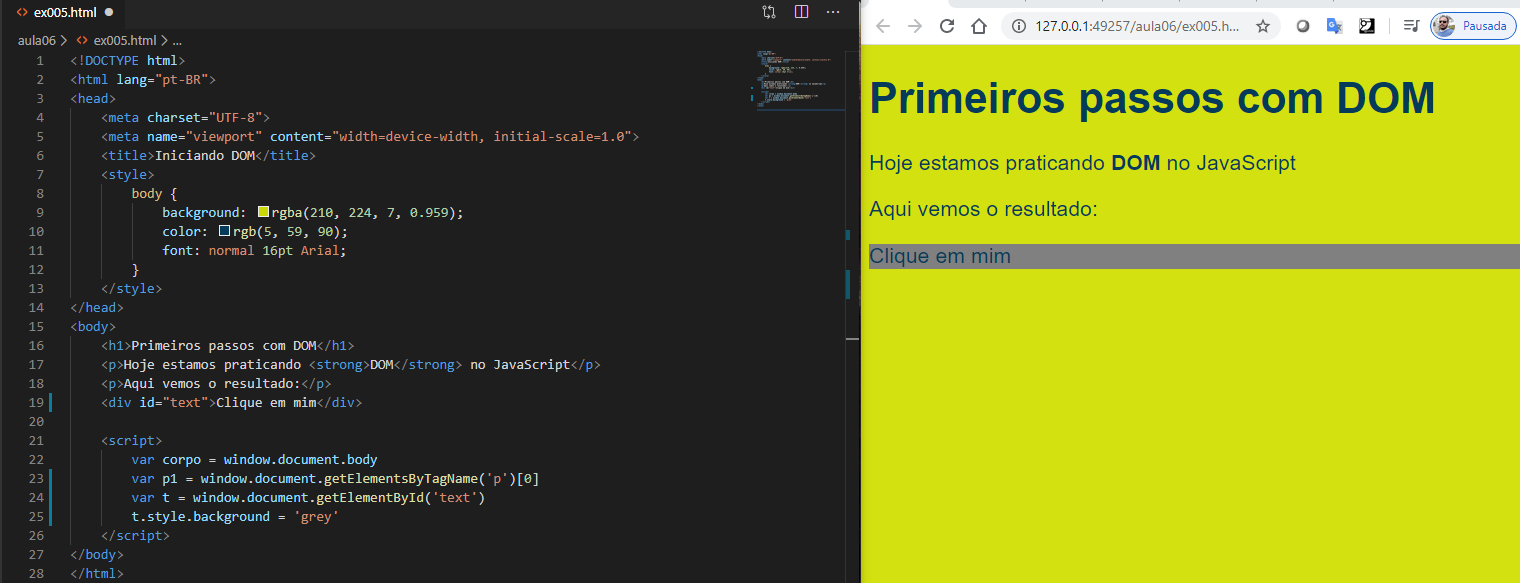
getElementById()

Para exemplificar. Vamos dar um id para nossa div. Vamos chamar de text

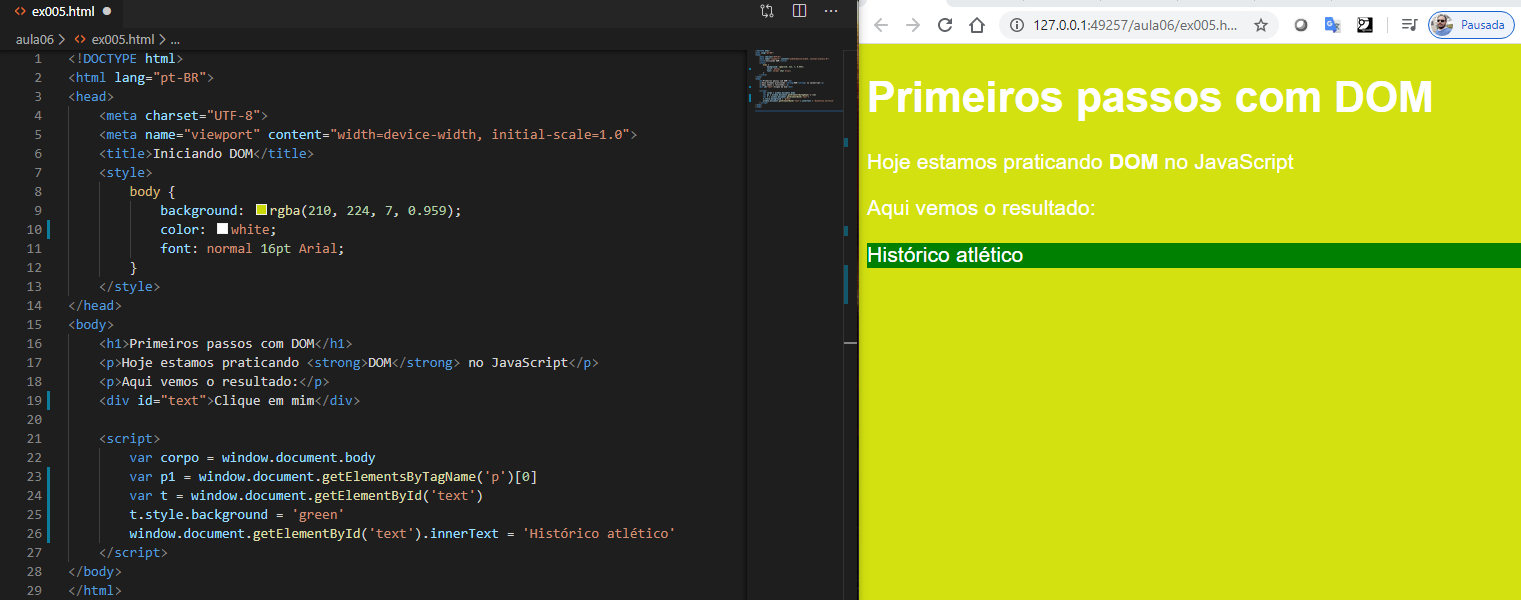


Observem que não precisamos colocar os colchetes porque o getElement está no singular.

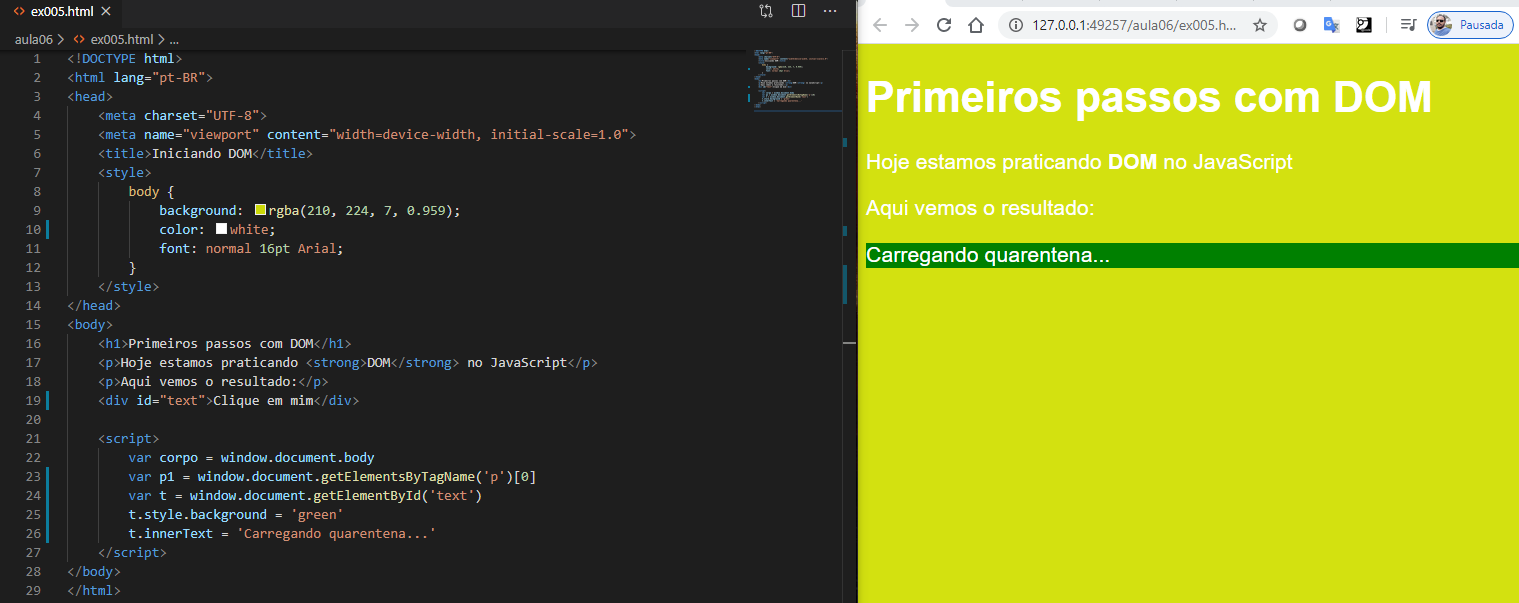
Acrescentando um estilo na variável t que criamos, temos esse resultado.



Podemos modificar o texto com o comando completo



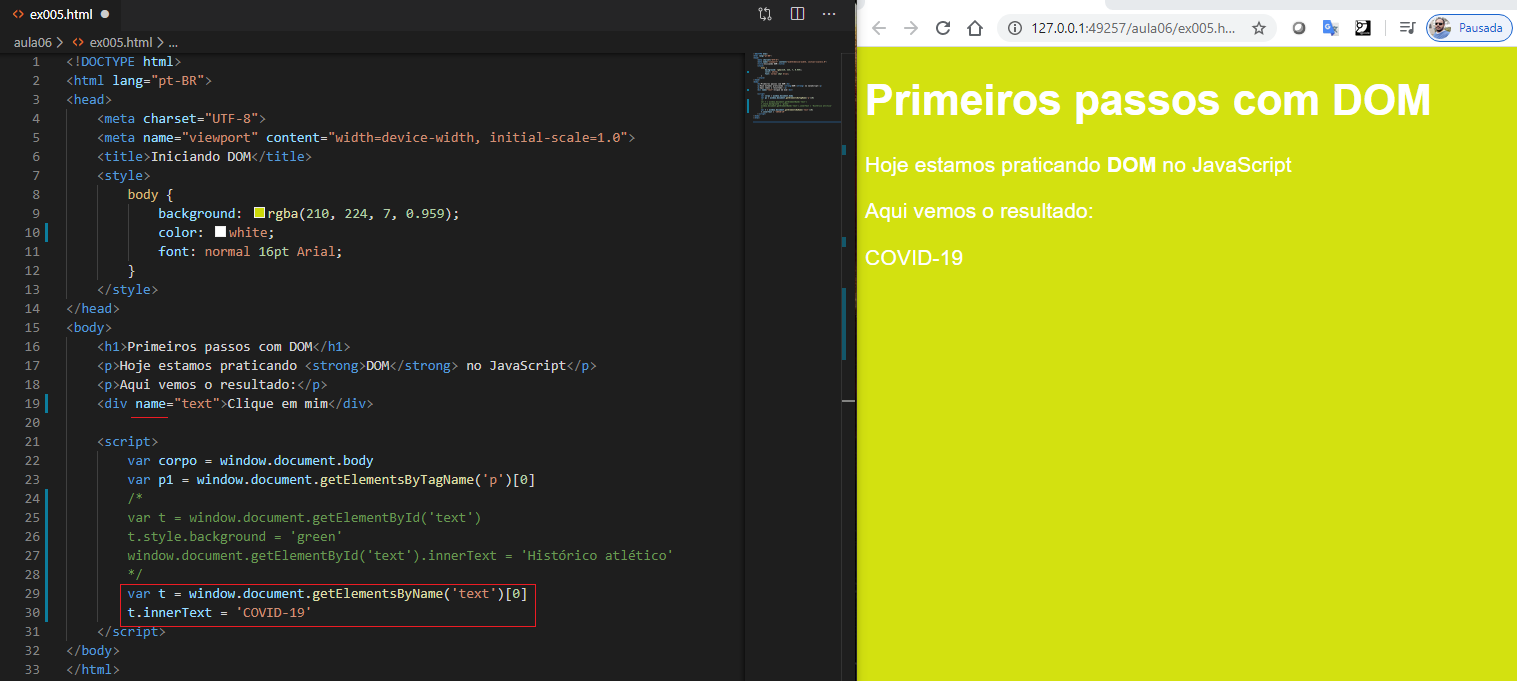
O modo simplificado para modificar o texto



## por Nome

getElementsByName()

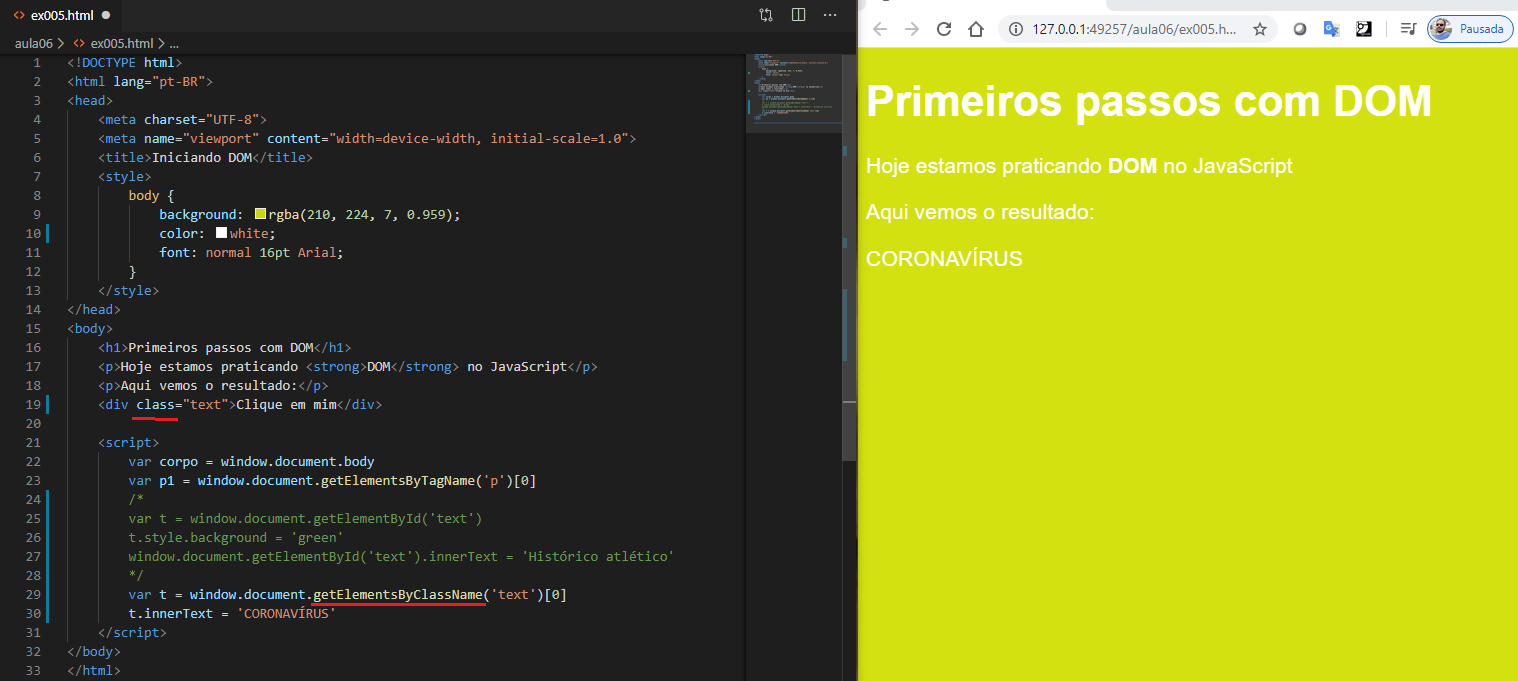
Exemplo:



## por Classe

getElementsByClassName()

Exemplo:



## por Seletor

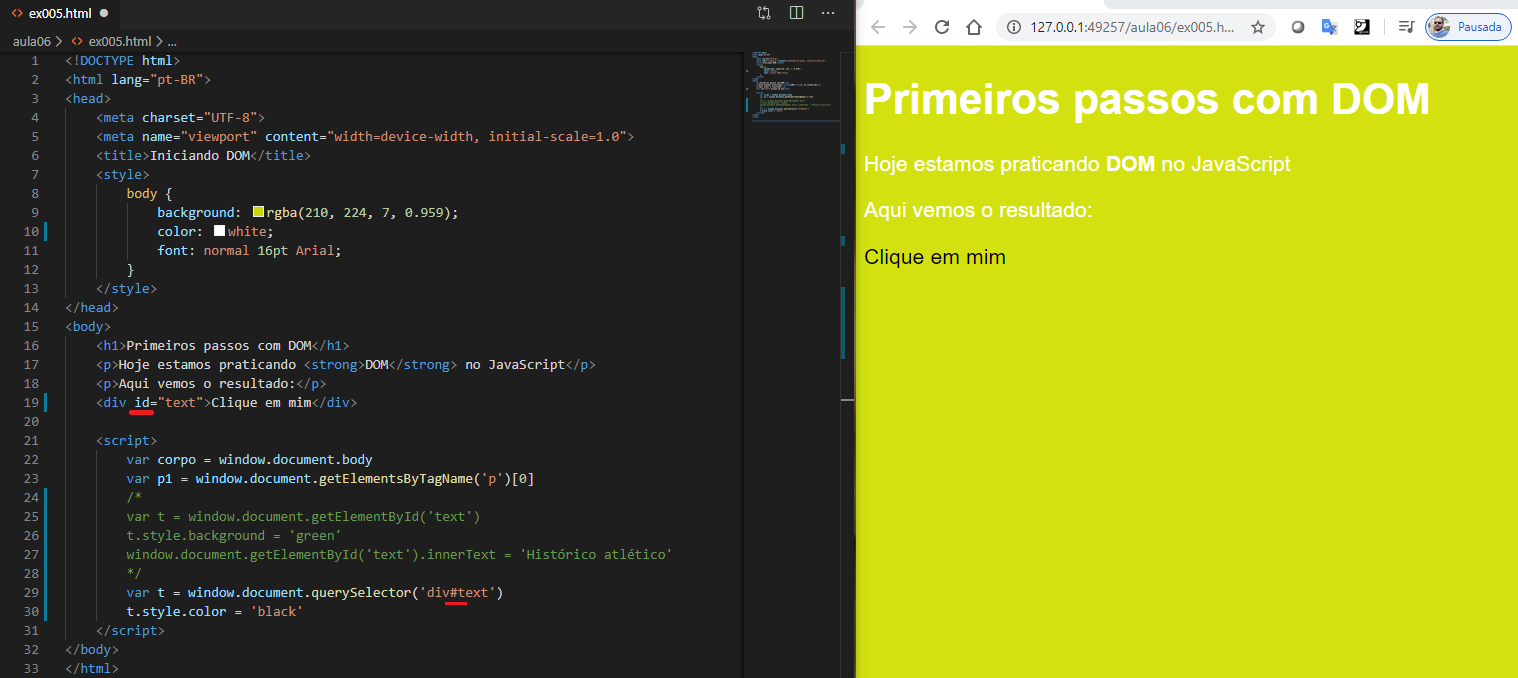
Os seletores estão entre os mais recomendados pela nova documentação.

querySelector()

querySelectorAll()

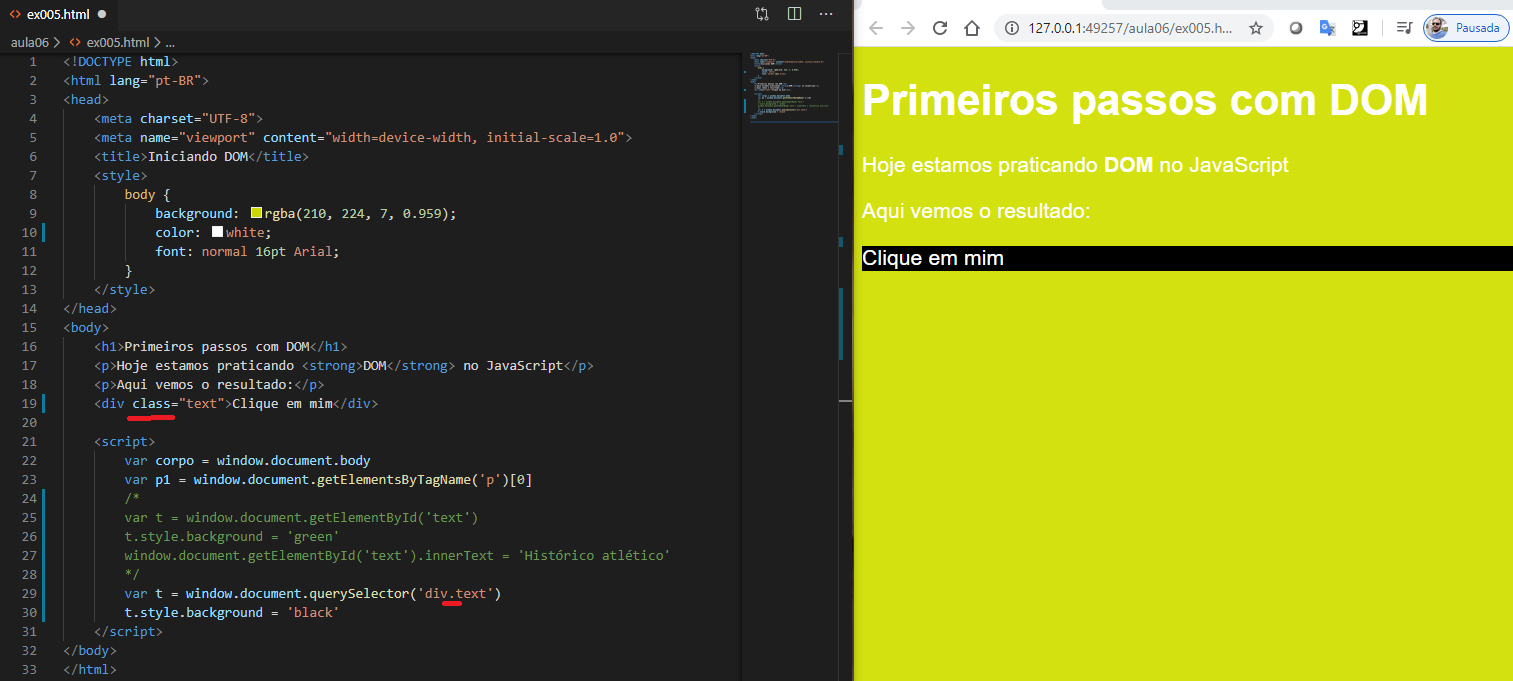
Vamos utilizar para teste o id e a # da mesma forma que no css.

Utilizamos: # id



Vamos utilizar agora a classe o . (ponto) da mesma forma que no css.

Utilizando: . class



# Github

https://github.com/endroni/JavaScript.git

# Conclusão

Uma estruturação em quatro camadas pode ser um tanto complexo no início, porém possibilita um grande reaproveitamento em trabalhos que possuem funções semelhantes.

Espero que tenha valia para seus estudos.

# BIBLIOGRAFIA BÁSICA:

# FLANAGAN, David. JavaScript: o guia definitivo / David Flanagan; tradução: João Eduardo Nóbrega Tortello; revisão técnica: Luciana Nedel. – 6. ed. – Porto Alegre: Bookman, 2013. O Guia Definitivo. ed. O’Reilly.

SILVA, Mauricio Samy. **JavaScript: Guia do Programador:** Guia Completo das Funcionalidades de Linguagem JavaScript. 6ª edição. São Paulo. ed. Novatec 2010.

**BIBLIOGRAFIA COMPLEMENTAR:**

MDN web docs. Mozilla. **O que é JavaScript?** Aprenda o melhor em desenvolvimento web. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\_steps/O\_que\_e\_JavaScript>. Acesso em: 19 de março de 2020.

Documentação Oficial do Ecma. Standards@Internet Speed. Disponível em: <https://www.ecma-international.org/>. Acesso em: 21 de março de 2020.