

LTPW1

Capítulo 03

Primeiros passos com JavaScript

Pronto! Já vai começar a codificação que você está esperando tanto! Chegou a hora de colocar a mão na massa e criar os primeiros scripts em JS. E faremos do jeito tradicional: devagar e sempre. Ao utilizar esse método que eu prefiro aplicar às minhas aulas, veremos exercícios bem simples, com a aplicação de cada tag nova isoladamente. Tem gente que acha melhor começar com um projeto grande, mas em sala eu opto sempre por esse método. Vamos lá?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidades de obter ganho financeiro com ele.



Você já tem o seu editor de código instalado?

Se por acaso você não possui um editor instalado, vamos recomendar o uso do **Visual Studio Code** da Microsoft no seu computador para acompanhar esse curso. Na verdade, nós já discutimos sobre editores no **Curso de HTML5+CSS3**, durante o material da **Aula 04: Primeiros Passos com HTML5**. Recomendo a leitura desse material antes de começar por aqui.



Como já falei anteriormente, você pode usar qualquer editor de código se a sua preferência for o Atom, Sublime, Brackets, Notepad++ ou qualquer outro, mas eu optei pelo uso do **VS code** para preparar esse curso.



APRENDA MAIS: Para acessar todo o material do Curso de HTML5 e CSS3, basta entrar no link a seguir e clicar no link para acessar o material em PDF do curso.

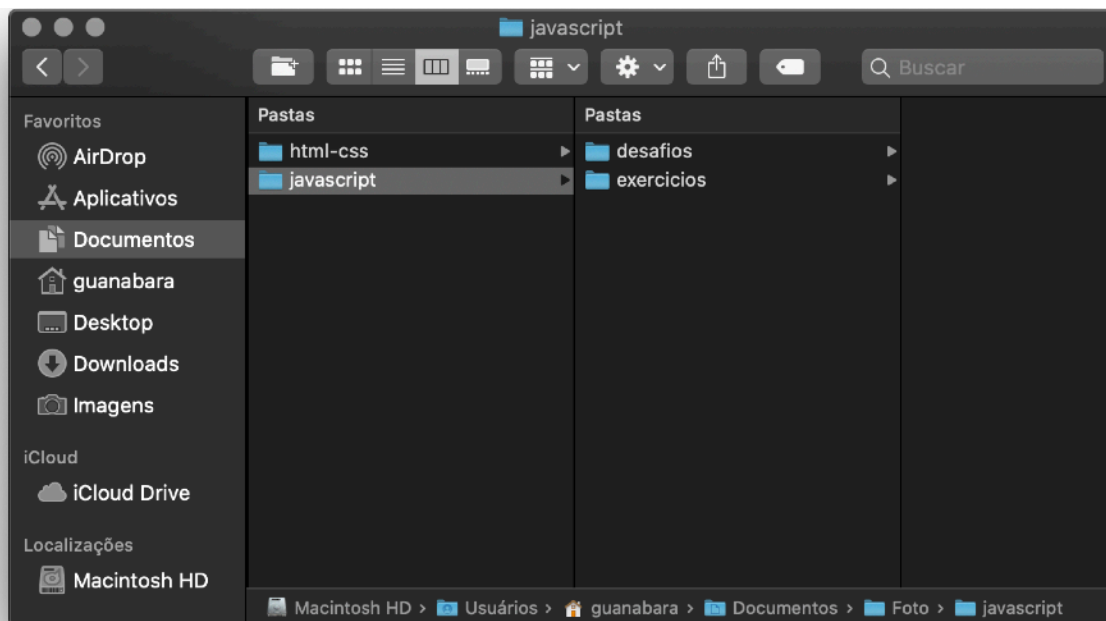
<https://gustavoguanabara.github.io>

Além de instalar o **VS code**, no PDF da **Aula 04 do Curso de HTML** eu te ensinei também a habilitar algumas configurações importantes do editor, como deixar a interface em Português, melhorar o suporte a JavaScript, ligar o Salvamento Automático, quebra automática de linhas e até mudar a interface do modo escuro para o modo claro.

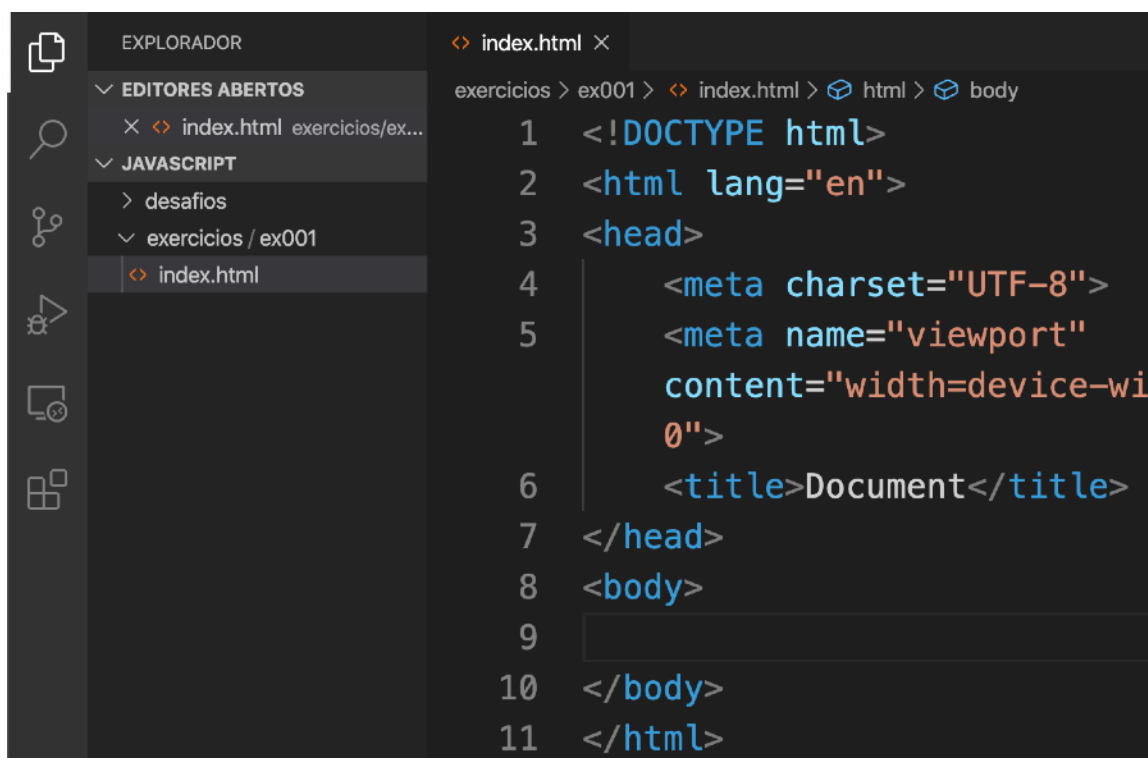
Vamos organizar a nossa pasta de projeto

Durante o **Curso de HTML**, criamos uma pasta dentro dos **Meus Documentos** chamada **html-css**. Para acompanhar esse curso, vamos criar também uma pasta em **Meus Documentos** chamada **javascript** (em minúsculas, sem espaços).

Dentro da pasta **javascript** que você acabou de criar, crie também as pastas **desafios** e **exercicios**, da mesma maneira que você fez com o curso de HTML. A organização das pastas vai ficar exatamente igual ao que estamos apresentando na próxima imagem.



Agora já temos a nossa pasta de projeto **javascript**, vamos abri-la usando o **VS code** da mesma maneira que foi ensinado no outro curso (leia o material em PDF da aula 04 do curso de HTML).



Veja na imagem acima que estou com a pasta **javascript** aberta, e dentro da pasta **exercicios** criei uma pasta chamada **ex001** e dentro dessa pasta, criei o arquivo **index.html**. Dentro desse documento, digitei uma exclamação **!** e pressionei **Enter** para criar o código base em HTML.



IMPORTANTE: Se por acaso você estiver com a pasta **html-css** aberta e quiser fechá-la, basta acessar o menu do **VS code** e acessar a opção **Arquivo > Fechar Pasta**.

Nosso primeiro exercício JavaScript

Agora que você já criou seu primeiro exercício JavaScript, vamos ao código.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
6     content="width=device-width, initial-scale=1.0">
7   <title>Primeiro JS</title>
8 </head>
9 <body>
10   <h1>Primeiro JavaScript</h1>
11   <script>
12     window.alert('Olá, Mundo!')
13   </script>
14 </body>
15 </html>
```

Note que a maneira mais simples de usar JS é incluindo o script em um documento HTML. Sendo assim, se você está fazendo o nosso **Curso de HTML** com a gente (recomendo muito), você já consegue identificar e classificar quase todas as linhas do documento acima. Apenas as **linhas 10, 11 e 12** são uma novidade!

Na recomendação atual, os scripts JS devem ser incluídos no final do corpo do documento HTML, posicionando a tag `<script></script>` logo acima do fechamento da tag `</body>`.

Tudo aquilo que estiver dentro da tag de script será considerado automaticamente JavaScript pelo seu navegador.

Nosso primeiro comando é

```
window.alert('Olá, Mundo!')
```

A palavra `window` (optional) é um objeto que indica uma referência à janela atual do navegador. A palavra `alert()` é um método capaz de disparar uma mini-janela de alerta para avisar algo ao usuário.

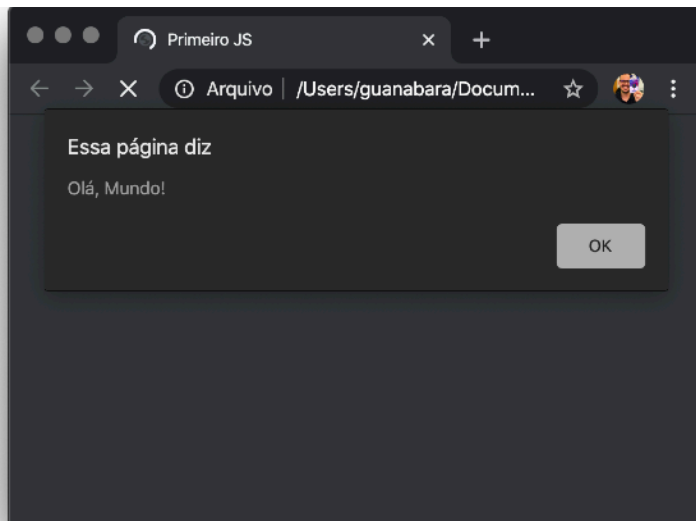
Dentro dos parênteses do método `alert()`, temos uma **string** entre aspas (que podem ser *'aspas simples'*, *"aspas duplas"* ou *`crases`*) para delimitar as palavras que efetivamente aparecerão na janela de alerta.

Para executar nosso script, basta seguir o mesmo procedimento usado para abrir um arquivo HTML: vá até a pasta onde o documento está salvo (provavelmente Documentos/javascript/exercicios/ex001/, se você seguiu os passos), clique com o botão direito do mouse sobre o arquivo index.html e escolha a opção **Abrir com Google Chrome**.

Note que assim que o arquivo abre, a mensagem de alerta já aparecerá automaticamente, antes mesmo de exibir o título <h1> que foi colocado até mesmo antes da tag <script>.

Quando você apertar o botão OK do alerta, a página vai aparecer automaticamente.

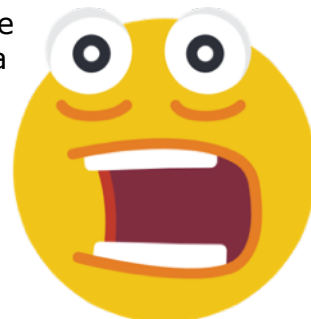
Eu sei que esse é um exercício simples, mas você precisa criar todos os passos na sua casa. É apenas com a prática que você vai poder dizer que aprendeu efetivamente. Sem a prática, nada feito!



Não funcionou? Não se desespere!

A dica que vou te dar agora vale ouro (mas você nem precisa me pagar por isso, relaxa!). Quando estamos começando a desenvolver coisas em JavaScript, é muito comum cometer pequenos erros de digitação que vão fazer nosso código simplesmente parar de funcionar.

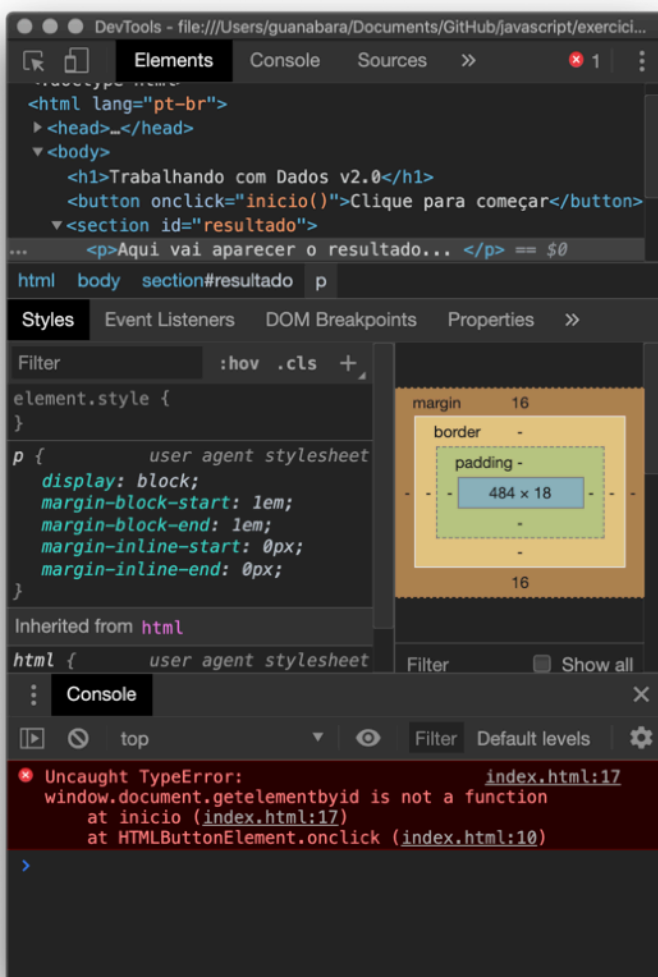
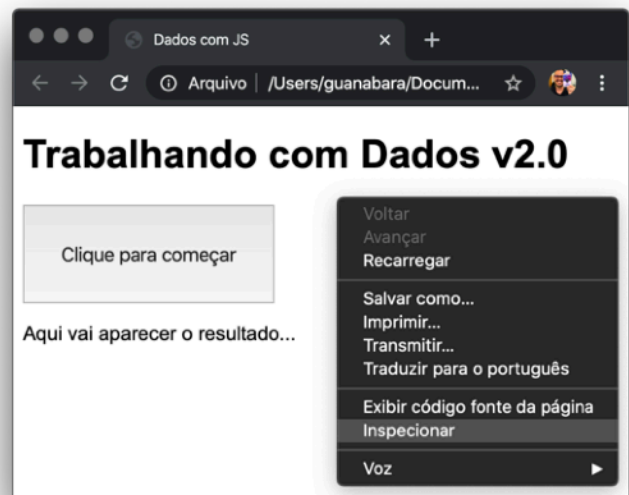
Dependendo do deslize, pode ser que uma única linha pare de funcionar ou até mesmo o script inteiro! Mas tem um jeito de usar o **Google Chrome** para ver o que aconteceu de errado.



Olha bem a **linha 11** do código que eu coloquei na página anterior. As palavras `window.alert` estão todas juntas, todas em minúsculas e não possuem nenhum espaço entre elas. Pois é **EXATAMENTE** assim que você tem que digitá-las! Qualquer pequeno deslize ou falha vai acarretar em um **ERRO** e seu script simplesmente não vai funcionar e a janela de alerta não vai aparecer!

Se algo do tipo acontecer com você, vá até o Chrome, clique com o botão direito do mouse sobre uma área em branco do fundo do seu site e escolha a opção **Inspecionar** (veja a tela ao lado). Outra forma é pressionando Ctrl + Alt + I no teclado ou clicar nos três pontos no canto superior direito da janela e escolher **Mais Ferramentas > Ferramentas do Desenvolvedor**.

Na janela que vai abrir, foque sua atenção no canto superior direito e veja o ícone vermelho que vai aparecer lá em cima.



Este ícone aqui ao lado só vai aparecer se você clicar no botão e tentar executar o código com erro no navegador.

Agora vamos para a parte de baixo da janela, na guia **Console**. Note que existe um erro em Inglês indicando qual foi o erro que você cometeu!

O erro também indica a linha aproximada onde ele foi encontrado que, mas pode se acalmar.

Ao notar que você cometeu esse erro, corrija-o, salve as alterações (mantenha sempre o Auto Save ligado) e volte a atualizar o navegador (geralmente apertando F5 ou Shift+F5).

Toda vez que você esperar uma ação do seu script e ela não acontecer, dê sempre uma olhada nessa janela de **DevTools**. Ela é uma das suas melhores amigas a partir de hoje!

IMPORTANTE! Essa janela **DevTools** não vai te mostrar falhas no seu código HTML, apenas no seu código JavaScript, ok? Falhas ou deslizes em HTML simplesmente são ignorados pelo seu navegador. Sendo assim, se algum componente nas suas tags não estiver se comportando como você espera, simplesmente vá lá no seu código e verifique se sua tag está digitada corretamente. Não conte com o **DevTools** pra isso, infelizmente.

Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo vai diretamente para a playlist completa do curso que já está totalmente disponível e mostra todos os procedimentos passo-a-passo. Acesse agora mesmo!



Curso em Vídeo: https://www.youtube.com/playlist?list=PLHz_AreHm4dlsK3Nr9GVvXCbpQyHQ11o1

Identificadores

- Podem começar com **letra, \$ ou _**
- Não podem começar com **números**
- É possível usar **letras** ou **números**
- É possível usar **acentos** e **símbolos**
- Não podem conter **espaços**
- Não podem ser **palavras reservadas**

Data Types

number

Infinity

NaN

string

boolean

null

undefined

object

Array

function