LTPW1

Capítulo 02

Como o JS funciona?

Agora que você já sabe de onde veio o JavaScript, chegou a hora de entender como a tecnologia funciona. É importante entender os mecanismos de uma linguagem antes de começar a estudá-la. Nesse material, falaremos de JS puro, com os objetivos primordiais definidos para a tecnologia. Vamos entender o passo-a-passo que um código JS leva para ser executado em nosso navegador.

Antes de começar...

... é importante deixar claro que esse capítulo do material vai explicar o funcionamento do JavaScript da maneira tradicional da sua aplicação.

JS é uma linguagem que hoje funciona em servidores? Claro, mas não foi feita pra isso! Para rodar em servidores, precisamos de um software criado por terceiros (que é o Node.JS).

JS pode ser usada para criar aplicativos para Windows? Claro, mas não foi feita pra isso! Usando uma biblioteca chamada Electron, dá pra criar uns programas bem legais, tipo o VS Code da Microsoft, que é feito em JavaScript.

JS pode ser usada para criar programas para celular? Claro, mas não foi feita pra isso! Se você aprender um pouco sobre frameworks como PhoneGap, Xamarin, Ionic ou React, vai conseguir criar aplicações portáteis programando em JavaScript.

Sendo assim, não falaremos sobre os "super poderes" que a linguagem JavaScript vem ganhando com o tempo. Estamos abordando aqui a maneira primordial para qual a linguagem foi criada: interações na web, usando seu navegador.

Combinado?

Lembra de como funciona a Internet?

Lá no material PDF do **Curso de HTML5 e CSS3**, que está nesse mesmo repositório, eu expliquei como funcionava a Internet. Se por acaso você ainda não leu, está ferindo uma regra básica do nosso acordo aqui: não pular nenhum passo. No primeiro capítulo desse material eu dei os pré-requisitos desse curso aqui, e ler o material introdutório de HTML era uma exigência!

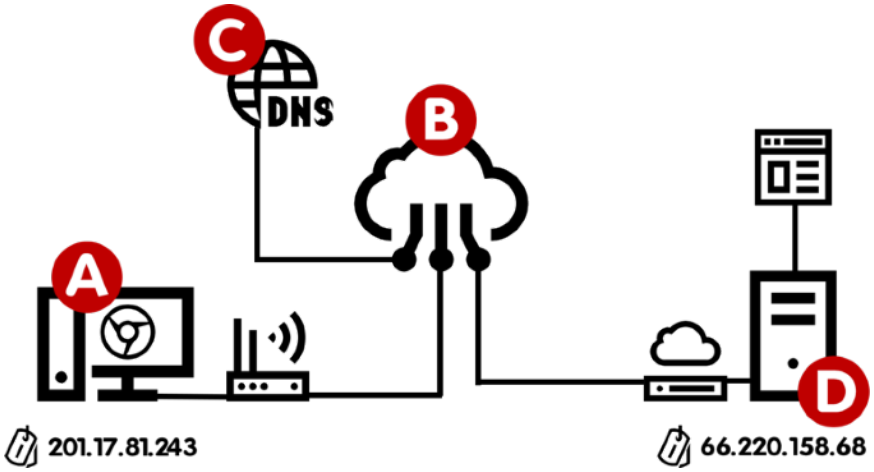
APRENDA MAIS: Para acessar todo o material do Curso de HTML5 e CSS3, basta entrar no link a seguir e clicar no link para acessar o material em PDF do curso.

https://gustavoguanabara.github.io

Agora que você já levou um leve puxão de orelha, vamos continuar.

A Internet funciona baseada no paradigma **cliente-servidor**. Nesse exato momento você é o **cliente**, que está com seu navegador aberto e acessando alguns sites (inclusive o repositório do GitHub, no link acima). Essa página que está aparecendo no seu computador/celular está vindo de um **servidor**, que é um outro computador que está preparado para prover os arquivos.

Vamos voltar àquela imagem que está no material de HTML:



Quando está conectado à Internet (**B**), um visitante abre um navegador (**A**) e digita uma URL (*endereço do site*), o servidor web (**D**) deve ser acessado e precisa consultar um servidor DNS (**D**) para resolver o nome do domínio, retornando o IP atual do servidor.

Uma vez que a conexão entre o cliente (**A**) e o servidor (**D**) está estabelecida, o `arquivo` solicitado vai ser transferido para a máquina de destino.

IMPORTANTE: Quando não solicitamos um arquivo específico ao servidor, ele vai entregar o **arquivo de índice** do site. Para sites em HTML5, esse arquivo de índice se chama `index.html`.

Quando o arquivo HTML inicial chega ao cliente, o navegador vai analisar todas as tags e vai gerar o resultado visual da página. Quando ele encontra tags de mídia (`<img>`, `<video>`, `<audio>`, etc), novas solicitações são enviadas para o servidor (**D**), pedindo mais arquivos (`jpg`, `png`, `mp3`, `mp4`, etc).

Quando os comandos que chegam estão em **JavaScript**, a *Engine* do navegador será a responsável por executá-los (no caso do **Google Chrome**, o motor JS se chama **V8**). Ele vai conseguir criar interatividade do visitante com os componentes do site (*DOM*) e executar rotinas definidas pelos programadores do site.

HTML e CSS não é programação. JavaScript é?

Quando estamos criando código **HTML** e **CSS**, podemos dizer que estamos *"desenvolvendo em HTML"*, mas é errado dizer que estamos *"programando em HTML"*. HTML e CSS são linguagens, mas não são **linguagens de programação**. Essas linguagens são mais classificadas como **linguagens de marcações**.

Já a **JavaScript** é uma linguagem de programação, considerada uma *linguagem de scripts não compilados*. Todos os códigos JS são enviados para o navegador do cliente e são interpretados linha-a-linha, gerando o resultado.

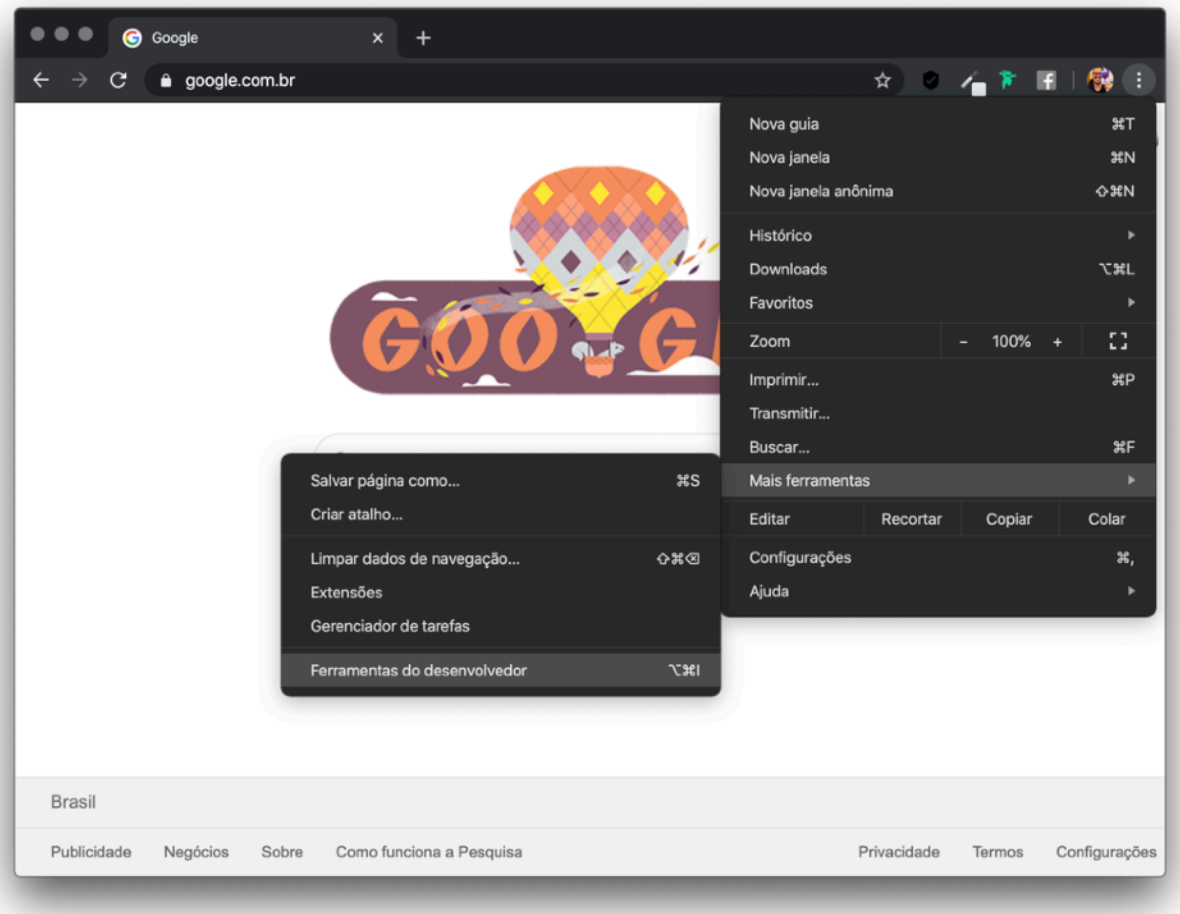
Sendo assim, podemos dizer *"estou programando em JS"*, mas não podemos dizer *"estou programando em HTML"*.

Sacou? 😉

Me mostra um exemplo desse tipo de interatividade com JS?

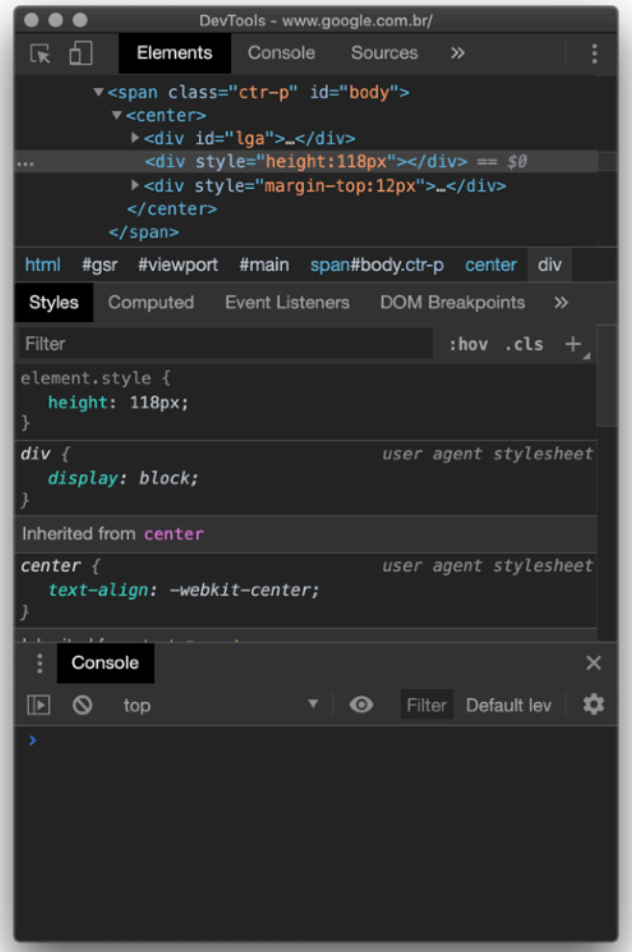
Mas é claro! Vamos fazer um exemplo que se tornou clássico nas minhas aulas.

Abra aí o seu **Google Chrome** (tem que ser ele, pois tem ferramentas que ajudam programadores como nós) e acesse o site `www.google.com.br`



Depois clique naqueles três pontinhos que existem lá em cima na direita (veja a imagem), escolha o menu `Mais ferramentas` e depois `Ferramentas do desenvolvedor`.

A tela que vai abrir se chama **DevTools** e está representada aí ao lado. Ela pode aparecer colada ao seu navegador ou flutuando sobre a tela. Você muda isso clicando sobre os três pontinhos lá em cima na direita e mudando o `Dock side`. Mude essa opção para Undock e deixe ela sempre `flutuando`, `vai ajudar bastante na` visualização das coisas na maioria dos casos.



Essa ferramenta é muito útil para quem desenvolve sites. Como você pode perceber, ele mostra os elementos HTML, CSS e permite identificar detalhes do código.

É com o DevTools que você também vai ser capaz de detectar erros nos seus códigos que você vai começar a criar agora. Ele com certeza será um dos seus melhores amigos na caminhada por aprender a programar em JavaScript.

Mas o que mais importa no momento é a parte de baixo da tela, chamada **Console**. Ele permite que possamos interagir com componentes do site aberto no momento.

Deixe o site do Google aberto, vá até a janela do `DevTools`, procure o `Console` e digite o código a seguir:

```
window.alert('Olá, Mundo!')
```

Depois aperte Enter e veja a magia acontecendo. Você conseguiu fazer o site do Google falar contigo! 😱

Achei meio bobo 😒. Tem como melhorar?

Você nunca está satisfeito(a) mesmo, não é? Pois agora você vai ser capaz de mudar o site do Google como nunca imaginou antes. Clique no botão Ok do alerta que apareceu para fechá-lo e volte para a sua janela de console. Dessa vez, o comando vai ser um pouco maior, mas seu efeito vai ser mais legal! Confia em mim!

```
document.body.style.backgroundColor = 'black'
```

Eu não preciso te ensinar JS agora, mas só de olhar o comando acima e você vai perceber que praticamente vai conseguir prever o que vai acontecer. Basta ter um conhecimento básico de Inglês. E se você não conseguiu entende, não se desespere! Isso vai melhorar com o tempo.

IMPORTANTE: Em JavaScript, as letras maiúsculas e minúsculas fazem toda a diferença! Por exemplo, na linha acima, perceba que existe uma única letra `C` em maiúsculas. Se você esquecer de colocar assim, vai dar erro! Tecnicamente, dizemos que JS é uma linguagem **case sensitive**. Já vai aprendendo os termos aí.

Você viu isso? O site do Google ficou preto! 🤯

Mas não sai por aí dizendo que você hackeou o site do Google! Na verdade, você só alterou a configuração do código que foi enviado para o seu computador. Qualquer outra pessoa que esteja acessando o Google da sua casa, vai ver o site normal, com o fundo branco.

Por hoje é só, pessoal!

Paramos por aqui por enquanto, mas você não pode parar de aprender, não pode parar de praticar. Tire um tempo pra dar uma aprofundada nos seus conhecimentos antes de prosseguir, e pra isso eu já tenho uma indicação pra te fazer.

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 um vídeo que mostra o conteúdo que você acabou de ver aqui, mas de uma maneira mais aprofundada e com outro ponto de vista. Acesse agora mesmo!

Curso em Vídeo: https://youtu.be/Ptbk2af68e8