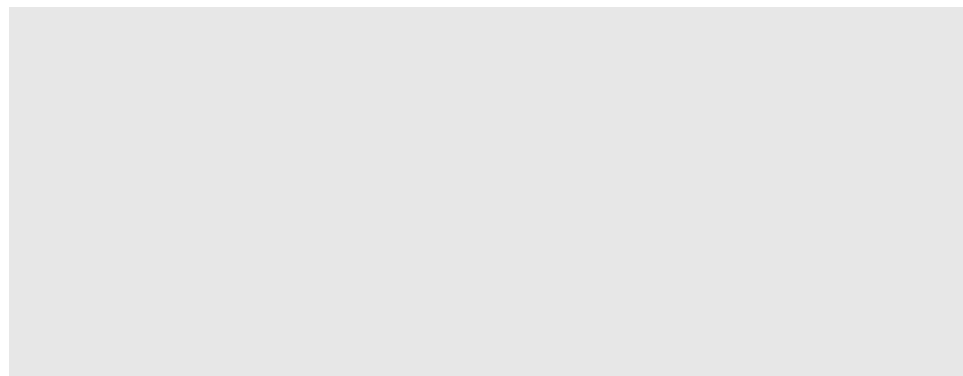


LTPW1

Capítulo 05

Interações básicas com componentes da página

No capítulo anterior, aprendemos uma maneira bem simples de interagir com o usuário usando os métodos `prompt()` e `alert()`. Agora chegou a hora de aprender com partes da sua página, utilizando alguns componentes do seu documento HTML5. Essa técnica que eu vou te ensinar agora se utiliza de um conceito maior chamado DOM, mas por enquanto não vamos entrar nesse assunto. Só confia no tio e vem!



Já vamos direto para um código de exemplo

Vou pular a parte de papo e conceitos e vou direto para a mão na massa nesse caso. Lá na sua pasta de projetos JavaScript, dentro de **exercícios**, crie uma pasta **ex004** e dentro dela um arquivo `index.html` com aquele código base que já fizemos algumas vezes. Modifique-o e adapte-o para que fique como apresentado a seguir:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
6          initial-scale=1.0">
7      <title>Dados com JS</title>
8  </head>
9  <body>
10     <h1>Trabalhando com Dados v2.0</h1>
11     <button onclick="inicio()">Clique para começar</button>
12     <section id="resultado">
13         <p>Aqui vai aparecer o resultado... </p>
14     </section>
15 </body>
16 </html>
```

O nosso foco inicial será relembrar o código que escrevi lá na **linha 10**, onde criei um botão com uma chamada ao método `inicio()` quando for clicado pelo visitante. Isso já foi feito no capítulo anterior.

A novidade aqui está nas **linhas de 11 a 13**, onde criei uma seção identificada como `resultado`, com um parágrafo interno. Ele será o nosso elemento mais importante desse exercício.

Identificar os elementos que desejamos acessar via script é essencial para evitar dores de cabeça futuros. Quando identificamos o `<section>` com o nome `resultado`, deixamos as “portas abertas” para o JavaScript entrar.

Agora, para resolver esse exercício, vamos criar o método de resposta ao evento `onclick` do botão. Monte uma tag `<script></script>` na **linha 14**, logo depois do `</section>` e escreva o código a seguir. Não se preocupe se a confusão tomar conta da sua cabeça, logo logo eu vou explicar o script linha a linha.

```

14     <script>
15         function inicio() {
16             let nome = window.prompt('Qual é o seu nome?')
17             let res = window.document.getElementById('resultado')
18             res.innerHTML = `<p>Olá, <strong>${nome}</strong>! É
                               um grande prazer te conhecer! &#x1F596;`
19         }
20     </script>

```

Dentro do script, entre as **linhas 15 e 19**, criamos a função `inicio()` e um bloco com `{}`. Dentro declaramos duas variáveis: `nome` e `res`. A variável `nome` vai receber o resultado de um `prompt()`. Até aí, nenhuma novidade.

IMPORTANTE: A qualquer momento, se você sentir alguma dificuldade nas linhas em que não explico com tantos detalhes, como a **linha 16** do código acima, talvez seja sinal de que você tenha pulado algum material anterior. A explicação mais aprofundada do uso do `prompt()` está no capítulo anterior.

Sendo assim, talvez seja melhor dar uns passos pra trás de vez em quando. Isso é super normal! Tentar aprender as coisas com pressa, geralmente causam resultados assim. E não se esqueça de praticar sempre! Só ler esse material, não adianta muita coisa. Foco total, mas sem perder a paciência! Você consegue!

A grande novidade aqui foi a **linha 17**. É nela onde vamos focar nossos esforços agora. Iniciando a linha, usamos a instrução `let` que vai criar uma variável chamada `res` (na verdade, o que é criado é um **objeto**. Mas por enquanto pense em um objeto como uma *variável especial*).

O objeto `res` será o nosso **elo** entre o JavaScript e o HTML. Lembra da **linha 11** do código HTML? Lá eu dei um id para o elemento `section` e agora vou tirar proveito disso. Só segue a linha 17 e tenta ler em Português:

"Na janela do navegador (window), dentro do documento atual (document), pegue o elemento que tenha o id (getElementById) com o nome 'resultado'."

É exatamente isso que você leu! A partir de agora, tudo aquilo que eu fizer com o objeto `res` em JavaScript, vai se refletir o objeto que tenha o id citado no código. Nesse código, se mexermos com `res`, vamos afetar diretamente a `<section>` do nosso documento HTML. Sacou?

Agora vamos olhar com atenção para a **linha 18** do código acima. Ela vai modificar o "HTML que está dentro" (`innerHTML`) da `section`. Dá só uma olhada na **linha 12** do nosso código HTML. Esse é o parágrafo que vai aparecer no navegador quando abirmos o `index.html`. Porém, a **linha 18** vai mudar esse parágrafo quando o botão da **linha 10** for clicado.

Para facilitar a programação, usei uma *template string* (já estudamos ela nos capítulos anteriores) e incluí um *placeholder* `${nome}` para personalizar a mensagem. No fim da linha, ainda coloquei um código do emoji 🙌 para a mensagem ficar mais simpática.

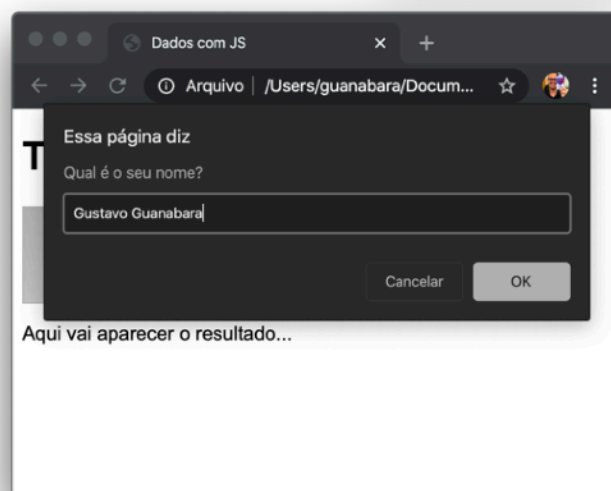
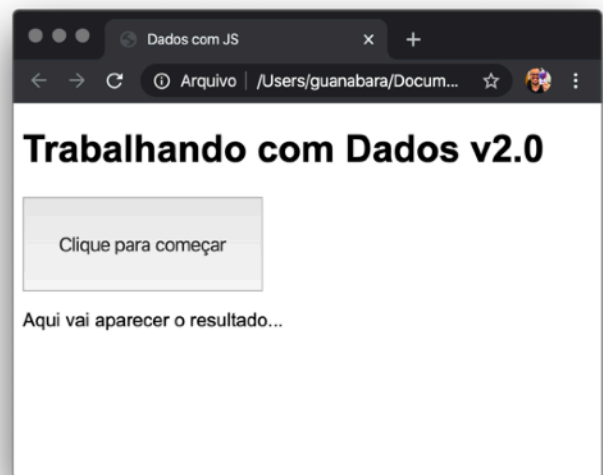
EMOJI? Se você não entendeu como o código `🖖` virou uma mãozinha, recomendo que abra nosso material de HTML e vá para o PDF da **Aula 05**. Lá eu explico como descobrir o código de qualquer emoji existente.

Agora vamos fazer tudo funcionar

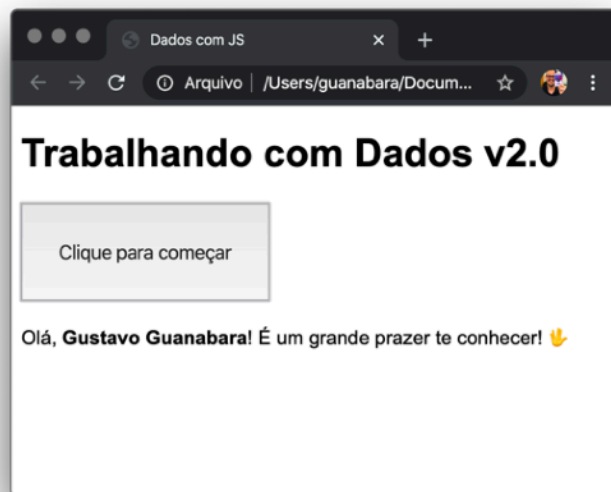
Vamos abrir o nosso arquivo `index.html` da pasta **ex004** e ver o funcionamento do script.

Logo no início, o parágrafo da seção que está logo abaixo do botão vai apresentar o texto padrão que configuramos no documento HTML. Talvez na sua casa, o botão fique um pouco menor, pois fiz umas configurações adicionais em CSS.

Ao clicar no botão, um `prompt()` será disparado, pedindo que o visitante digite o seu nome.



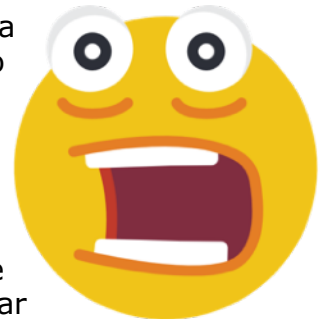
Por fim, depois de apertar o botão OK que fica na janela do `prompt()`, o parágrafo vai ser substituído pela mensagem de boas-vindas.



SUGESTÃO: Se o seu programa funcionou corretamente, meus parabéns! Só sugiro que faça uma pequena alteração na **linha 18** e veja a diferença. Substitua o símbolo de = por += (sem espaços) e perceba a diferença! Conseguiu entender? Mais pra frente, falamos melhor sobre isso.

Não funcionou? Não se desespere!

Eu sei que já falamos sobre isso no capítulo 03, mas como dessa vez tivemos que escrever um código bem maior, acho importante repetir o conselho. 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.

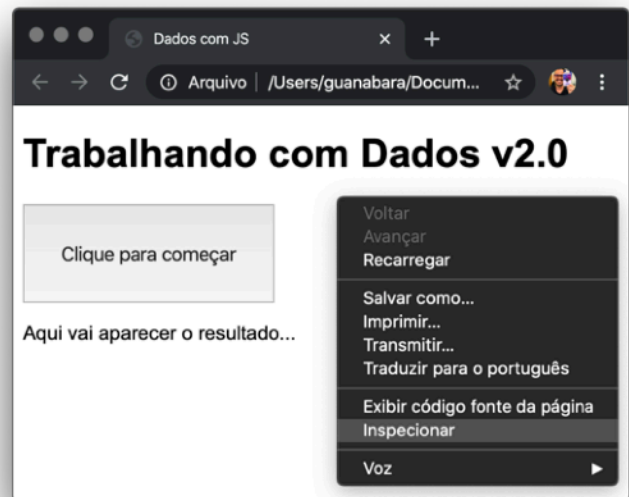
Vamos ver um erro muito comum que os iniciantes acabam cometendo. Analise a linha a seguir.

```
17 | let res = window.document.getelementbyid('resultado')
```

Mudei a **linha 17** do código, escrevendo `getElementById()` completamente em letras minúsculas (o que é um erro). Ao clicar no botão, o script pede o nome do visitante, mas não é capaz de mostrar o resultado na tela.

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.



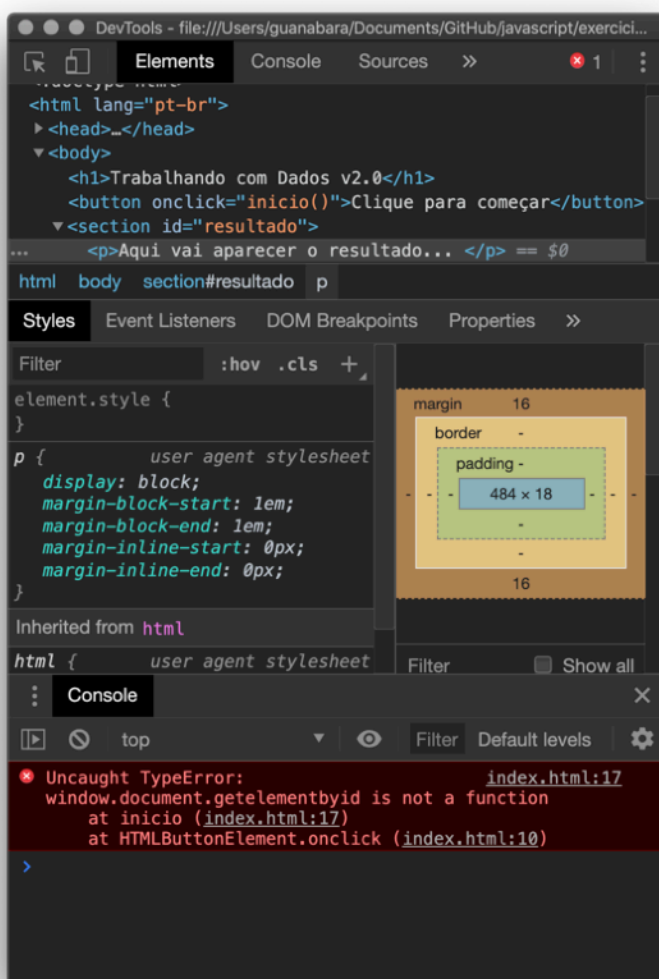
Este ícone aqui ao lado só vai aparecer se você clicar no botão e tentar executar o código com erro no navegador.

Agora vamos para a parte de baixo da janela, na guia **Console**. Note que existe um erro dizendo (em Inglês) que *"window.document.getelementbyid não é uma função em inicio()"*. Inclusive ele me indicou que o erro apareceu na linha 17 do arquivo index.html.

O erro também indica que algo aconteceu na linha 10, mas pode se acalmar. Ele apenas está dizendo que tudo se originou quando você clicou no botão (que está na linha 10).

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).

Não se esqueça: a janela **DevTools** do Google Chrome é uma das suas melhores amigas! #BFF



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!