Curso Informática para Internet

LTPW1

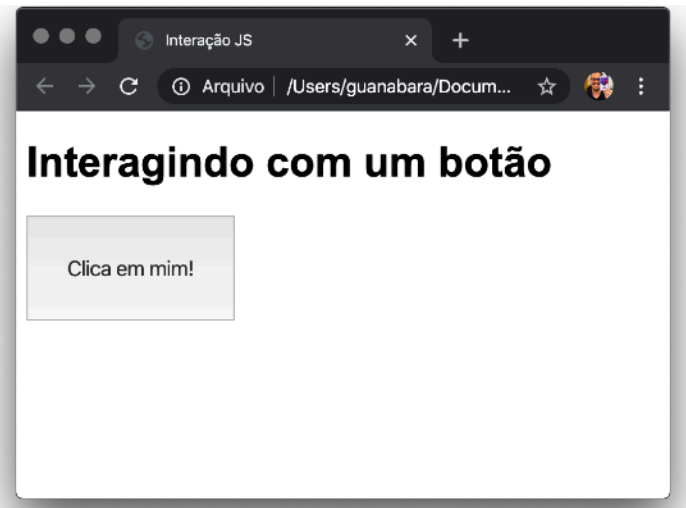
Capítulo 04

Interações básicas com o usuário e com o navegador

A sua caminhada para aprender JavaScript já começou no capítulo anterior. Agora chegou a hora de criar uma interatividade com o usuário e também com os componentes exibidos no navegador. Vamos começar fazendo um botão reagir às nossas ações com ele e também vamos aprender a pedir dados para o usuário. Vamos nessa, que o trabalho não pode parar!

Interagindo com um botão

A maneira mais simples que eu posso imaginar para que possamos interagir com uma página web é com um botão. Nosso objetivo aqui é criar uma página simples, com um botão sensível ao clique e que vai reagir de acordo com essa ação. Dá só uma olhada na imagem a seguir.



Vamos ver como isso foi criado e como ele reage ao clique. Crie uma pasta **ex002** dentro dos seus **exercícios**, adicione um arquivo **index.html** a essa pasta e digite o código a seguir. Em seguida, vamos analisar as linhas mais importantes.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport"
       content="width=device-width, initial-scale=1.0">
6      <title>Interação JS</title>
7      <style>
8          body { font: 12pt Arial; }
9          button { font-size: 12pt; padding: 30px; }
10     </style>
11 </head>
12 <body>
13     <h1>Interagindo com um botão</h1>
14     <button onclick="clicou()">Clica em mim!</button>
15     <script>
16         function clicou() {
17             window.alert('Você clicou no botão!')
18         }
19     </script>
20 </body>
21 </html>
```

Vamos olhar com atenção a **linha 14**, onde criamos o botão. Existe um parâmetro na tag `<button>` que é o `onclick`. Note que dentro das aspas eu coloquei o nome de um **evento** que será tratado pelo JavaScript. A ligação entre o HTML e o JS será pelo nome desse evento (não esqueça dos parênteses no final, mais tarde você entenderá pra que eles servem).

Agora analise a **linha 16**. Ela é exatamente a definição da função que tratará do evento. Ao contrário do que fizemos no capítulo anterior (se esqueceu, retorna lá no PDF e confere), o `window.alert()` não está sozinho dentro do `<script>`. Ele está justamente dentro da função (identificada pela palavra `function` do JS).

Toda função em JS é relacionada a um **bloco**, que nada mais é do que um conjunto de comandos que estão entre chaves **{ }**. Tudo o que estiver entre chaves em JS, chamaremos de **bloco**.

Sendo assim, o comando que está na **linha 17** não vai executar assim que a página for carregada. No lugar disso, ela será executada só quando a função será disparada, e o modo de disparo nesse caso será feito na **linha 14**, pelo evento `onclick` do botão.

CÓDIGO NA MÃO: Todos os códigos dos exercícios já estão disponíveis no meu repositório público. Mas isso não significa que você deve copiar os códigos para "aprender mais rápido". O objetivo aqui é ter o código dos exercícios sempre à mão em casos de emergência ou consulta rápida. Dá uma olhada também na pasta de **desafios**, pois lá você vai provar para si mesmo(a) que realmente aprendeu. Para acessar tudo isso, é só ir direto para o endereço a seguir:

Interagindo com o usuário

Outra maneira de usar o JavaScript básico para criar interações é através do método `prompt()`. Com ele, podemos pedir para o usuário digitar dados e usar isso para causar uma resposta personalizada para ele.

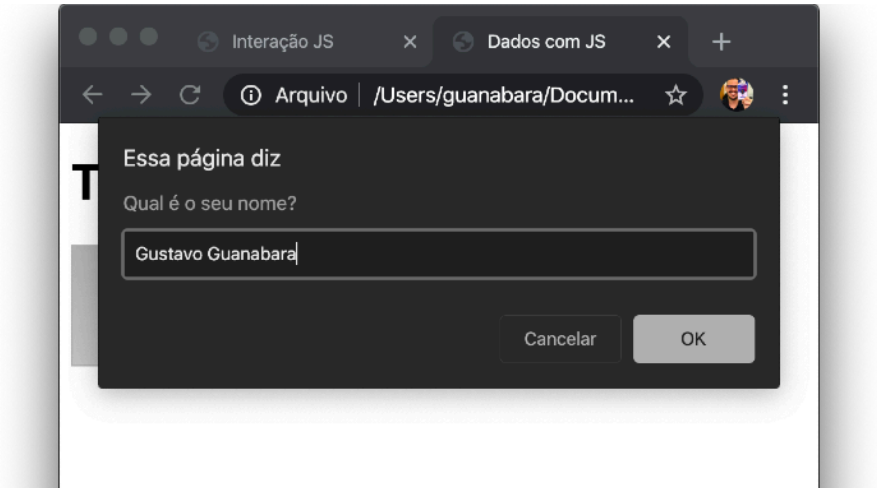
Vamos começar criando uma pasta chamada **ex003** dentro da sua pasta de **exercícios** e criar um arquivo **index.html** dentro dela.

O código que virá a seguir é muito parecido com o anterior, na parte de HTML e CSS. Inclusive o botão, que na **linha 14** vai disparar o evento de clique, chamando a função `inicio()`.

A grande novidade desse exercício está nas **linhas 17 e 18**, e é com elas que vamos nos preocupar especialmente.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
       initial-scale=1.0">
6      <title>Dados com JS</title>
7      <style>
8          body { font: 12pt Arial; }
9          button { font-size: 12pt; padding: 30px; }
10     </style>
11 </head>
12 <body>
13     <h1>Trabalhando com dados v1.0</h1>
14     <button onclick="inicio()">Clique para começar</button>
15     <script>
16         function inicio() {
17             let nome = window.prompt('Qual é o seu nome? ')
18             window.alert(`Olá, ${nome}! É um prazer te
               conhecer!`)
19         }
20     </script>
21 </body>
22 </html>
```

A **linha 17** tem o método `window.prompt()`, que gera uma solicitação para que o usuário digite o seu nome. Essa janela se parece bastante com um alerta, mas tem a diferença de incluir uma caixa disponível para aceitar a digitação. Veja o resultado de um `prompt()` na imagem abaixo.



Mas como vamos guardar o nome do visitante? Aí entra o início da **linha 17**. A instrução `let nome` serve para declarar uma **variável** chamada `nome`, que vai guardar o nome que a pessoa vai digitar.

Em JavaScript, o símbolo de = não se lê como "igual". Na verdade, sempre que você encontrar um =, leia como "recebe".

Lendo então a **linha 17** depois de aprender tudo isso, ficamos com:

```
let nome = window.prompt('Qual é seu nome?')
```

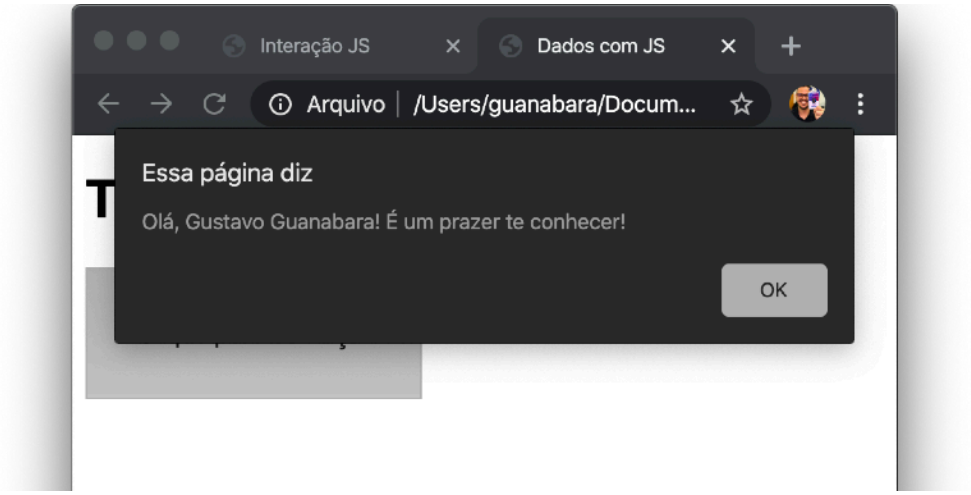
*"A **variável nome** vai **receber** o resultado de um **prompt** que vai aparecer na janela perguntando qual é o nome do usuário"*.

> **MAIS SOBRE VARIÁVEIS:** Em JavaScript declaramos variáveis com as palavras `let` ou `var`. Basicamente, usando var nós podemos ter problemas com o escopo da variável (falaremos mais sobre isso na aula 07, mas por enquanto vamos só com `let` mesmo).
>
> Outra coisa que podemos criar são variáveis imutáveis ou constantes. São variáveis que ficam na memória até o fim da execução do script, mas que não podem ter seu valor alterado de forma alguma. A declaração de uma constante é feita colocando a palavra `const` no lugar da palavra `let`.

Já na **linha 18**, temos também uma novidade. Note que dentro do `alert()`, usamos crases para delimitar a **string** dessa vez.

```
window.alert(`Olá, ${nome}! É um prazer te conhecer!`)
```

Uma **string** que está entre crases tem um nome especial: se chama **template string**. Esse tipo de string é uma das novidades do ECMAScript moderno e quebram um galho gigante!



Dentro de uma **template string** podemos usar **placeholders** no seu interior. Um placeholder é representado pelos símbolos `${}` e podem ser usados para facilitar a exibição de conteúdos de variáveis ou expressões.

Olhando a **linha 18**, perceba que `${nome}` vai ser substituído pelo conteúdo da variável nome, criada na **linha 17** e que está guardando o nome da pessoa que está rodando o script.

Experimente fazer o **ex003** no seu computador e veja o resultado! Se precisar olhar o código original, não se esqueça de visitar o repositório público. Está tudo lá!

E começam os desafios!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d001 e d002**. Acesse o repositório público, abra a área do curso de JavaScript e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!

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_AreHm4dlsK3Nr9GVvXCbpQyHQl1o1