

AULA 5 – CRIANDO SOLUÇÕES WEB

OBJETIVO DA AULA

Praticar os conceitos vistos até o momento.

APRESENTAÇÃO

Nesta aula colocaremos a mão na massa. Isso mesmo! Vamos fazer exemplos práticos juntos para testar os nossos conhecimentos em relação aos conceitos aprendidos nas aulas anteriores. Preparados? Vamos lá! O primeiro exemplo que faremos é o mesmo que usamos na aula de algoritmos, a ideia é mostrar o resultado da soma de dois números.

EXEMPLOS PRÁTICOS

EXEMPLO 1

Para criar o nosso primeiro exemplo: pedir dois números para o usuário e apresentar o resultado da soma destes números na tela, precisaremos de um documento de HTML e outro documento de JavaScript. Neste exemplo, vamos fazer com dois arquivos separados, mas você poderia fazer tudo no documento de HTML, porém esta não é uma boa prática.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Meu Primeiro Exemplo</title>
6  </head>
7  <body>
8      <input type="text" id="n1"> <br>
9      <input type="text" id="n2"> <br>
10     <p id="res">Resultado: </p>
11     <button onclick="Soma()"> Soma </button>
12
13     <script src="teste.js"></script>
14 </body>
15 </html>
```

Link para ter acesso ao código: <https://github.com/GRANCodigo/PraticaDeProgramacao/blob/3be66e0788cdebdea8ac45dbaadd0b9e8f060c2/Livro%20Eletr%C3%B4nico/Unidade1Aula5a.>

No documento de HTML criamos a estrutura básica que aprendemos na aula 1. Dentro do `<body>`, nas *linhas 8 e 9* inserimos duas caixas de texto para que o usuário consiga digitar os 2 números que serão somados. Para isso, utilizamos a tag `<input>` com o atributo **type="text"**, essa combinação projeta na tela um campo de texto. O atributo *type* pode receber outros valores como: *password*, *button*, *radio* e entre outros.



LINK

Aprenda mais sobre os campos de formulário em: https://www.w3schools.com/html/html_forms.asp. Acesso em: 05/01/2023



Repare que em cada um dos inputs criamos um identificador (**id**) que deve ser único por todo o documento, é como se fosse o nosso CPF. E para que serve? Serve para identificarmos aquele elemento, seja utilizando scripts ou estilizando com CSS. Na *linha 10*, criamos um parágrafo `<p>` onde o resultado será exibido. Na *linha 11*, criamos um botão com a tag `<button>` e utilizamos também o evento **onclick = Soma()**, ou seja, ao clicarmos no botão SOMA ele “chamará” a função **Soma()**. Agora, você não precisa se preocupar em entender o que é e para que serve uma função, pois ainda estamos aprendendo os conceitos básicos. Em outro momento, falaremos da função em detalhes. O botão também poderia ter sido criado desta forma: `<input type="button" value="Soma" onclick="Soma()">`. Na *linha 13*, estamos associando o documento de HTML a um documento de JavaScript externo. Agora vamos analisar o nosso script:

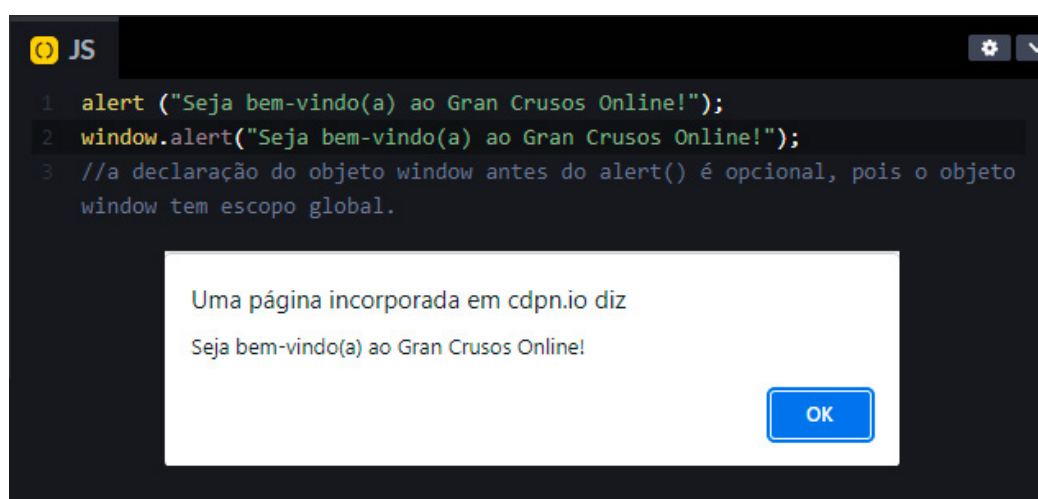
```
1  function Soma()
2  {
3      var num1 = parseInt(document.getElementById("n1").value);
4      var num2 = parseInt(document.getElementById("n2").value);
5      var soma = num1 + num2;
6      document.getElementById("res").innerHTML = "Resultado: " + soma;
7  }
```

No arquivo de JavaScript criamos uma função (*function*) chamada Soma() e nela somaremos os dois números digitados pelo usuário. O evento **onclick="Soma()"**, que vimos acima, executa a função Soma(). Nas *linhas 3 e 4*, estamos criando duas variáveis (num1, num2) que irão receber os valores digitados pelo usuário. A função **parseInt()** está sendo utilizada para converter a *string* recebida pela caixa de texto para um número inteiro. O método **getElementById()** está sendo utilizado para obter um elemento do documento a partir de seu atributo ID especificado, por isso colocaremos um id para cada um dos inputs no HTML. Resumindo, esta função está pegando os valores que foram digitados nos elementos que possuem os identificadores n1 e n2 e armazenando nas variáveis num1 e num2, respectivamente. Na *linha 5*, criamos uma variável **soma** para armazenar o resultado obtido por *num1 + num2*. Na *linha 6*, estamos exibindo na tela o resultado da soma, além de usar o **getElementById()** também estamos utilizando a propriedade **innerHTML** para exibir o conteúdo no HTML, neste caso, o conteúdo é o texto "Resultado" que está sendo concatenado com o resultado da variável **soma**. Pronto! Finalizamos aqui o mesmo exemplo que fizemos na aula de Algoritmos, só que agora usando a linguagem de programação JavaScript. Vale ressaltar que, este mesmo exemplo pode ser feito de outras maneiras.

EXEMPLO 2

Agora aprenderemos a manipular três caixas de diálogos que são muito comuns em JavaScript: *alert*, *confirm* e *prompt*:

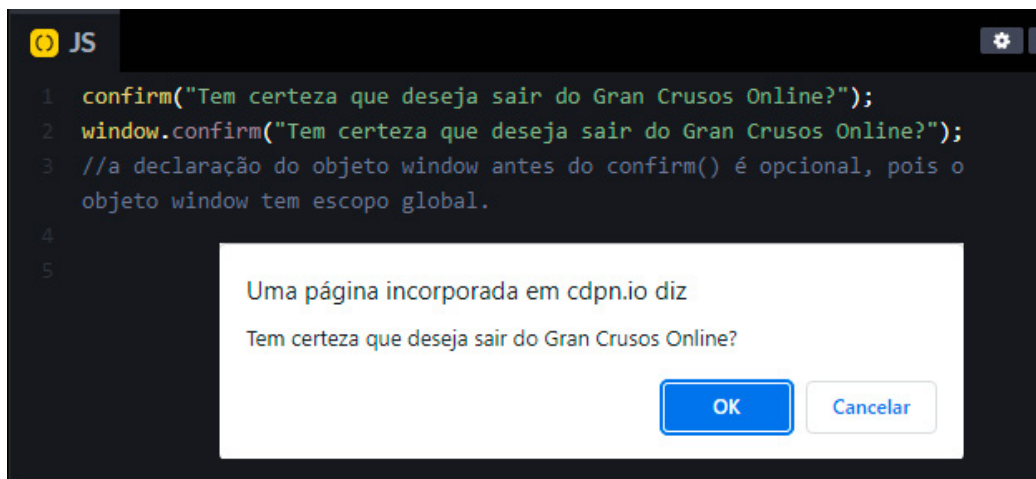
- **alert()**: "o método alert() exibe uma mensagem no navegador por meio de uma caixa de diálogo, que nada mais é que uma pequena janela *popup*" (NOLETO, 2022). O método alert() pode ser inserido de duas formas diferentes, como pode ser visto no exemplo abaixo.



A caixa de diálogo exibe, além do texto digitado, um botão de confirmação para indicar que a pessoa realmente leu a mensagem. Este recurso é muito utilizado para avisar sobre uma determinada falha, para dar boas-vindas em um novo cadastro efetuado em uma página e muitas mais (NOLETO, 2022).

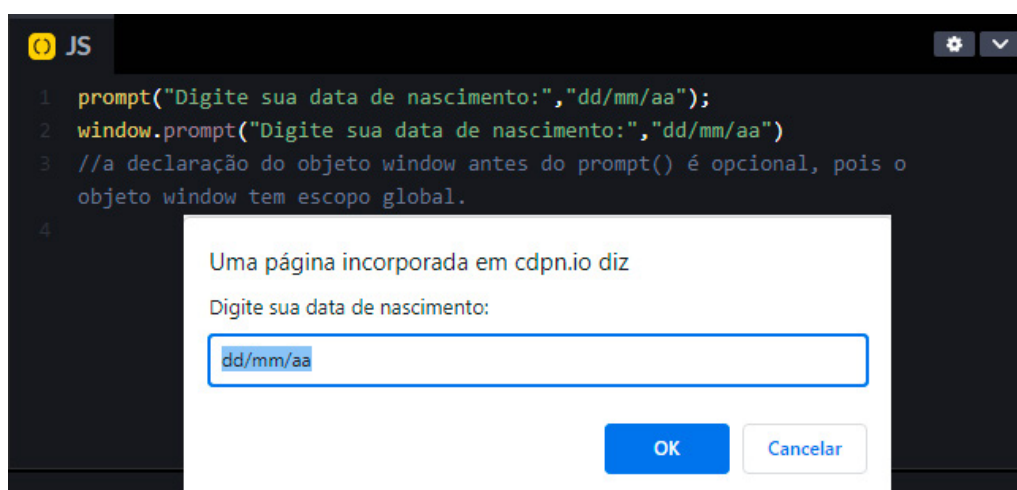
- **confirm()**: o método confirm():

(...) permite que a pessoa usuária da página decida se deseja ou não executar uma ação determinada. Para isso, exibe uma janela modal com uma mensagem e dois botões: um de confirmação e outro que cancela a ação (NOLETO, 2022).



Vale ressaltar que, se a sua opção for o botão "OK" o método confirm() retornará VERDADEIRO, se for o botão "CANCELAR" o método confirm() retornará FALSO, por isso, se faz necessário utilizar uma variável para armazenar o retorno e decidir o que será a partir desta resposta. Outro ponto importante é que no método confirm() não é possível alterar o texto dos botões OK e CANCELAR.

- **prompt()**: o método prompt() é muito utilizado para obter alguma informação do usuário, pois exibe uma caixa de diálogo com uma caixa de texto, onde o usuário pode entrar com a informação solicitada.



Neste caso, também se faz necessário utilizar uma variável para armazenar o valor digitado pelo usuário.



PRA PRATICAR

Aproveite para deixar como desafio o seguinte exercício: tente aprimorar o código do exemplo 1 utilizando o `prompt()` para pegar os números digitados pelo usuário e o `alert()` para exibir o resultado da soma.



LINK

Acesse o link: <https://blog.betrybe.com/javascript/javascript-alert/> para visualizar e testar aplicações reais utilizando os métodos `alert()`, `confirm()` e `prompt()`.



EXEMPLO 3

Para finalizar o nosso capítulo de mão na massa, vamos fazer um exemplo baseado no exercício da W3C disponível em: https://www.w3schools.com/js/js_intro.asp. Segue o link das imagens que vamos utilizar para reproduzir o exemplo:

Lâmpada acesa – https://www.w3schools.com/js/pic_bulbon.gif

Lâmpada apagada – https://www.w3schools.com/js/pic_bulboff.gif

Neste exemplo veremos como podemos utilizar o JavaScript para alterar valores de atributo HTML. Neste caso, o JavaScript vai alterar o valor do atributo **src** (source) de uma tag ``.

```

1  <html lang="pt-br">
2  <head>
3      <meta charset="UTF-8">
4      <title>Exemplo JS</title>
5  </head>
6  <body>
7      <h2>O que o JavaScript pode fazer?</h2>
8      <p>JavaScript pode alterar os valores de atributo HTML.</p>
9      <p>Nesse caso, o JavaScript altera o valor do atributo src
      (source) de uma imagem.</p>
10
11     <button onclick="document.getElementById('imagem').src='https://
      www.w3schools.com/js/pic_bulbon.gif'">Acende Lâmpada</button>
12
13     
14
15     <button onclick="document.getElementById('imagem').src='https://
      www.w3schools.com/js/pic_bulboff.gif'">Apaga Lâmpada</button>
16
17 </body>
18 </html>

```

Link para ter acesso ao código: <https://github.com/GRANCodigo/PraticaDeProgramacao/blob/cc91e7a2747dfcb7c69e862ea4623eabfe0140a2/Unidade1Aula5b>

Também usamos o método **getElementById()** e do evento **onclick**, vistos no Exemplo 1. Ao clicar no botão “Acende Lâmpada”, o método **getElementById()** altera o atributo **src** da tag **** com o caminho da lâmpada acesa. Ao clicar no botão “Apaga Lâmpada”, o método **getElementById()** altera o atributo **src** da tag **** com o caminho da lâmpada apagada. Legal, né? Veja como fica:

O que o JavaScript pode fazer?

JavaScript pode alterar os valores de atributo HTML.

Nesse caso, o JavaScript altera o valor do atributo src (source) de uma imagem.



Acende Lâmpada

Apaga Lâmpada

O que o JavaScript pode fazer?

JavaScript pode alterar os valores de atributo HTML.

Nesse caso, o JavaScript altera o valor do atributo src (source) de uma imagem.



Acende Lâmpada

Apaga Lâmpada

Agora vamos fazer o mesmo exemplo de forma diferente, utilizaremos um documento de JS externo e os botões chamarão as funções presentes neste documento. Para isso, nosso HTML deve ficar assim:

```

1  <html lang="pt-br">
2  <head>
3    <meta charset="UTF-8">
4    <title>Exemplo JS</title>
5  </head>
6  <body>
7    <h2>O que o JavaScript pode fazer?</h2>
8    <p>JavaScript pode alterar os valores de atributo HTML.</p>
9    <p>Nesse caso, o JavaScript altera o valor do atributo src
    (source) de uma imagem.</p>
10
11    <button onclick="acendeLampada()">Acende Lâmpada</button>
12
13    
14
15    <button onclick="apagaLampada()">Apaga Lâmpada</button>
16
17    <script src="teste.js"></script>
18
19  </body>
20  </html>

```

Lembrando que, é necessário estabelecer o elo entre o HTML e o JavaScript, como pode ser visto na *linha 17*. Repare que neste caso, os eventos **onclick** presentes nos botões apontam para as funções (*linhas 11 e 15*).

Nosso documento externo de JavaScript deve ficar assim:

```

1  function acendeLampada()
2  {
3    document.getElementById('imagem').src='https://www.w3schools.com/
    js/pic_bulbon.gif'
4  }
5
6  function apagaLampada()
7  {
8    document.getElementById('imagem').src='https://www.w3schools.com/
    js/pic_bulboff.gif'
9  }

```

Ambos os exemplos fazem a mesma coisa, porém de forma diferente.

CONSIDERAÇÕES FINAIS

Nesta aula, tivemos a oportunidade de colocar a mão na massa e testar alguns dos conceitos vistos nas aulas anteriores. A ideia é que a partir de agora você consiga dar os seus próprios passos. Aproveite para explorar e editar os códigos dos exemplos acima, deixando tudo do seu jeitinho. Coloque em prática outros recursos que não foram contemplados aqui.

Resumindo, aprendemos neste capítulo como converter uma *string* em um inteiro, utilizando a função **parseInt()** e como obter um elemento do documento a partir de seu atributo ID especificado, utilizando o método **getElementById()**. Além disso, aprendemos também como manipular as caixas de diálogo *alert*, *confirm* e *prompt*.

MATERIAIS COMPLEMENTARES

Playlist JavaScript e HTML para iniciantes:

https://youtube.com/playlist?list=PLWd_VnthxxLdQyD4SiYIXFJK9nsxUFkHt

REFERÊNCIAS

NOLETO, Cairo. *Javascript alert, confirm e prompt: caixas de diálogo Popup!* 2022. Disponível em: < <https://blog.betrybe.com/javascript/javascript-alert/> >. Acesso em: 06 de nov. de 2022.