01

Campo de texto e botão

Transcrição

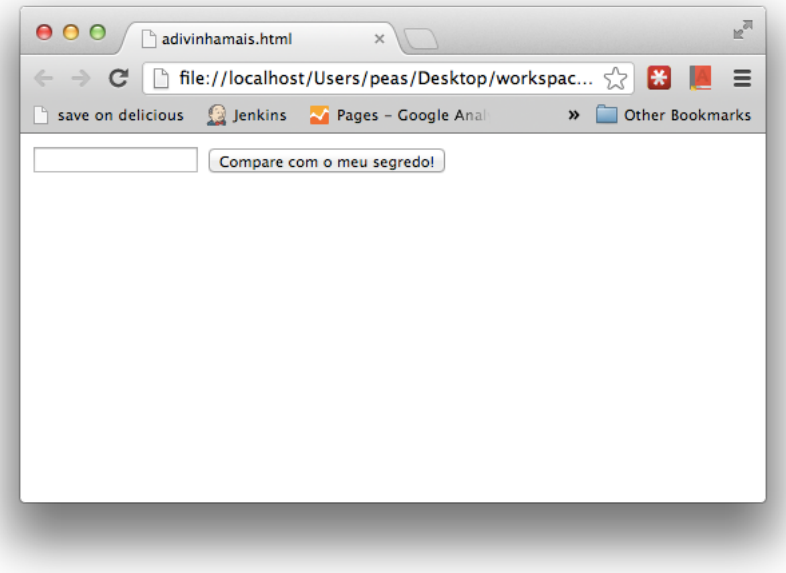
Começando deste ponto? Você pode fazer o DOWNLOAD (https://github.com/alura-cursos/logica-de-programacao-I/archive/aula-8.zip) do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

É comum que o JavaScript trabalhe com itens e campos que aparecem no HTML. Vamos criar um novo arquivo, que será a nova versão do exercício de adivinhação. Chamaremos ele de `adivinha_mais.html` e como vamos aprender novos conceitos, o jogo será constituído aos poucos.

Em primeiro lugar, vamos usar um novo item HTML o chamado `input`, ele servirá para capturar a entrada do usuário e um `button` para executar a comparação. Observe:

```
<input/>
<button>Compare com o meu segredo!</button>
```

Agora, abra a página do seu navegador. Veja que temos uma caixa de texto e um botão! Esse é primeiro **formulário**!



Clique no botão! O que acontece? Nada!

Em grande parte das páginas da internet nas quais armazenamos dados, quando clicamos em um botão, os dados são enviados para outros computadores, conhecidos como servidores, para que as informações sejam armazenadas. Mas, podemos trabalhar também com dados em nosso próprio computador, usando o JavaScript.

Agora, vamos começar nosso código JavaScript! O primeiro passo é acessar a caixa de texto que contém o número e dar um `alert`. Eis o código completo:

```
<input/>
<button>Compare com o meu segredo!</button>
```

```
<script>
    var input = document.querySelector("input");
    alert(input.value);
</script>
```

Note que muita coisa aconteceu em nosso código! O `document.querySelector("input")` é a forma de fazer com que o JavaScript vasculhe o código HTML e encontre um elemento representado pela tag `input`. Essa variável ainda não é o valor que digitamos, ela corresponde a caixa de texto em si! Fazendo o `input.value` acessamos o valor do conteúdo da caixa, isto é, o que é digitado dentro dela.

Abra o arquivo, o que acontece agora? O `alert` é executado logo que abrimos a página, sem nos dar tempo de preencher um valor na caixa e por isso mostra algo vazio! Mas, não queremos fazer o `alert` logo que a página for carregada. Nosso objetivo é chamar o `alert` **quando o botão for clicado**!

Para isso, vamos criar a função que queremos que seja executada quando o botão for clicado. Remova o `alert` e adicione:

```
function verifica() {
    alert(input.value);
}
```

Mas mesmo se quisermos abrir a página, digitar o número e clicar no botão, nada ocorre!

O que está faltando? Falta definir que quando o usuário clicar no botão "adivinhar", a função `verifica` deve ser chamada! Fazemos isso de uma forma muito simples, adicionando as seguintes linhas:

```
var button = document.querySelector("button");
button.onclick = verifica;
```

Na primeira linha pegamos o botão e guardamos ele na variável `button`. A segunda linha é a importante, pois ela avisa quando o botão "adivinhar" foi clicado (`onclick`) e assim chama a função `verifica`. É muito frequente relacionar um determinado acontecimento a execução de uma função. Esses acontecimentos são frequentemente chamados de **eventos** e as funções a serem executadas em determinados eventos são conhecidas como **funções de callback**. Essa nomenclatura aparecerá cada vez com mais frequência nos seus estudos.

E a adivinhação? Bem, ficou faltando guardar o número que pensamos para depois comparar com o digitado. Vamos revisar o código já inserindo essa funcionalidade.

Vamos refazer o código, mas dessa vez utilizaremos o jogo. Primeiro passo, criar o arquivo `adivinha_mais.html` e nele começaremos declarando dois elementos HTML: a caixa que conterá o número e o botão que verifica se o usuário acertou. Observe como fica o código:

```
<input/>
<button>Compare com o número que estou pensando!</button>
```

Agora começamos com o SCRIPT, em vez de sortear um número, vamos deixá-lo fixo por uma questão de simplicidade:

```
<script>
    var segredo = 8;
```

Pegamos o elemento do HTML por meio do código grande, o chamado `document.querySelector()`.

```
var input = document.querySelector("input");
```

Criamos uma função para verificar se o número digitado (que é `input.value`) é igual ao `segredo`. Dependendo do resultado, imprimimos diferentes respostas:

```
function verifica() {
    if(segredo == input.value) {
        alert("Parabéns! Você acertou o número secreto");
    }
    else {
        alert("Infelizmente você errou!");
    }
}
```

Falta o último passo, que é fazer com que o botão quando clicado, isto é, assim que o evento de clique aconteça, chame a função `verifica`. Observe :

```
var button = document.querySelector("button");
button.onclick = verifica;
</script>
```

Como já vimos, esse tipo de função que recebe a notificação de um evento (por exemplo o clique do mouse, o teclar de uma letra, o clique da direita do mouse, o mexer da barra de rolagem, etc) chama-se *callback*.