

02

Mostrando o placar

Transcrição

Quando começamos o jogo, no nosso placar não há nenhuma informação relevante para o usuário. O jogo poderia ser iniciado com o placar escondido, e com um botão para mostrá-lo ou escondê-lo. Então vamos dar essa opção ao usuário, ele que escolherá se quer ver o placar ou não.

Vamos começar adicionando um novo botão, do lado do botão de reiniciar. Então, na página `principal.html`, dentro da `div btoes`, adicionamos o botão:

```
<div class="btoes">
  <a id="botao-reiniciar" class="btn-floating btn-large waves-effect waves-light red">
    <i class="material-icons">restore</i>
  </a>
  <!-- NOVO BOTÃO AQUI -->
  <a id="botao-placar" class="btn-floating btn-large waves-effect waves-light green">
    <i class="material-icons">assignment</i>
  </a>
</div>
```

Como queremos que o jogo comece com o placar escondido, precisamos alterar o seu CSS, mais precisamente a propriedade `display`. Então, no `estilos.css`, adicionamos:

```
.placar {
  display: none;
}
```

Agora, quando clicarmos no botão que adicionamos acima, o placar é mostrado. Para fazer isso, precisamos implementar um código JavaScript, e como esse código é referente ao `placar`, vamos adicioná-lo ao arquivo `placar.js`. Como o placar será mostrado quando **clicarmos** no botão, precisamos escutar o seu evento de `click`:

```
$("#botao-placar").click(mostraPlacar);
```

Agora precisamos implementar a função `mostraPlacar`, que como o nome já diz, será a função responsável por mostrar o placar:

```
$("#botao-placar").click(mostraPlacar);

function mostraPlacar() {
```

Uma das formas de mostrarmos o placar, é justamente alterar novamente a sua propriedade CSS `display` de `none` para `block`. Já sabemos como alterar uma propriedade CSS com o jQuery:

```

$( "#botao-placar" ).click(mostraPlacar);

function mostraPlacar() {
    $(".placar").css("display", "block");
}

```

Agora podemos abrir a página **principal.html** no navegador, clicar no botão para mostrar o placar e perceber que o placar realmente é mostrado. Mas se clicarmos novamente no botão, o que acontece? Nada! O ideal é que se clicarmos novamente no botão, o placar seja escondido novamente. Logo, ao clicarmos no botão, se o placar estiver escondido, ele será exibido, se não estiver, ele será escondido.

Poderíamos fazer uma condição `if` para verificar o valor atual do `display` e invertê-lo. Mas como esse recurso de *mostrar e esconder* é tão comum, o jQuery já facilitou a nossa vida, criando a função `show`. Com isso não precisamos mais alterar o CSS no JavaScript:

```

$( "#botao-placar" ).click(mostraPlacar);

function mostraPlacar() {
    $(".placar").show();
}

```

E para esconder algo, existe a função oposta, a `hide`:

```

$( "#botao-placar" ).click(mostraPlacar);

function mostraPlacar() {
    $(".placar").hide();
}

```

Só que mais uma vez teríamos que fazer um `if` para verificar quando chamar a função `show` e quando chamar a função `hide`. Só que mais uma vez o jQuery facilita a nossa vida, criando a função `toggle`. Ela é semelhante à função `toggleClass`, que vimos no último módulo, que adiciona ou remove uma classe. Só que a `toggle` adiciona ou esconde um elemento:

```

$( "#botao-placar" ).click(mostraPlacar);

function mostraPlacar() {
    $(".placar").toggle();
}

```

Agora nossa funcionalidade está pronta! Mas ela pode ficar ainda melhor. Podemos reparar que quando clicamos no botão para exibir o placar, nosso texto é deslocado levemente para a esquerda. Isso porque quando o placar é adicionado, a barra de rolagem do navegador também aparece, ocupando um espaço e deslocando o texto. Podemos resolver isso mexendo novamente no CSS, para que o espaço da barra de rolagem **já venha na página**, independente se houver placar ou não. Logo, adicionamos o seguinte código no `estilos.css`:

```

body {
    overflow: scroll;
}

```

Para melhorar mais ainda, podemos melhorar a exibição do placar, porque atualmente ele aparece de maneira abrupta na página. Podemos animar essa transição, deixando-a mais suave. Para isso, podemos ver a função `slideDown`, que tem a mesma funcionalidade que a função `show`, mas realiza a transição de uma maneira mais suave:

```
$("#botao-placar").click(mostraPlacar);

function mostraPlacar() {
    $(".placar").slideDown();
}
```

Podemos reparar agora que o placar é mostrado aos poucos, de uma maneira mais lenta. Se queremos deixar mais suave ainda, podemos passar um tempo em milissegundos por parâmetro para a função:

```
$("#botao-placar").click(mostraPlacar);

function mostraPlacar() {
    $(".placar").slideDown(600);
}
```

Além do `slideDown`, analogamente temos o `slideUp`, para **mostrar um elemento**. No nosso caso queremos alternar entre essas duas funções, para isso existe a função `slideToggle`:

```
$("#botao-placar").click(mostraPlacar);

function mostraPlacar() {
    $(".placar").slideToggle(600);
}
```

Agora conseguimos concluir a nossa nova funcionalidade!

O que aprendemos?

- Como exibir um elemento com a função `show` e ocultá-lo com a função `hide`.
- Como alternar entre as funções `show` e `hide` com a função `toggle`.
- Como exibir e ocultar um elemento de uma maneira mais suave, com as funções `slideDown` e `slideUp`, respectivamente.
- Como alternar entre as funções `slideDown` e `slideUp` com a função `slideToggle`.