

Salvando a pontuação no placar

Transcrição

Neste capítulo, queremos que a pontuação do usuário seja salva ao final de cada jogada. Para isto, vamos montar um placar, ou seja, uma pequena tabela que deve ficar na parte de baixo da nossa aplicação que salvará o número de palavras digitadas e o nome do usuário a cada jogada.

Primeiramente, vamos montar a estrutura HTML necessária para comportar o placar. Em `principal.html`, montaremos uma `<section>`, dentro dela, colocaremos uma tabela que terá o **nome** do usuário e **número** de palavras digitadas.

```
<ul class="center">
  <li><span id="contador-caracteres">0</span> caracteres</li>
  <li><span id="contador-palavras">0</span> palavras</li>
</ul>

<section class="placar">
  <h3>Placar</h3>
  <table>
    <thead>
      <tr>
        <th>Usuário</th>
        <th>No. de palavras</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Douglas</td>
        <td>10</td>
      </tr>
    </tbody>
  </table>
</section>

<script src="js/jquery.js"></script>
<script src="js/main.js"></script>

</body>
</html>
```

Iremos inserir algumas classes da nossa *framework frontend* para que a tabela possua uma melhor aparência. Adicionaremos a classe `center` para centralizar a tabela, e também a `centered bordered` que adiciona linhas entre o nome dos jogadores.

```
<section class="placar">
  <h3 class="center">Placar</h3>
  <table class="centered bordered">
    <thead>
      <tr>
```

```
<th>Usuário</th>
<th>No. de palavras</th>
</tr>
</thead>
<tbody>
<tr>
<td>Douglas</td>
<td>10</td>
</tr>
</tbody>
</table>
</section>
```

Com a estrutura da tabela montada, podemos partir para o Javascript. Criaremos a função responsável para inserção de valores no placar. Em `main.js` criaremos a função `inserePlacar()` :

```
function inserePlacar(){
  console.log("Linha inserida no placar");
}
```

Devemos chamar esta função ao término do nosso jogo, ou seja, quando o tempo chegar a 0. Sabemos que esse evento ocorre na função `inicializaCronometro()` , em seu `if` :

```
function inicializaCronometro() {
  var tempoRestante = $("#tempo-digitacao").text();
  campo.one("focus", function() {
    var cronometroID = setInterval(function() {
      tempoRestante--;
      $("#tempo-digitacao").text(tempoRestante);
      if (tempoRestante < 1) {
        campo.attr("disabled", true);
        clearInterval(cronometroID);
        campo.addClass("campo-desativado");
      }
    }, 1000);
  });
}
```

Na função `inserePlacar()` , precisamos buscar pelo corpo da tabela, que é a área em que iremos inserir novas linhas com a pontuação do usuário. Vamos primeiro selecionar a `section` placar inteira:

```
function inserePlacar(){
  var placar = $(".placar");
}
```

Em seguida, queremos obter o `tbody` que fica dentro da `<table>` , que por sua vez está dentro da `<section>` . Quando queremos buscar filhos de um elemento que já selecionamos previamente, podemos utilizar a função `.find()` do jQuery, que funciona de modo semelhante a função seletora (`$`), porém fazendo a busca apenas do filho do elemento:

```
function inserePlacar(){
  var placar = $(".placar");
```

```
var corpoTabela = placar.find("tbody");  
}
```

A função `.find()` recebe como parâmetro seletores CSS e busca em seus filhos algum elemento que atenda aquela busca. Podemos simplificar e fazer tudo em uma linha apenas:

```
function inserePlacar(){  
  var corpoTabela = $(".placar").find("tbody");  
}
```

Criando uma nova linha

Como queremos criar uma nova linha na tabela, com o nome do usuário e o número de palavras, precisamos pegar essas informações:

```
function inserePlacar(){  
  var corpoTabela = $(".placar").find("tbody");  
  var usuario = "Seu-nome";  
  var numPalavras = $("#contador-palavras").text();  
}
```

Com os dados obtidos, daremos início à criação da nossa linha, que deve ser um `<tr>` com dois `<td>` dentro, cada um com uma informação. Vamos começar criando esta estrutura como uma string:

```
function inserePlacar(){  
  var corpoTabela = $(".placar").find("tbody");  
  var usuario = "Seu-nome";  
  var numPalavras = $("#contador-palavras").text();  
  
  var linha = "<tr>"+  
    "<td>"+ usuario + "</td>"+  
    "<td>"+ numPalavras + "</td>"+  
    "</tr>";  
}
```

Repare que concatenamos pedaços da string com o nossas variáveis que continham os dados, pois nosso objetivo final é ter algo assim:

```
<tr>  
  <td>Douglas</td>  
  <td>99</td>  
</tr>
```

Adicionando HTML na página com jQuery

Com nossa linha criada, precisamos adicioná-la ao corpo da tabela, e para isto vamos utilizar a função `.append()` do jQuery. Esta função adiciona a string ou elemento HTML que é passada como parâmetro como último filho do elemento em qual ela for chamada.

Vamos à **linha** como última filha do `tbody` :

```
function inserePlacar(){
    var corpoTabela = $(".placar").find("tbody");
    var usuario = "Seu-nome";
    var numPalavras = $("#contador-palavras").text();

    var linha = "<tr>"+
                "<td>"+ usuario + "</td>"+
                "<td>"+ numPalavras + "</td>"+
                "</tr>";

    corpoTabela.append(linha);
}
```

Ao final do jogo, uma nova linha é criada e adicionada ao corpo da tabela, contendo o nome do usuário e número de palavras.

Adicionando como primeiro item

Se quisermos adicionar a nova linha como primeiro item da tabela, devemos utilizar uma função que é prima da função `.append()` , que é a `.prepend()` . Ela adiciona a string/HTML passada como primeiro filho do elemento selecionado:

```
function inserePlacar(){
    var corpoTabela = $(".placar").find("tbody");
    var usuario = "Seu-nome";
    var numPalavras = $("#contador-palavras").text();

    var linha = "<tr>"+
                "<td>"+ usuario + "</td>"+
                "<td>"+ numPalavras + "</td>"+
                "</tr>";

    corpoTabela.prepend(linha);
}
```

Assim, o placar mais recente do usuário aparece como primeiro item da lista.

O que aprendemos?

- Buscar nos filhos através da função `.find()`
- Anexar HTML como um último filho, com `.append()`
- Anexar HTML como um primeiro filho, com `.prepend()`

