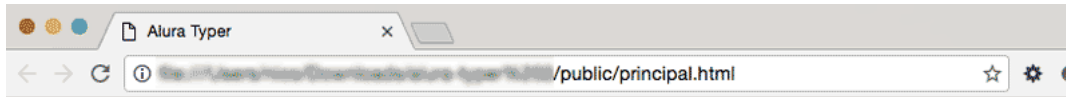


Eventos do jQuery

Transcrição

Nós já implementamos corretamente o contador de palavras da frase a ser digitada pelo jogador, O próximo passo será disponibilizarmos um campo na página para que a frase possa ser digitada. Começaremos a ter uma interatividade no programa, já que atualmente nosso jogo somente conta as palavras.



Alura Typer

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- 19 palavras
- 8 segundos

Vamos colocar um `textarea` para o usuário digitar a frase e dois contadores, um para contar quantos caracteres foram digitados, e o outro fará a contagem de quantas palavras foram digitadas.

Então, na página `principal.html`, logo após a `ul` de informações, colocaremos a `textarea`. Feito isso, criaremos uma classe denominada `campo-digitacao`.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Alura Typer</title>
</head>
<body>
  <h1>Alura Typer</h1>
  <p class="frase">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod ter

  <ul class="informacoes">
    <li><span id="tamanho-frase">19</span> palavras</li>
    <li>15 segundos</li>
  </ul>

  <textarea class="campo-digitacao" rows="8" cols="40"></textarea>

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

</body>
</html>
```

Logo abaixo, criamos uma `ul` (semelhante a que já temos) com duas `li`s, cada uma representando um contador:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Alura Typer</title>
</head>
<body>
  <h1>Alura Typer</h1>
  <p class="frase">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod ter

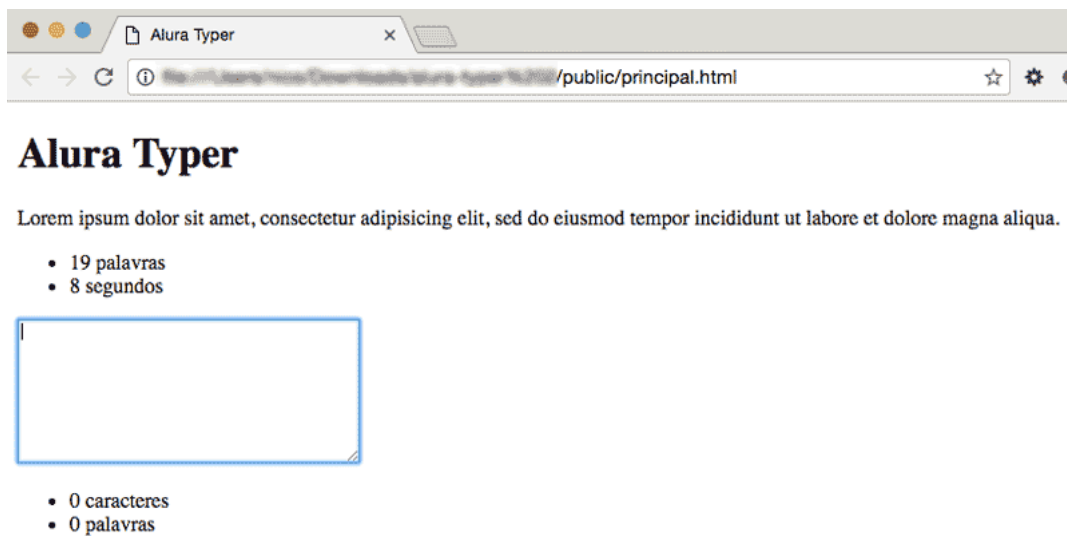
  <ul class="informacoes">
    <li><span id="tamanho-frase">19</span> palavras</li>
    <li>15 segundos</li>
  </ul>

  <textarea class="campo-digitacao" rows="8" cols="40"></textarea>

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

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

</body>
</html>
```



Após adicionarmos a `textarea` será exibido o campo em que o usuário digitará as palavras.

Trabalhando com eventos

O usuário consegue digitar a frase no campo específico, mas os contadores permanecem zerados. Faremos com que os contadores sejam atualizados toda vez que o usuário clicar no `textarea`, contando, assim, os caracteres e palavras já digitados e atualizando os respectivos `span`s.

Para fazer isso, temos que entender os famosos **eventos** do JavaScript. Como vocês já sabem, o JavaScript possui uma série de eventos: "click", "double click", "scroll", entre outras interatividades.

O jQuery facilitou nosso trabalho do momento de escutar os eventos JS dos elementos.

Para comprovar isso, vamos utilizar jQuery no caso acima, para atualizar os contadores toda vez que o usuário clicar no `textarea`.

Primeiramente, no `main.js`, se queremos detectar um evento de clique no campo, primeiro devemos selecioná-lo. Usaremos a função seletora do jQuery (`$`) e acionaremos a classe `campo-digitacao`.

```
var frase = $(".frase").text();
var numPalavras = frase.split(" ").length;
var tamanhoFrase = $("#tamanho-frase");
tamanhoFrase.text(numPalavras);

var campo = $(".campo-digitacao");
```

Agora, quando o campo for clicado, teremos que "fazer alguma coisa".

A ação de "quando o campo for..." faz referência à função `on` do jQuery, e como queremos o evento do clique no campo, passaremos `"click"` para a função:

```
var campo = $(".campo-digitacao");
campo.on("click");
```

Conseguiremos "fazer algo" quando o campo for clicado, passando um segundo parâmetro para a função `on`, no caso, uma **função anônima**, indicada pelos parênteses vazio:

```
var campo = $(".campo-digitacao");
campo.on("click", function() {

});
```

Dentro da função anônima, iremos implementar o que deve acontecer quando o usuário clicar no campo.

Ao clicarmos no campo, essa função sempre será chamada. Para comprovar isso, podemos imprimir uma mensagem no console do navegador toda vez que o campo for clicado, para isso colocaremos um `console.log` dentro da função:

```
var campo = $(".campo-digitacao");
campo.on("click", function() {
    console.log("Cliquei no campo");
});
```

Atualizamos a página, abrimos o console do navegador e vemos que toda vez que clicamos no campo, a mensagem **"Cliquei no campo"** é exibida.

Com isso, podemos partir para o nosso objetivo, que é contar as palavras. Nós já vimos anteriormente que uma maneira eficiente de contarmos as palavras, é "pegar" o conteúdo do campo, usar a função `split` no espaço e identificar o seu tamanho.

Podemos utilizar essa mesma tática, mas como saber o valor do `textarea` ?

No caso do `textarea`, o conteúdo não estará na propriedade `text` e sim no `value`, ou como é chamado pelo jQuery: `val`.

```
var frase = $(".frase").text();
var numPalavras = frase.split(" ").length;
var tamanhoFrase = $("#tamanho-frase");
tamanhoFrase.text(numPalavras);

var campo = $(".campo-digitacao");
campo.on("click", function() {
    console.log(campo.val());
});
```

Agora toda vez que clicamos no campo, o seu conteúdo é impresso no console. Lembrando que o `val` nos dá acesso ao que está dentro de uma tag de *input*, como as tags `input` e `textarea`. Já o `text` nos dá acesso ao que está dentro de uma tag de texto, como `p`, `span` e `h1`.

Contando os caracteres e as palavras digitadas

Sabemos que `val` é referente ao conteúdo da frase, agora, faremos o `split` para contar as palavras. Para testar, vamos aproveitar para imprimir a quantidade de palavras para ver se estamos contando corretamente:

```
var frase = $(".frase").text();
var numPalavras = frase.split(" ").length;
var tamanhoFrase = $("#tamanho-frase");
tamanhoFrase.text(numPalavras);

var campo = $(".campo-digitacao");
campo.on("click", function() {
    var conteudo = campo.val();
    var qtdPalavras = conteudo.split(" ").length;
    console.log(qtdPalavras);
});
```

Aparentemente está funcionando, a cada clique no campo, é impresso a quantidade de palavras nele escrita. Mas contar as palavras é insuficiente, temos que atualizar o contador, ou seja, temos que selecionar o `span` e alterar o seu `text`, atribuindo a quantidade de palavras a ele:

```
var frase = $(".frase").text();
var numPalavras = frase.split(" ").length;
var tamanhoFrase = $("#tamanho-frase");
tamanhoFrase.text(numPalavras);

var campo = $(".campo-digitacao");
campo.on("click", function() {
    var conteudo = campo.val();
    var qtdPalavras = conteudo.split(" ").length;

    $("#contador-palavras").text(qtdPalavras);
});
```

Podemos escrever a frase, e a cada clique o contador é atualizado corretamente.

Falta ainda contar os caracteres, uma tarefa bem simples. Pegaremos o conteúdo do campo, e o seu tamanho (`length`) será a quantidade de caracteres. Usaremos isso para atualizar o seu contador:

```
var campo = $(".campo-digitacao");
campo.on("click", function() {
    var conteudo = campo.val();

    var qtdPalavras = conteudo.split(" ").length;
    $("#contador-palavras").text(qtdPalavras);

    var qtdCaracteres = conteudo.length;
    $("#contador-caracteres").text(qtdCaracteres);
});
```

O evento input

No entanto, para atualizar os contadores, temos sempre que clicar dentro do campo. O ideal seria que o contador fosse atualizado **enquanto** o usuário digita. E para isso existe um evento específico de quando digitamos, colocamos dados em um campo: `input` :

```
var campo = $(".campo-digitacao");
campo.on("input", function() {
```

Observe que à medida que digitarmos, o campo será atualizado. Era justamente esse o nosso objetivo.

##Separando com Regexp

A contagem tanto das palavras quanto dos caracteres parece estar funcionando, mas se analisarmos atentamente, quando apagamos alguma frase, o contador de palavras ainda conta uma. Além disso, se pressionarmos várias vezes a barra de espaço, o contador irá considerar que um deles equivale a uma palavra.

Isso tem relação com o modo que estamos contando as palavras, fazendo um *split* em um espaço vazio.

Para sermos mais precisos na contagem, utilizaremos uma expressão regular em vez dos espaço vazio. A expressão regular será responsável por buscar qualquer caractere, exceto espaço vazio: `/\S+/` .

```
var qtdPalavras = conteudo.split(/\S+/).length;
$("#contador-palavras").text(qtdPalavras);
```

Agora os espaços não são mais considerados como palavras, mas a contagem sempre mostra soma mais uma no cálculo. Resolveremos isso, subtraindo `-1` do `length` do conteúdo:

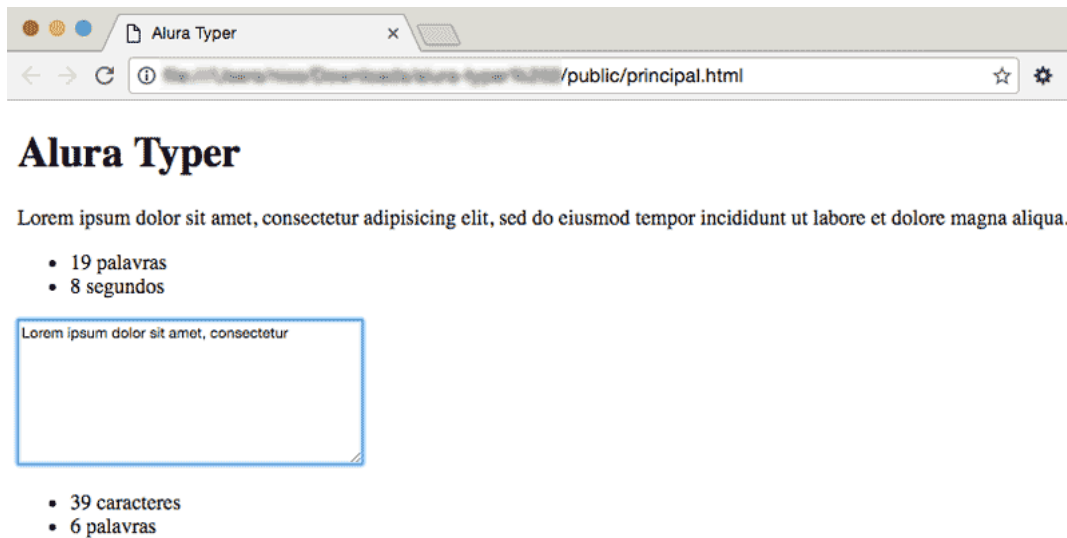
```
var frase = $(".frase").text();
var numPalavras = frase.split(" ").length;
var tamanhoFrase = $("#tamanho-frase");
tamanhoFrase.text(numPalavras);
```

```
var campo = $(".campo-digitacao");
campo.on("input", function() {
    var conteudo = campo.val();

    var qtdPalavras = conteudo.split(/\S+/).length - 1;
    $("#contador-palavras").text(qtdPalavras);

    var qtdCaracteres = conteudo.length;
    $("#contador-caracteres").text(qtdCaracteres);
});
```

Podemos testar e ver tudo funcionando como desejado.



O que aprendemos?

- Implementar o campo de digitação
- Adicionando os contadores de caracteres e palavras a página
- Escutar eventos com a função `.on()` do jQuery
- O evento de click
- O evento de input
- Como melhorar a contagem de palavras com uma expressão regular