

Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Design Hipermedia

1º Ano/1º Semestre

Docente: Marco Miguel Olival Olim

Data 30/11/2017

ESTE EXERCÍCIO PRETENDE EXPLORAR AS DIFERENTES ABORDAGENS À PROGRAMAÇÃO COM JAVASCRIPT NO CONTEXTO DO DESENVOLVIMENTO WEB E A ADOÇÃO DE BIBLIOTECAS DE JAVASCRIPT (EX: JQUERY)

Segundo a Wikipédia, o **Javascript** é “uma linguagem de programação interpretada, originalmente implementada como parte dos navegadores web (browser) para que scripts pudessem ser executados do lado do cliente e interagissem com o utilizador sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido. É, por isso, a principal linguagem para programação client-side em navegadores web” e uma das três tecnologias de base para a produção de conteúdos para a web, juntamente com o **html** e o **css** já estudados.

Para ser utilizado num browser não carece de qualquer tipo de instalação prévia, bastando a sua inserção numa tag <script>. Neste exercício vamos utilizar o Javascript para adicionar css a um documento de html:

```
body {
  background-color: #fff;
  color: #000;
  font-family: Helvetica, Arial, sans-serif;
}

h1,
h2,
h3 {
  margin-bottom: .2em;
}

.poem {
  margin: 0 2em;
}

.highlight {
  background-color: #ccc;
  border: 1px solid #888;
  font-style: italic;
  margin: 0.5em 0;
  padding: 0.5em;
}
```

Cofinanciado por:

A primeira página deste site deverá ter o seguinte html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Through the Looking-Glass</title>
  <link rel="stylesheet" href="jstojq.css">
</head>

<body>
  <h1>Through the Looking-Glass</h1>
  <div class="author">by Lewis Carroll</div>
  <div class="chapter" id="chapter-1">
    <h2 class="chapter-title">1. Looking-Glass House</h2>
    <p>There was a book lying near Alice on the table, and while she sat watching
the White King (for she was still a little anxious about him, and had the ink all ready
to throw over him, in case he fainted again), she turned over the leaves, to find
    some part that she could read, <span class="spoken">
"&mdash;for it's all in some language I don't know,"
    </span> she said to herself.</p>
    <p>It was like this.</p>
    <div class="poem">
      <h3 class="poem-title">YKCOWREBBAJ</h3>
      <div class="poem-stanza">
        <div>sevot yhtils eht dna ,gillirb sawT'</div>
        <div>;ebaw eht ni elbmig dna eryg diD</div>
        <div>,sevogorob eht erew ysmim lla</div>
        <div>.ebargtuo shtar emom eht dnA</div>
      </div>
    </div>
    <p>She puzzled over this for some time, but at last a bright thought struck her.
<span class="spoken">
"Why, it's a Looking-glass book, of course! And if
I hold it up to a glass, the words will all go the
right way again."</span></p>
    <p>This was the poem that Alice read.</p>
    <div class="poem">
      <h3 class="poem-title">JABBERWOCKY</h3>
      <div class="poem-stanza">
        <div>'Twas brillig, and the slithy toves</div>
        <div>Did gyre and gimble in the wabe;</div>
        <div>All mimsy were the borogoves,</div>
        <div>And the mome raths outgrabe.</div>
      </div>
    </div>
  </div>
</body>

</html>
```

Cofinanciado por:



Vamos utilizar Javascript para adicionar a classe “highlight” ao <div> que já contém a classe “poem-stanza”:

```
<script>
  window.onload = function() {
    var divs = document.getElementsByTagName('div');
    for (var i = 0; i < divs.length; i++) {
      if (hasClass(divs[i], 'poem-stanza') && !hasClass(divs[i], 'highlight')) {
        divs[i].className += ' highlight';
      }
    }
    function hasClass(elem, cls) {
      var reClass = new RegExp(' ' + cls + ' ');
      return reClass.test(' ' + elem.className + ' ');
    }
  };
</script>
```

O site deverá agora ter aplicado os seguinte estilos:

Through the Looking-Glass

by Lewis Carroll

1. Looking-Glass House

There was a book lying near Alice on the table, and while she sat watching the White King (for she was still a little anxious about him, and had the ink all ready to throw over him, in case he fainted again), she turned over the leaves, to find some part that she could read, “—for it’s all in some language I don’t know,” she said to herself.

It was like this.

YKOWREBBAJ

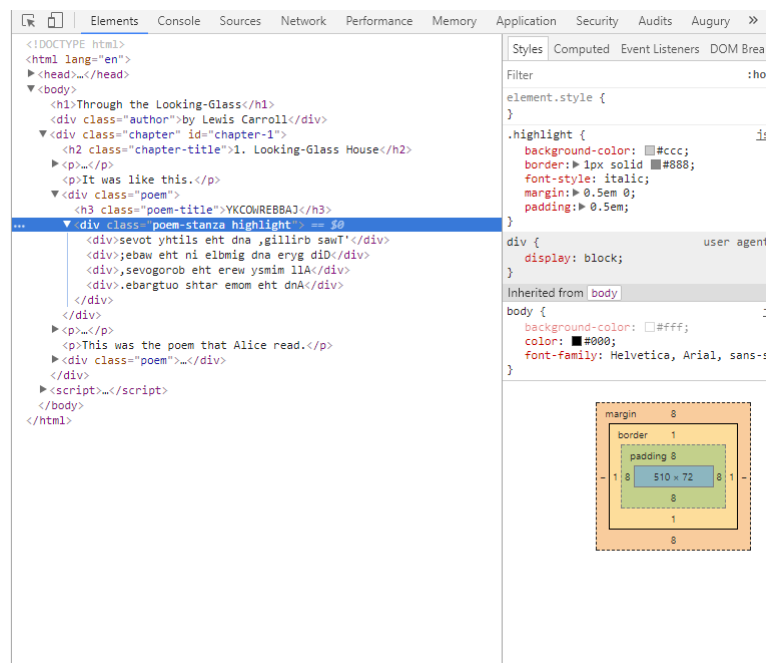
sevot yhtils eht dna ,gillirb sawT
;ebaw eht ni elbmig dna eryg diD
;sevogorob eht erew ysmim lIA
;ebartuo shtar emom eht dnA

She puzzled over this for some time, but at last a bright thought struck her. “Why, it’s a Looking-glass book, of course! And if I hold it up to a glass, the words will all go the right way again.”

This was the poem that Alice read.

JABBERWOCKY

’Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.



Esta inserção dinâmica de uma classe parece ser uma operação relativamente comum e, no entanto, dá origem a um código de javascript relativamente complexo. Com vista a proporcionar uma abstração genérica e mais inteligível surgiram então bibliotecas de javascript, como o jQuery, para simplificar a programação deste tipo de tarefas. Neste caso, o script de jQuery equivalente seria:

```
$(function() {
  jQuery('div.poem-stanza').addClass('highlight');
});
```

Refira-se que o jQuery não é mais do que javascript e pode ser escrito de diferentes formas de acordo com a sintaxe da linguagem. A própria função nuclear, jQuery(), pode ser representada pela notação equivalente \$().

Cofinanciado por:



Note-se que embora o javascript seja nativo ao browser, o jQuery precisa de ser importado para poder ser utilizado:

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>

<script>
  function addHighlightClass() {
    $('div.poem-stanza').addClass('highlight');
  }
  $(document).ready(addHighlightClass);
</script>
```

O jQuery já tem mais de 12 anos de evolução e, consequentemente, conta já com um considerável número de métodos implementados que servem, essencialmente, os seguintes propósitos:

- **Aceder a elementos no DOM (Document Object Model)**

```
$('div.content').find('p');
```

- **Modificar estilos CSS**

```
$('ul > li:first').addClass('active');
```

- **Alterar o próprio html**

```
$('#container').append('<a href="more.html">more</a>');
```

- **Proporcionar resposta à interação com o utilizador**

```
$('#button.show-details').click(function() {
  $('#div.details').show();
});
```

- **Animações de elementos**

```
$('#div.details').slideDown();
```

Cofinanciado por:



- **Ajax (Asynchronous JavaScript and XML)** - obter dados do servidor sem que seja necessário carregar novamente a página de html

```
$('#div.details').load('more.html #content');
```

- **Simplificar as tarefas mais comuns realizadas com JavaScript**

```
$.each(obj, function(key, value) {  
    total += value;  
});
```

No exercício que se segue pretende-se implementar uma funcionalidade de zoom ao texto do html. Para o efeito adicionamos 3 botões para o utilizador poder interagir com o documento:

```
<body>  
  
  <div id="switcher" class="switcher">  
    <button id="switcher-default">Default</button>  
    <button id="switcher-reduz">Reduz</button>  
    <button id="switcher-aumenta">Aumenta</button>  
  </div>  
  
  <h1>Through the Looking-Glass</h1>
```

Primeiramente adicionamos as novas classes ao ficheiro de CSS existente:

```
.aumenta {  
    font-size: 1.5em;  
}  
  
.reduz {  
    font-size: 0.5em;  
}  
  
.selecionado {  
    font-weight: bold;  
}
```

Cofinanciado por:



O método jQuery utilizado é basicamente o mesmo já enunciado anteriormente:

```
$('#body').addClass('aumenta');
```

Para associar ao botão “aumenta” utilizamos o método on()

```
$(document).ready(function() {  
    $('#switcher-large').on('click', function() {  
        $('#body').addClass('aumenta');  
    });  
});
```

Sempre que aplicamos uma classe temos, neste caso, de remover as já aplicadas. Sendo assim, implementamos os restantes botões da seguinte forma:

```
$(document).ready(function() {  
    $('#switcher-default').on('click', function() {  
        $('#body').removeClass('reduz ');  
        $('#body').removeClass('aumenta ');  
    });  
    $('#switcher-reduz').on('click', function() {  
        $('#body').addClass('reduz ');  
        $('#body').removeClass('aumenta ');  
    });  
    $('#switcher-aumenta').on('click', function() {  
        $('#body').removeClass('reduz ');  
        $('#body').addClass('aumenta ');  
    });  
});
```

Para distinguir qual o botão selecionado aplicamos a estes a classe “selecionado”:

```
$(this).addClass('selected');
```

Para remover esta classe dos outros botões podemos tirar partido do princípio de interação implícita característica do jQuery que, neste caso, retira a classe “selecionado” de todos os botões incluídos na tag com o id “switcher”:

```
$('#switcher button').removeClass('selecionado');
```

Cofinanciado por:



Aplicando este princípio a todos os botões:

```
$(document).ready(function() {  
    $('#switcher-default')  
        .addClass('selecionado')  
        .on('click', function() {  
            $('body').removeClass('reduz');  
            $('body').removeClass(' aumenta ');  
            $('#switcher button').removeClass(' selecionado ');  
            $(this).addClass('selected');  
        });  
    $('#switcher-reduz ').on('click', function() {  
        $('body').addClass(' reduz ');  
        $('body').removeClass(' aumenta ');  
        $('#switcher button').removeClass(' selecionado ');  
        $(this).addClass(' selecionado ');  
    });  
    $('#switcher-aumenta ').on('click', function() {  
        $('body').removeClass(' reduz ');  
        $('body').addClass(' aumenta ');  
        $('#switcher button').removeClass(' selecionado ');  
        $(this).addClass(' selecionado ');  
    });  
});
```

Podemos denotar que o código é pouco eficiente pois existem muitas repetições. Numa primeira tentativa de **refactoring** deste código podemos isolar o procedimento de selecionar o texto do botão, podemos também encadear os próprios métodos na mesma instrução, além de colocar diversas instruções em fila:

```
$(document).ready(function() {  
    $('#switcher-default')  
        .addClass(' selecionado ')  
        .on('click', function() {  
            $('body').removeClass(' reduz ').removeClass(' aumenta ');  
        });  
    $('#switcher-reduz ').on('click', function() {  
        $('body').addClass(' reduz ').removeClass(' aumenta ');  
    });  
    $('#switcher-aumenta ').on('click', function() {  
        $('body').removeClass(' reduz ').addClass(' aumenta ');  
    });  
    $('#switcher button').on('click', function() {  
        $('#switcher button').removeClass(' selecionado ');  
        $(this).addClass(' selecionado ');  
    });  
});
```

Cofinanciado por:



Numa segunda interação podemos utilizar o método `removeClass()` sem parâmetros, pois desta forma este remove todas as classes indiferenciadamente:

```
$(document).ready(function() {
    $('#switcher-default')
        .addClass(' selecionado ')
        .on('click', function() {
            $('body').removeClass();
        });
    $('#switcher-reduz ').on('click', function() {
        $('body').removeClass().addClass(' reduz ');
    });
    $('#switcher-aumenta ').on('click', function() {
        $('body').removeClass().addClass(' aumenta ');
    });
    $('#switcher button').on('click', function() {
        $('#switcher button').removeClass(' selecionado ');
        $(this).addClass(' selecionado ');
    });
});
```

Ainda existe repetições nos botões que poderiam ser transferidas para o botão genérico:

```
$(document).ready(function() {
    $('#switcher-default').addClass(' selecionado ');
    $('#switcher button').on('click', function() {
        $('body').removeClass();
        $('#switcher button').removeClass(' selecionado ');
        $(this).addClass(' selecionado ');
    });
    $('#switcher-reduz').on('click', function() {
        $('body').addClass('reduz');
    });
    $('#switcher-aumenta').on('click', function() {
        $('body').addClass('aumenta');
    });
});
```

Cofinanciado por:



Por último, refira-se que podemos deixar de utilizar os próprios identificadores das classes (handlers) ao explorar o contexto destes eventos, nomeadamente com a utilização do `this` para retornar o próprio elemento do DOM. Numa nota final, refira-se a existência de abreviaturas no jQuery, como por exemplo o `.click()` que substitui o `.on('click',...)`.

```
$(document).ready(function() {  
    $('#switcher-default').addClass(' selecionado ');  
    $('#switcher button').click(function() {  
        var bodyClass = this.id.split('-')[1];  
        $('body').removeClass().addClass(bodyClass);  
        $('#switcher button').removeClass(' selecionado ');  
        $(this).addClass(' selecionado ');  
    });  
});
```

Para o caso de pretender esconder elementos podemos utilizar esta técnica de alternar classes. Primeiro criamos uma classe para esconder tudo:

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

Seguidamente colocamos o botão que efetua esta operação

```
<button id="switcher-toggle">X</button>
```

Por ultimo usamos o `.toggleClass()` para inserir/remover esta classe

```
$('#switcher-toggle').click(function() {  
    $('#switcher').toggleClass('hidden');  
});
```

Como exercício final, procure aplicar estes conceitos ao site da UMaCantina

Cofinanciado por:

