

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

jQuery

Curso de Tecnologia em Análise e Desenvolvimento de
Sistemas

Desenvolvimento de Sistemas web

Introdução

- jQuery é uma biblioteca JavaScript que tem como objetivo principal facilitar a programação do lado cliente. jQuery simplifica tarefas como manipulação de elementos HTML, animações e Ajax.
- Além de facilitar a escrita de código, jQuery possui outras características interessantes:
 - Cross-browser.
 - Biblioteca de componentes (jQuery UI).
 - Open source.
 - Grande base de usuários.
 - jQuery é provavelmente a biblioteca JavaScript mais popular da atualidade.

Utilizando jQuery

- Por tratar-se de código JavaScript, basta incluir o arquivo da biblioteca com a tag `<script>`. O arquivo pode ser baixado do site **jQuery.com** ou estar disponível em um endereço web:

```
<script src="jquery-1.10.2.min.js">
```

```
<script
```

```
  src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
```

- A versão 2.0 é totalmente compatível com a versão 1.9, mas não suporta as versões 6, 7 e 8 do Internet Explorer.

Função `jQuery()`

- É utilizada para acessar os elementos HTML da página. Retorna um objeto **jQuery** que encapsula os elementos obtidos. Este objeto oferece métodos para manipulação em lote dos elementos obtidos.
- Seu argumento é uma string indicativa dos elementos desejados.
 - Esta string é um **seletor** css.
- O caractere **\$** é um alias para a função **jQuery()**.

Função jQuery()

- Exemplo 1: alterando a cor da fonte de todas as tags **p**.

- JS:

```
var e = document.getElementsByTagName('p');  
for(var i = 0; i < e.length; i++)  
    e[i].style.color = 'red';
```

- jQuery:

```
jQuery('p').css('color', 'red');
```

- Exemplo 2: alterando a cor da fonte e o texto da tag com id igual a **texto**.

- JS:

```
var p = document.getElementById('texto');  
p.style.color = 'red';  
p.innerHTML = 'Novo texto';
```

- jQuery:

```
$('#texto').css('color', 'red').text('Novo texto.');
```

jQuery e elementos DOM

- O objeto **jQuery** retornado pela função **\$()** encapsula elementos DOM, os quais podem ser acessados através de índices como em um array.
- Note que ao usar índices, é obtido um objeto DOM, não sendo possível a execução de métodos jQuery sobre este objeto. Para tal, basta utilizar o objeto DOM como argumento da função **\$()**.

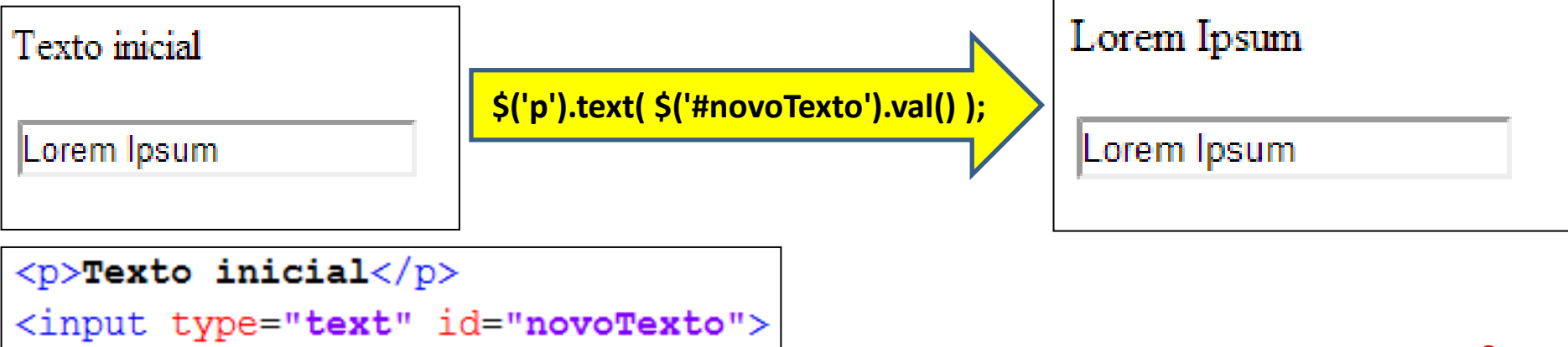
```
var e = $(':text');  
e[0].val(); //Gera erro, pois e[0] é um objeto DOM  
$(e[0]).val();
```

Alguns seletores

Seletor	Descrição	Exemplo
*	Todos os elementos da página.	<code>\$('*')</code>
elemento	Todas as ocorrências de determinada tag.	<code>\$('p')</code>
#id	O elemento com o id indicado.	<code>\$('#email')</code>
.classe	Todos os elementos associados à classe.	<code>\$('.destaque')</code>
elemento.classe	Todas as ocorrências do elemento e que estão associadas a uma classe específica.	<code>\$('p.destaque')</code>
seletor1,seletor2,s eletorN	Todos os elementos retornados pelos seletores indicados.	<code>\$('p, .destaque')</code>
[atributo]	Todos os elementos com o atributo especificado.	<code>\$('[href]')</code>
elemento[atributo]	Todas as ocorrências do elemento e que possuem o atributo especificado.	<code>\$('a[target]')</code>
elemento[atributo ='valor']	Todas as ocorrências do elemento e que possuem o atributo especificado com determinado valor.	<code>\$('a[target="_blank"]')</code>

Manipulando o DOM

- O objeto **jQuery** oferece diversos métodos para manipulação dos elementos DOM.
- Métodos para obter ou alterar o conteúdo dos elementos:
 - **text()**: conteúdo textual dos elementos.
 - **html()**: conteúdo HTML dos elementos.
 - **val()**: valores dos campos de formulário.



Manipulando o DOM

- Métodos para manipulação de atributos: **attr()**, **prop()**, **removeAttr()**, **removeProp()**, ...
- Métodos para acréscimo de conteúdo ou elementos: **append()**, **appendTo()**, **prepend()**, **prependTo()**, **after()**, **before()**, **insertAfter()**, **insertBefore()**, ...
- Métodos para remoção de elementos: **empty()**, **remove()**, **replaceAll()**, **replaceWith()**, ...

Manipulando o DOM

- exemplo1.html

Manipulação de CSS

- jQuery disponibiliza vários métodos para manipular as propriedades CSS dos elementos da página: **addClass()**, **css()**, **hasClass()**, **height()**, **width()**, **position()**, **removeClass()**, **toggleClass()**, ...
- **css('propriedade')**: retorna o valor da propriedade do primeiro elemento encapsulado pelo objeto jQuery.
- **css('propriedade', 'novoValor')**: altera o valor da propriedade de todos os elementos encapsulados pelo objeto jQuery.

Eventos

- Para cada evento DOM há um respectivo método jQuery: **click**, **dblclick**, **mouseenter**, **hover**, **keypress**, **submit**, **focus**, **load**, ...
- O objeto jQuery atribui o evento para todos os seus elementos encapsulados.

Eventos

- Atribuindo uma ação para o evento **onclick** de todos os elementos **p**:

```
<head>
  <script src="jquery-1.11.0.min.js"></script>
  <script>
    function atribuirEventos(){
      $('p').click(function(){
        $(this).css('color','red');
      });
    }
  </script>
</head>
<body onload="atribuirEventos()">
  <p>Primeiro parágrafo.</p>
  <p>Segundo parágrafo.</p>
</body>
```

Eventos

- No desenvolvimento com jQuery é comum utilizar o evento **ready** do objeto **document** para atribuir eventos aos elementos da página.

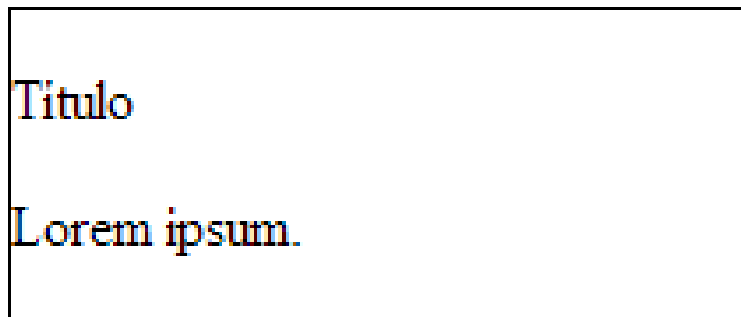
```
<head>
  <script src="jquery-1.11.0.min.js"></script>
  <script>
    $(document).ready(function(){
      $('p').click(function(){
        $(this).css('color','red');
      });
    });
  </script>
</head>
<body>
  <p>Primeiro parágrafo.</p>
  <p>Segundo parágrafo.</p>
</body>
```

Efeitos

- jQuery oferece métodos para aplicação de efeitos durante a alteração do estado de visualização de um elemento: **hide()**, **show()**, **toggle()**, **fadeIn()**, **fadeOut()**, **fade()**, **fadeToggle()**, **slide()**, ...

Efeitos

```
<body>  
  <div style="border-style:solid;border-width:1px;">  
    <p onclick="$('#conteudo').fadeToggle()">  
      Título  
    </p>  
    <p id="conteudo">Lorem ipsum.</p>  
  </div>  
</body>
```



jQueryUI

- É uma extensão de jQuery que tem como objetivo principal facilitar o desenvolvimento de páginas web ricas em termos de componentes visuais e interação com o usuário.
- Agrega componentes de UI (widgets), suporte a temas visuais, aplicação de efeitos de animação e funções de interação entre o usuário e os componentes.
- Sua página oferece um sistema de download personalizado, na qual o desenvolvedor pode especificar os recursos desejados, livrando-o do download completo da biblioteca.
 - A página também disponibiliza uma ferramenta de criação de temas.

Utilizando jQueryUI

- Diretórios do arquivo de download personalizado após sua extração: **/css**, **/development-bundle** e **/js**.
- Na página HTML, apontar para a folha de estilo do tema, para o arquivo da jQuery e para o arquivo da jQueryUI:

```
<link rel="stylesheet"
      href="css/themename/jquery-ui.custom.css" />
<script src="js/jquery.min.js"></script>
<script src="js/jquery-ui.custom.min.js"></script>
```

Inserindo componentes

- Para cada componente visual deve-se declarar o elemento HTML correspondente e o comando jQuery de configuração do componente.

```
<script>  
    $(function() {  
        $("#datepicker").datepicker();  
    });  
</script>
```

```
<input type="text" id="datepicker">
```

jQueryUI

- `exemplo2.html`

Referências

- jQuery. **API Documentation**. Disponível em <<http://jquery.com>>.
- Balduino, Plínio. **Dominando JavaScript com JQuery**. São Paulo: Casa do Código, 2013.
- w3schools. **jQuery Tutorial**. Disponível em <<http://www.w3schools.com/Jquery/default.asp>>.