

jQuery

André Tavares da Silva

andre.silva@udesc.br

jQuery

- jQuery é uma biblioteca em JavaScript;
- Seu objetivo é simplificar a programação em JavaScript;
- jQuery foi desenvolvida para ser de aprendizado fácil e rápido.

Características

- Seleção de elementos HTML;
- Manipulação de elementos HTML;
- Manipulação de estilos (CSS);
- Funções de eventos em HTML;
- Efeitos e animações em JavaScript;
- AJAX;
- Ferramentas diversas.

Inclusão da biblioteca na página

- A biblioteca jQuery está armazenada em um único arquivo *script*.

```
<head>  
<script type="text/javascript" src="jquery.js">  
</script>  
</head>
```

- A tag `<script>` deve ficar dentro da tag `<head>`.

Inclusão da biblioteca na página

- Hospedado na Google:

```
<head>  
<script type="text/javascript" src =  
"http://ajax.googleapis.com/ajax/libs/jquery/1.8.  
0/jquery.min.js">  
</script>  
</head>
```

- Ou na Microsoft:

```
http://ajax.microsoft.com/ajax/jquery/jquery-  
1.8.0.min.js
```

A sintaxe do jQuery

- A sintaxe do jQuery é feita para selecionar elementos HTML e realizar ações sobre os mesmos.
- A sintaxe básica é: **`$ (selector) .action ()`**
 - O sinal de dólar define que é jQuery.
 - O seletor é usado para "buscar" elementos HTML.
 - A função (action) jQuery é a que será executada no(s) elemento(s) HTML.

Exemplos de sintaxe

- `$(this).hide()`
 - Demonstra o método jQuery `hide()` usado para esconder o elemento HTML atual.
- `$("#test").hide()`
 - Demonstra o método jQuery `hide()` usado para esconder o elemento de `id="test"`.
- `$("p").hide()`
 - Demonstra o método jQuery `hide()` usado para esconder todos os elementos dentro das *tags* `<p>`.
- `$(".test").hide()`
 - Demonstra o método jQuery `hide()` usado para esconder todos os elemento da classe "test".

Seletor de elementos

- jQuery usa CSS para selecionar elementos HTML.
 - `$ ("p")` seleciona todos os elementos com *tag* `<p>`.
 - `$ ("p.intro")` seleciona todos os elementos com *tag* `<p>` da classe "intro".
 - `$ ("p#demo")` seleciona todos os elementos com *tag* `<p>` e `id="demo"`.

Seletor por atributos

- `$ (" [href] ")` seleciona todos os elementos com atributo href.
- `$ (" [href=' # '] ")` seleciona todos os elementos com atributo href e valor igual a "#".
- `$ (" [href!=' # '] ")` seleciona todos os elementos com atributo href e valor diferente de "#".
- `$ (" [href$=' .jpg '] ")` seleciona todos os elementos com atributo href cujo valor termina com ".jpg".

Seletor CSS

- O seletor CSS do jQuery pode ser utilizado para modificar propriedades CSS de elementos HTML.
- O exemplo abaixo muda a cor de fundo de todos os elementos p para azul:

```
$ ("p") .css ("background-color" , "blue" ) ;
```

Função Document Ready

- Em todos os exemplos a seguir, as funções jQuery estarão dentro de uma função `document.ready()`:

```
$ (document) .ready (function () {  
    // funções jQuery...  
}) ;
```

- Isto serve para prevenir que alguma função seja executada antes da leitura completa do arquivo HTML.
- Algumas ações irão falhar caso a página ainda não tenha sido lida. Por exemplo, se uma função tenta esconder um elemento que ainda não existe.

Exemplo 1

Manipulação de eventos

- Os métodos de manipulação de eventos é a função principal do jQuery.
- *Event handlers* são métodos que são chamados (disparados) quando "alguma coisa acontece" em HTML.
- É comum colocar o código de manipulação de eventos dentro da seção <head>:

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script>
```

```
<script type="text/javascript">
```

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("p").hide();  
    });  
});
```

```
});
```

```
</script>
```

```
</head>
```

Efeitos

- Hide, Show, Toggle,
- Slide: `slideDown`, `slideUp`, `slideToggle`,
- Desvanecer: `fadeIn`, `fadeOut`, `fadeTo`
- Animações customizadas: `animate`

Hide, Show e Toggle

```
$(selector).hide(speed, callback)
```

```
$(selector).show(speed, callback)
```

```
$(selector).toggle(speed, callback)
```

- Os parâmetros para especificar um tempo são: "slow", "fast", "normal" ou valor em milissegundos
- O parâmetro callback é o nome da função a ser executada após completar a execução da tarefa.

```
$("#button").click(function() {  
    $("#p").hide(1000);  
});
```

Exemplo 2

slideDown, slideUp, slideToggle

```
$(selector).slideDown(speed, callback)
```

```
$(selector).slideUp(speed, callback)
```

```
$(selector).slideToggle(speed, callback)
```

- Os parâmetros para especificar um tempo são: "slow", "fast", "normal" ou valor em milissegundos
- O parâmetro callback é o nome da função a ser executada após completar a execução da tarefa.

```
$(".flip").click(function() {  
    $(".panel").slideToggle("slow");  
});
```

Exemplo 3

fadeIn, fadeOut, fadeTo

- Muda gradualmente a opacidade dos elementos.

```
$ (selector) .fadeIn (speed, callback)
```

```
$ (selector) .fadeOut (speed, callback)
```

```
$ (selector) .fadeTo (speed, opacity, callback)
```

- Os parâmetros para especificar um tempo são: "slow", "fast", "normal" ou valor em milissegundos.
- O parâmetro callback é o nome da função a ser executada após completar a execução da tarefa.

```
$ ("button") .click (function () {  
    $ ("div") .fadeOut (4000) ;  
}) ;
```

Funções callback

- É a função a ser executada após completar a execução de uma tarefa.

```
$ ("p") .hide (1000, function () {  
    alert ("O parágrafo foi escondido!");  
});
```

Exemplo 4

Animações Customizadas

```
$(selector).animate({params}, [duration],  
[easing], [callback])
```

- O parâmetro importante aqui é `params`, que define propriedades CSS que serão animadas. A maioria das propriedades podem ser animadas simultaneamente:

```
animate({width: "70%", opacity: 0.4, marginLeft:  
"0.6in", fontSize: "3em"});
```

Exemplo 5

Manipulação de conteúdo HTML

- jQuery contém funções para manipular elementos HTML e seus atributos.

- **Mudando o conteúdo:**

`$(selector).html(content)`

– O método `html()` muda o conteúdo de elementos HTML.

– Exemplo: `$("#p").html("TES-04");`

Manipulação de conteúdo HTML

- **Adicionando conteúdo**

- **`$(selector).append(content)`**

- Adiciona conteúdo ao final de elementos HTML.

- **`$(selector).prepend(content)`**

- Adiciona conteúdo no início de elementos HTML.

- **`$(selector).before(content)`**

- Adiciona conteúdo antes do(s) elemento(s) HTML.

- **`$(selector).after(content)`**

- Adiciona conteúdo depois do(s) elemento(s) HTML.

Exemplo 6

Manipulação de estilos

- O método `css()` é usado para manipulação de estilos.
 - `css(name)` - Retorna o valor de uma propriedade
 - `css(name,value)` - Seta o valor de uma propriedade
 - `css({properties})` - Seta múltiplos valores e propriedades
 - Exemplo:

```
$ ("p") .css ( { "color" : "red" , "font-size" : "200%" } ) ;
```
- Os métodos `height()` e `width()` são usados para mudar o tamanho de elementos.
 - Exemplo:

```
$ ("#div2") .width ("300px") ;
```

Exemplo 7

AJAX

- jQuery também provê métodos para desenvolvimento AJAX (*Asynchronous JavaScript and XML*). É possível fazer solicitação de dados TXT, HTML, XML ou JSON de um servidor remoto usando os métodos GET e POST do HTML. É possível ler dados remotos diretamente para um elemento HTML.

`$(selector).load(url,data,callback)`

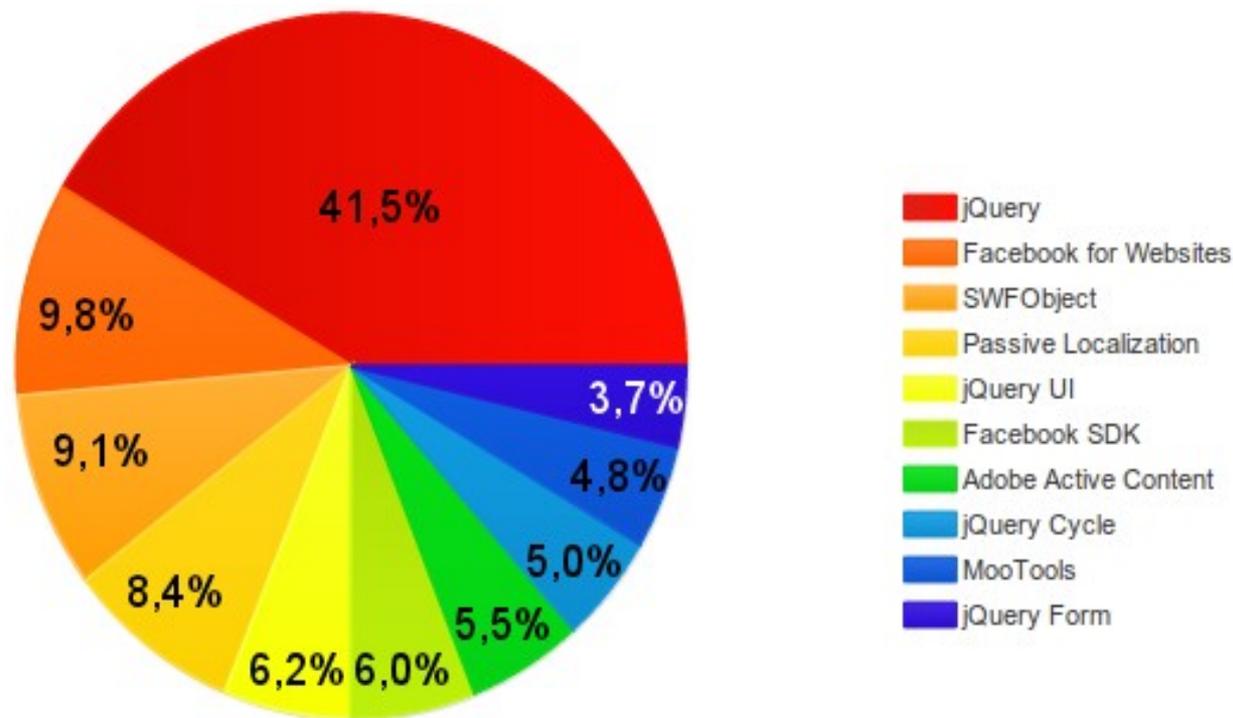
- selector define o(s) elemento(s) HTML a ser alterado e o parâmetro url especifica o endereço web da informação.

`$.ajax(options)`

- ajax() é uma função de mais baixo nível. O parâmetro option define pares de nomes/valores definindo url, senha, tipos de dados, filtros, conjunto de caracteres, tempo limite e funções de erro.

O que está sendo usado?

jQuery	51.54%
Facebook for Websites	21.18%
Facebook SDK	11.85%
SWFObject	10.86%
jQuery UI	10.69%
Twitter Platform	6.63%
jQuery Cookie	4.7%
Yahoo User Interface	4.44%
jQuery Cycle	4.43%
Google JS Api	3.99%
Adobe Active Content	3.74%
Passive Localization	3.51%
Prototype	3.39%
PNG Fix	3.03%
jQuery Easing	2.98%
script.aculo.us	2.91%
jQuery Tools	2.87%
Fancybox	2.85%
JCarousel	2.84%
Hover Intent	2.84%
jQuery Form	2.32%
MooTools	2.22%
HTML 5 Shiv	2.18%
jQuery Validate	1.94%
cufón	1.86%



1.000.000 "top" sites

10.000 "top" sites

Fonte: BuiltWith

jQuip

"90% do jQuery, 13% do tamanho"

- Principal objetivo é mostrar aos desenvolvedores do jQuery que uma reorganização modular é melhor, dizendo "nós provamos que as partes mais úteis da jQuery são apenas uma fração do seu código fonte".
- O arquivo jquip.js compactado (gzip) é de apenas 4,28 KB (19,8 KB minimizado sem compressão).
- A página <http://www.servicestack.net/jqbuilder/> permite aos usuários selecionar o plugin e criar uma versão customizada (com apenas as funções utilizadas).
- jQuip disponibilizado sobre licença MIT e pode ser baixado no seu repositório github: <https://github.com/mythz/jquip>.

- Crie um comando que exiba uma mensagem, avisando ao usuário que ele clicou em um link.
- Crie um comando que, quando o usuário clicar em um parágrafo (defina uma classe ou id para ele), o navegador adicione uma borda azul, com 4 pixels de espessura e que seja tracejada.
- Construa um documento que contenha uma lista ordenada(), e, toda vez que o usuário passar o mouse sobre a lista, ela receba um fundo vermelho.
- Crie um documento, com dois parágrafos, onde haja dois botões; um que adicione uma borda preta ao redor do parágrafo e outro que remova essa borda.