



jQuery

—

Prof. Victor Farias

v 1.0

Referências

w3schools.com

<http://api.jquery.com/>

Introdução

jQuery

- Biblioteca JavaScript
- Objetivo é facilitar o JavaScript
- Mais sucinto que JS puro
- Funcionalidades:
 - Manipulação HTML DOM
 - Manipulação CSS
 - Métodos para eventos HTML
 - AJAX
 - Utilidades

Como Importar?

- Local:

```
<head>
```

```
  <script src="jquery-3.1.1.min.js"></script>
```

```
</head>
```

- CDN

```
<head>
```

```
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

```
</head>
```

Sintaxe

- Sintaxe básica é: **\$(selector).action()**
 - Símbolo **\$** define acesso ao jQuery
 - **selector** é um seletor CSS para recuperar elementos HTML
 - **action** é uma ação jQuery sobre os elementos recuperados

`$(this).hide()` - hides the current element.

`$("p").hide()` - hides all `<p>` elements.

`$(".test").hide()` - hides all elements with `class="test"`.

`$("#test").hide()` - hides the element with `id="test"`.

Documento Pronto

- Para prevenir que algum código seja executado antes da página ser renderizada, podemos usar:

```
$(document).ready(function(){
```

```
    // jQuery methods go here...
```

```
});
```

- Ou podemos usar atalho

```
$(function(){
```

```
    // jQuery methods go here...
```

```
});
```

Eventos jQuery

- Executa ações a partir de eventos HTML

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Eventos jQuery

Exemplos

```
$("#p").click(function(){  
    // action goes here!!  
});
```

```
$("#p").dblclick(function(){  
    $(this).hide();  
});
```

Para conhecer
mais sobre
eventos

<http://api.jquery.com/category/events/>

Efeitos jQuery

Hide/Show

- É possível esconder e mostrar elementos usando métodos **hide** e **show**

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

- É possível especificar a velocidade da transição
 - `$(selector).hide(speed, callback);`
 - Parâmetro **speed** recebe "slow", "fast", or milliseconds
 - Parâmetro **callback** recebe função que é executada ao final do efeito

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

Hide/Show

- Ou, ainda, é possível mudar os estados de **show** para **hide** ou de **hide** para **show**

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

- Obs: **hide** e **show** manipulam o atributo CSS display para mostrar e esconder elemento

Fade

- `$(selector).fadeIn(speed,callback);`
 - Aplica fade em elementos escondidos
 - Parâmetro **speed** recebe "slow", "fast", or milliseconds
 - Parâmetro **callback** recebe função que é executada ao final do efeito

```
$("#button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});
```

- `$(selector).fadeOut(speed,callback);`
 - Aplica fade em elementos visíveis
- `$(selector).fadeToggle(speed,callback);`
 - Aplica fadeOut em elemento visíveis
 - Aplica fadeIn em elementos escondidos

Sliding

- `$(selector).slideDown(speed,callback);`
 - Aplica slide em elementos escondidos
 - Parâmetro **speed** recebe "slow", "fast", or milliseconds
 - Parâmetro **callback** recebe função que é executada ao final do efeito

```
$("#flip").click(function(){  
    $("#panel").slideDown();  
});
```

- `$(selector).slideUp(speed,callback);`
 - Aplica fade em elementos visíveis
- `$(selector).slideToggle(speed,callback);`
 - Aplica slideDown em elemento visíveis
 - Aplica slideUp em elementos escondidos

Para conhecer
mais sobre
efeitos

<https://api.jquery.com/category/effects/>

HTML DOM c/ jQuery



Acessando Conteúdo

- jQuery oferece 3 métodos principais para acessar conteúdo dos elementos
 - `text()` - Define ou retorna o texto do elemento selecionado
 - `html()` - Define ou retorna HTML dentro de um elemento
 - `val()` - Define ou retorna o valor de um campo de formulário

```
$("#btn1").click(function(){  
    alert("Text: " + $("#test").text());  
});  
$("#btn2").click(function(){  
    alert("HTML: " + $("#test").html());  
});
```

- Método `attr()` serve para acessar atributos

```
$("#button").click(function(){  
    alert($("#w3s").attr("href"));  
});
```

Definindo Conteúdo

```
$("#btn1").click(function(){  
    $("#test1").text("Hello world!");  
});  
  
$("#btn2").click(function(){  
    $("#test2").html("<b>Hello world!</b>");  
});  
  
$("#btn3").click(function(){  
    $("#test3").val("Dolly Duck");  
});
```

Definindo Conteúdo

- Usando função callback
 - Função de callback recebe recebe dois parâmetros:
 - O índice do elemento na lista de elementos selecionados
 - O valor original do conteúdo

```
$("#btn1").click(function(){  
    $("#test1").text(function(i, origText){  
        return "Old text: " + origText + " New text: Hello world!"  
        (index: " + i + " ");  
    });  
});
```

Adicionando elementos/conteúdo

- Método **append()**
 - Adiciona conteúdo no final do HTML selecionado

```
$("#p").append("Some appended text.");
```

- Também é possível adicionar elementos HTML

```
function appendText() {  
  
    var txt1 = "<p>Text.</p>";           // Create element with HTML  
  
    var txt2 = $("#<p></p>").text("Text."); // Create with jQuery  
  
    var txt3 = document.createElement("p"); // Create with DOM  
  
    txt3.innerHTML = "Text.";   
  
    $("#body").append(txt1, txt2, txt3); // Append the new elements  
  
}
```

Adicionando elementos/conteúdo

- Método **prepend()**
 - Adiciona conteúdo no início do elemento

```
$("#p").prepend("Some prepended text.");
```

- Método **after()**
 - Adiciona conteúdo depois do elemento
- Método **before()**
 - Adiciona conteúdo antes do elemento

Removendo Elementos

- Método **remove()**
 - Remove o elemento selecionado

```
$("#div1").remove();
```

- Método **empty()**
 - Remove os filhos do elemento selecionado

```
$("#div1").empty();
```

Manipulando Classes

- Método **addClass()**

- Adiciona classes ao elemento selecionado

```
$("#button").click(function(){  
    $("#h1, h2, p").addClass("blue");  
    $("#div").addClass("important blue");  
});
```

- Método **removeClass()**

- Remove classes do elemento selecionado

```
$("#button").click(function(){  
    $("#h1, h2, p").removeClass("blue");  
});
```


Manipulando Classes

- Método **toggleClass()**
 - Alterna classe do elemento

```
$("#button").click(function(){  
    $("#h1, h2, p").toggleClass("blue");  
});
```

Manipulando CSS

- Método **css()**
 - Define ou retorna um ou mais propriedades de estilo CSS

Acessando:

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

Definindo:

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

Definindo várias propriedades

```
$("#p").css({"background-color": "yellow", "font-size": "200%"});
```

Para conhecer
mais sobre DOM

<http://api.jquery.com/category/manipulation/>

AJAX c/ jQuery

AJAX

- Método **ajax()** para fazer requisições HTTP
 - Recebe um **objeto** como parâmetro
 - Esse objeto contém uma coleção de pares chave/valor especificando os parâmetros da requisição HTTP
 - Sintaxe: \$.ajax({name:value, name:value, ... })

AJAX

Exemplo GET:

```
$.ajax({  
  type: 'GET',  
  url: 'http://rest.learncode.academy/api/johnbob/friends',  
  success: function(data) {  
    console.log("I have friends!", data); //returns all of johnbob's friends  
  }  
});
```

- **type:** especifica o tipo da requisição
- **url:** especifica a url de destino da requisição
- **success:** recebe um função de callback que é executada quando a requisição termina com sucesso. Essa função recebe como parâmetro o dados contidos no corpo da resposta

AJAX

Exemplo POST:

```
$.ajax({  
  type: 'POST',  
  url: 'http://rest.learncode.academy/api/johnbob/friends',  
  data: {name: 'Billy Bob', age: 27},  
  success: function(data) {  
    console.log("Friend added!", data); //the new item is returned with an ID  
  }  
});
```

- **data:** define os dados enviados

Para conhecer
mais sobre AJAX
c/ jQuery

[http://api.jquery.com
/category/ajax/](http://api.jquery.com/category/ajax/)

Perguntas?

Prof. Victor Farias