Prof. Pedro Clarindo da Silva Neto

O que é?

O jQuery é uma biblioteca JavaScript leve, "escreva menos, faça mais".

O objetivo do jQuery é facilitar o uso do JavaScript no seu site.

—— W3C

O que é?

O jQuery pega muitas tarefas comuns que exigem muitas linhas de código JavaScript para ser executadas e as agrupa em métodos que você pode chamar com uma única linha de código.

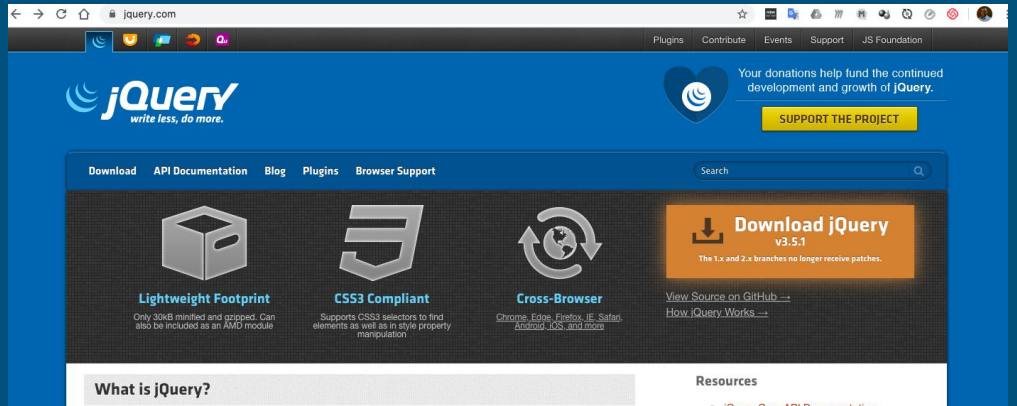
O jQuery também simplifica muitas coisas complicadas do JavaScript, como chamadas AJAX e manipulação de DOM.

A biblioteca jQuery contém os seguintes recursos:

- Manipulação HTML / DOM
- Manipulação de CSS
- Métodos de evento HTML
- Efeitos e animações
- AJAX
- Serviços de utilidade pública

Existem duas versões do jQuery disponíveis para download:

- Versão de produção - é para o seu site que está no ar porque ele foi minificado e compactado
- Versão de desenvolvimento - é para teste e desenvolvimento (código não compactado e legível)



Ambas as versões podem ser baixadas em jQuery.com

Usando JQuery

- A biblioteca jQuery é um único arquivo JavaScript, e você a referencia com a tag HTML <script> (observe que a tag <script> deve estar dentro da seção <head>):

```
<head>
<script src="jquery-3.5.1.min.js"></script>
</head>
```

Usando JQuery

- Se você não deseja fazer o download e hospedar o jQuery, pode incluí-lo em uma CDN (Content Delivery Network).
- O Google é um exemplo de alguém que hospeda o jQuery:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
```

Sintaxe

A sintaxe do jQuery é personalizada para selecionar elementos HTML e executar alguma ação no (s) elemento (s).

A sintaxe básica é: **$ (seletor). ação ()**

- Um sinal de $ para definir / acessar o jQuery
- Um (seletor) para "consultar (ou localizar)" elementos HTML
- Uma ação() jQuery a ser executada no (s) elemento (s)

Sintaxe

`$(this).hide()` - oculta o elemento atual.

`$("p").hide()` - oculta todos os elementos <p>.

`$(".test").hide()` - oculta todos os elementos com class = "test".

`$("#test").hide()` - oculta o elemento com id = "test".

Event Ready

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

  // jQuery methods go here...

});
```

Utilizar o *event ready* evita que qualquer código jQuery seja executado antes que o documento termine de carregar (esteja pronto).

É uma boa prática aguardar que o documento esteja totalmente carregado e pronto antes de trabalhar com ele. Isso também permite que você tenha seu código JavaScript antes do corpo do seu documento, na seção principal.

Event Ready

Aqui estão alguns exemplos de ações que podem falhar se os métodos forem executados antes do carregamento completo do documento:

- Tentando ocultar um elemento que ainda não foi criado
- Tentando obter o tamanho de uma imagem que ainda não foi carregada

Event Ready

A equipe do jQuery também criou um método ainda mais curto para o event ready do documento:

```
$(function(){

  // jQuery methods go here...

});
```

Seletores

O **seletor de elemento** jQuery seleciona elementos com base no nome do elemento.

Você pode selecionar todos os <p>elementos em uma página como esta:

```
$("p")
```

Seletores

Quando um usuário clica no botão, todos os elementos <p> ficam ocultos:

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click me to hide paragraphs</button>

</body>
</html>
```

Seletores

O seletor jQuery #id usa o atributo id de uma tag HTML para encontrar o elemento específico.

Um ID deve ser exclusivo dentro de uma página, portanto, você deve usar o seletor #id quando desejar encontrar um único elemento exclusivo.

Para localizar um elemento com um ID específico, escreva um caractere de hash (#) , seguido pelo ID do elemento HTML:

```
$("#test")
```

Seletores

Quando um usuário clica no botão, todos os elementos que tem o id `test` ficam ocultos:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>

<button>Click me</button>

</body>
</html>
```

Seletores

O seletor jQuery .class encontra elementos com uma classe específica.

Para encontrar elementos com uma classe específica, escreva um caractere de ponto, seguido pelo nome da classe:

```
$(".test")
```

Seletores

Quando um usuário clica no botão, todos os elementos que tem a classe `test` ficam ocultos:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(".test").hide();
  });
});
</script>
</head>
<body>

<h2 class="test">This is a heading</h2>

<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click me</button>

</body>
</html>
```

Seletores

Mais alguns seletores jQuery

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

Seletores

DICA: usar o jQuery Selector Tester da W3C para testar os vários seletores

https://www.w3schools.com/jquery/trysel.asp

Click a selector to see which element(s) gets selected in the result:

```
$("#Lastname")
$(".intro")
$(".intro, #Lastname")
$("h1")
$("h1, p")
$("p:first")
$("p:last")
$("tr:even")
$("tr:odd")
$("p:first-child")
$("p:first-of-type")
$("p:last-child")
$("p:last-of-type")
$("li:nth-child(1)")
$("li:nth-last-child(1)")
$("li:nth-of-type(2)")
$("li:nth-last-of-type(2)")
$("b:only-child")
$("h3:only-of-type")
$("div > p")
$("div p")
$("ul + p")
$("ul ~ table")
$("ul li:eq(0)")
$("ul li:gt(0)")
$("ul li:lt(2)")
$(":header")
$(":header:not(h1)")
$(":animated")
$(":focus")
$(":contains(Duck)")
$("div:has(p)")
$(":empty")
$(":parent")
$("p:hidden")
$("table:visible")
$(":root")
```

Result:

<h1>**Welcome to My Homepage**</h1>

My name is Donald Duck

I live in Duckburg

I have many friends:

<ul id="Listfriends>

- Goofy
- Mickey
- Daisy
- Pluto

I really like Daisy!!

Ciao bella

<h3>**We are all animals!**</h3>

 My latest discoveries has led me to believe that we are all animals:

<table>

Name	Type of Animal
Mickey	Mouse
Goofey	Dog
Daisy	Duck
Pluto	Dog

</table>

Seletores

DICA: Para uma referência completa de todos os seletores de jQuery, acesse a Referência de seletores de jQuery .

https://www.w3schools.com/jquery/jquery_selectors.asp

Selector	Example	Selects
*	$("*")	All elements
#id	$("#lastname")	The element with id="lastname"
.class	$(".intro")	All elements with class="intro"
.class,.class	$(".intro,.demo")	All elements with the class "intro" or "demo"
element	$("p")	All <p> elements
el1,el2,el3	$("h1,div,p")	All <h1>, and <p> elements
:first	$("p:first")	The first <p> element
:last	$("p:last")	The last <p> element
:even	$("tr:even")	All even <tr> elements
:odd	$("tr:odd")	All odd <tr> elements
:first-child	$("p:first-child")	All <p> elements that are the first child of their parent
:first-of-type	$("p:first-of-type")	All <p> elements that are the first <p> element of their parent
:last-child	$("p:last-child")	All <p> elements that are the last child of their parent
:last-of-type	$("p:last-of-type")	All <p> elements that are the last <p> element of their parent
:nth-child(n)	$("p:nth-child(2)")	All <p> elements that are the 2nd child of their parent
:nth-last-child(n)	$("p:nth-last-child(2)")	All <p> elements that are the 2nd child of their parent, counting from the last child
:nth-of-type(n)	$("p:nth-of-type(2)")	All <p> elements that are the 2nd <p> element of their parent
:nth-last-of-type(n)	$("p:nth-last-of-type(2)")	All <p> elements that are the 2nd <p> element of their parent, counting from the last child
:only-child	$("p:only-child")	All <p> elements that are the only child of their parent
:only-of-type	$("p:only-of-type")	All <p> elements that are the only child, of its type, of their parent
parent > child	$("div > p")	All <p> elements that are a direct child of a element
parent descendant	$("div p")	All <p> elements that are descendants of a element
element + next	$("div + p")	The <p> element that are next to each elements
element ~ siblings	$("div ~ p")	All <p> elements that appear after the element
:eq(index)	$("ul li:eq(3)")	The fourth element in a list (index starts at 0)
:gt(no)	$("ul li:gt(3)")	List elements with an index greater than 3
:lt(no)	$("ul li:lt(3)")	List elements with an index less than 3

Prof. Pedro Clarindo da Silva Neto

Funções em um arquivo separado

Se o seu site tiver muitas páginas e você desejar que as funções do jQuery sejam fáceis de dar manutenção, você poderá colocar as funções do jQuery em um arquivo .js separado.

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="my_jquery_functions.js"></script>
</head>
```

Métodos de evento jQuery

O jQuery é feito sob medida para responder a eventos em uma página HTML.

Todas as ações às quais uma página da Web pode responder são chamadas de eventos. Um evento representa o momento preciso em que algo acontece.

Exemplos:

- movendo o mouse sobre um elemento
- selecionando um botão de opção
- clicando em um elemento

Métodos de evento jQuery

Aqui estão alguns eventos DOM comuns:

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

Sintaxe do jQuery para métodos de evento

No jQuery, a maioria dos eventos DOM possui um método jQuery equivalente.

Para atribuir um evento de clique a todos os parágrafos em uma página, você pode fazer o seguinte:

```
$("p").click();
```

O próximo passo é definir o que deve acontecer quando o evento for disparado. Você deve passar uma função para o evento:

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

Métodos de evento jQuery comumente usados

O método click() anexa uma função de manipulador de eventos a um elemento HTML.

A função é executada quando o usuário clica no elemento HTML.

O exemplo a seguir diz:

Quando um evento click é acionado em um <p>elemento; oculte o <p>elemento atual :

Método click()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

Método dblclick()

O método dblclick() anexa uma função de manipulador de eventos a um elemento HTML.

A função é executada quando o usuário clica duas vezes no elemento HTML:

Método dblclick()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you double-click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

Método focus()

O método focus() anexa uma função de manipulador de eventos a um campo de formulário HTML.

A função é executada quando o campo do formulário obtém o foco:

Método focus()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("input").focus(function(){
    $(this).css("background-color", "yellow");
  });
  $("input").blur(function(){
    $(this).css("background-color", "green");
  });
});
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

Método blur()

O método blur() anexa uma função de manipulador de eventos a um campo de formulário HTML.

A função é executada quando o campo do formulário perde o foco:

Método blur()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("input").focus(function(){
    $(this).css("background-color", "yellow");
  });
  $("input").blur(function(){
    $(this).css("background-color", "green");
  });
});
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

Método on()

O método on() anexa um ou mais manipuladores de eventos para os elementos selecionados.

Anexe um evento de clique a um elemento <p>:

Método on()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on("click", function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

Método on()

click()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googl
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will d
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

on("click")

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on("click", function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

Método on()

Observe o exemplo com vários manipuladores de eventos anexados no método on() em um elemento <p>:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
      $(this).css("background-color", "lightblue");
    },
    click: function(){
      $(this).css("background-color", "yellow");
    }
  });
});
</script>
</head>
<body>

<p>Click or move the mouse pointer over this paragraph.</p>

</body>
</html>
```

Efeitos jQuery

Prof. Pedro Clarindo da Silva Neto

Efeitos jQuery - Ocultar e mostrar

Com o jQuery, você pode ocultar e mostrar elementos HTML com os métodos hide()e show():

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>

<p>If you click on the "Hide" button, I will disappear.</p>

<button id="hide">Hide</button>
<button id="show">Show</button>

</body>
</html>
```

Efeitos jQuery - Ocultar e mostrar

Sintaxe:

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

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

O parâmetro opcional speed especifica a velocidade da ocultação / exibição e pode assumir os seguintes valores: "slow", "fast" ou milissegundos.

O parâmetro opcional de retorno de chamada (callback) é uma função a ser executada após a conclusão do método hide()ou show().

Efeitos jQuery - Ocultar e mostrar

O exemplo a seguir demonstra o parâmetro de velocidade com hide():

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>

</body>
</html>
```

jQuery toggle()

Você também pode alternar entre ocultar e mostrar um elemento com o método toggle()

Os elementos mostrados são ocultos e os elementos ocultos são mostrados:

jQuery toggle()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").toggle();
  });
});
</script>
</head>
<body>

<button>Toggle between hiding and showing the paragraphs</button>

<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>

</body>
</html>
```

jQuery toggle()

Sintaxe:

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

O parâmetro de velocidade opcional pode assumir os seguintes valores: "slow", "fast" ou milissegundos.

O parâmetro opcional de retorno de chamada é uma função a ser executada após a conclusão do toggle().

Métodos Fading jQuery

Com o jQuery, você pode desvanecer (fading) e diminuir a visibilidade de um elemento.

O jQuery possui os seguintes métodos de desvanecimento:

- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()

Método jQuery fadeIn ()

O método jQuery fadeIn() é usado para fazer surgir um elemento oculto.

Sintaxe:

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

O exemplo a seguir demonstra o método fadeIn() com diferentes parâmetros:

Método jQuery fadeIn ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button>Click to fade in boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-
color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-
color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-
color:blue;"></div>

</body>
</html>
```

Método jQuery fadeOut ()

O método jQuery fadeOut() é usado para esmaecer um elemento visível.

Sintaxe:

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

O exemplo a seguir demonstra o fadeOut()método com diferentes parâmetros:

Método jQuery fadeOut ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeOut() with different parameters.</p>

<button>Click to fade out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;">
</div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;">
</div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

Método jQuery fadeToggle ()

O método jQuery fadeToggle() alterna entre os métodos fadeIn()e fadeOut().

Sintaxe:

```
$(selector).fadeToggle(speed,callback);
```

O exemplo a seguir demonstra o método fadeToggle() com diferentes parâmetros:

Método jQuery fadeToggle ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

<button>Click to fade in/out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div>
<br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

Método jQuery fadeTo ()

O método jQuery fadeTo() permite desbotar até uma determinada opacidade (valor entre 0 e 1).

Sintaxe:

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

O parâmetro de opacidade necessário no método fadeTo() especifica o desbotamento para uma determinada opacidade (valor entre 0 e 1).

Método jQuery fadeTo ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeTo("slow", 0.15);
    $("#div2").fadeTo("slow", 0.4);
    $("#div3").fadeTo("slow", 0.7);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeTo() with different parameters.</p>

<button>Click to fade boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;">
</div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;">
</div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

O exemplo a seguir demonstra o método fadeTo()com diferentes parâmetros:

Efeitos jQuery - Deslizando (Sliding)

Com o jQuery, você pode criar um efeito deslizante nos elementos.

O jQuery possui os seguintes métodos deslizantes:

- slideDown()
- slideUp()
- slideToggle()

Método jQuery slideDown (), slideUp()

O método jQuery slideDown() é usado para deslizar um elemento para baixo.

Sintaxe:

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

O método jQuery slideUp() é usado para deslizar um elemento para cima.

Sintaxe:

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

Método jQuery slideDown ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

Método jQuery slideUp()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideUp("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
}
</style>
</head>
<body>

<div id="flip">Click to slide up panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

Método jQuery slideToggle ()

O método jQuery slideToggle() alterna entre os métodos slideDown()e slideUp().

Sintaxe:

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

O exemplo a seguir demonstra o método slideToggle():

Método jQuery slideToggle ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideToggle("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

Efeitos jQuery - método animate ()

O método jQuery animate() é usado para criar animações personalizadas.

Sintaxe:

```
$(selector).animate({params},speed,callback);
```

O parâmetro params necessário define as propriedades CSS a serem animadas.

O exemplo a seguir demonstra um uso simples do método; animate() move um elemento para a direita, até atingir uma propriedade esquerda de 250px:

Efeitos jQuery - método animate ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({left: '250px'});
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>Por padrão, todos os elementos HTML têm uma posição estática e não podem ser
movidos. Para manipular a posição, lembre-se de definir primeiro a propriedade de
posição CSS do elemento como relativa, fixa ou absoluta!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

Efeitos jQuery - método animate ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left: '250px',
      opacity: '0.5',
      height: '150px',
      width: '150px'
    });
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>Por padrão, todos os elementos HTML têm uma posição estática e não podem ser
movidos. Para manipular a posição, lembre-se de definir primeiro a propriedade de
posição CSS do elemento como relativa, fixa ou absoluta!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

Observe que várias propriedades podem ser animadas ao mesmo tempo:

Método animate () - filas

Por padrão, o jQuery vem com funcionalidade de fila para animações.

Isso significa que, se você escrever várias chamadas animate() uma após a outra, o jQuery criará uma fila "interna" com essas chamadas de método. Em seguida, ele executa as chamadas animadas uma a uma.

Portanto, se você deseja executar animações diferentes, aproveitamos a funcionalidade da fila:

Método animate () - filas

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var div = $("div");
    div.animate({height: '300px', opacity: '0.4'}, "slow");
    div.animate({width: '300px', opacity: '0.8'}, "slow");
    div.animate({height: '100px', opacity: '0.4'}, "slow");
    div.animate({width: '100px', opacity: '0.8'}, "slow");
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>
```

Método animate () - filas

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var div = $("div");
    div.animate({left: '100px'}, "slow");
    div.animate({fontSize: '3em'}, "slow");
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:200px;
position:absolute;">HELLO</div>

</body>
</html>
```

O exemplo abaixo move primeiro o elemento para a direita e depois aumenta o tamanho da fonte do texto:

Funções jQuery de retorno de chamada

Instruções JavaScript são executadas linha por linha. No entanto, com efeitos, a próxima linha de código pode ser executada mesmo que o efeito não esteja concluído. Isso pode criar erros.

Para evitar isso, você pode criar uma função de retorno de chamada (callback).

Uma função de retorno de chamada é executada após a conclusão do efeito atual.

Sintaxe típica: **$(seletor).hide (velocidade, retorno de chamada);**

Funções jQuery de retorno de chamada

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide("slow", function(){
      alert("The paragraph is now hidden");
    });
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>
```

O exemplo possui um parâmetro de retorno de chamada que é uma função que será executada após a conclusão do efeito hide:

Funções jQuery de retorno de chamada

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
    alert("The paragraph is now hidden");
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>
```

O exemplo abaixo **não possui parâmetro de retorno de chamada** e a caixa de alerta será exibida antes que o efeito ocultar seja concluído:

Encadeamento jQuery (Chaining)

Com o jQuery, você pode encadear ações / métodos.

O encadeamento nos permite executar vários métodos jQuery (no mesmo elemento) em uma única instrução.

Para encadear uma ação, basta anexá-la à ação anterior.

Os seguintes exemplos encadeiam em conjunto os métodos css(), slideUp()e slideDown(). O elemento "p1" primeiramente muda para vermelho, depois desliza para cima e depois para baixo:

Encadeamento jQuery (Chaining)

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "red").slideUp(2000).slideDown(2000);
  });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>
</html>
```

Encadeamento jQuery (Chaining)

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "red")
      .slideUp(2000)
      .slideDown(2000);
  });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>
</html>
```

Ao encadear, a linha de código pode se tornar bastante longa. No entanto, o jQuery não é muito rigoroso na sintaxe; você pode formatá-lo como quiser, incluindo quebras de linha e recuos.

jQuery HTML

Prof. Pedro Clarindo da Silva Neto

Manipulação DOM do jQuery

Uma parte muito importante do jQuery é a possibilidade de manipular o DOM.

O jQuery vem com vários métodos relacionados ao DOM que facilitam o acesso e a manipulação de elementos e atributos.

Três métodos jQuery simples, mas úteis, para manipulação do DOM são:

- text() - Define ou retorna o conteúdo do texto dos elementos selecionados
- html() - Define ou retorna o conteúdo dos elementos selecionados (incluindo marcação HTML)
- val() - Define ou retorna o valor dos campos do formulário

Manipulação DOM do jQuery

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>

<p id="test">This is some <b>bold</b> text in a paragraph.</p>

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>

</body>
</html>
```

O exemplo demonstra como obter conteúdo com o métodos jQuery text() e html():

Manipulação DOM do jQuery

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert("Value: " + $("#test").val());
  });
});
</script>
</head>
<body>

<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>

<button>Show Value</button>

</body>
</html>
```

O exemplo demonstra como obter o valor de um campo de entrada com o método jQuery val() :

Obter atributos - attr ()

O método jQuery attr() é usado para obter valores de atributo.

O exemplo a seguir demonstra como obter o valor do atributo href em um link:

Obter atributos - attr ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert($("#w3s").attr("href"));
  });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" id="w3s">W3Schools.com</a></p>

<button>Show href Value</button>

</body>
</html>
```

Definir conteúdo - text(), html() e val()

Utiliza-se os mesmos três métodos para definir o conteúdo :

- text() - Define ou retorna o conteúdo do texto dos elementos selecionados
- html() - Define ou retorna o conteúdo dos elementos selecionados (incluindo marcação HTML)
- val() - Define ou retorna o valor dos campos do formulário

Definir conteúdo - text(), html() e val()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");
  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
  });
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
  });
});
</script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>
```

O exemplo demonstra como definir o conteúdo com os métodos jQuery text(), html()e val():

Definir atributos - attr ()

O método jQuery attr() também é usado para definir/alterar valores de atributo.

O exemplo a seguir demonstra como alterar (definir) o valor do atributo href em um link:

Definir atributos - attr ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
  });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" id="w3s">W3Schools.com</a></p>

<button>Change href Value</button>

<p>Mouse over the link (or click on it) to see that the value of the href
attribute has changed.</p>

</body>
</html>
```

Definir atributos - attr ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#w3s").attr({
      "href" : "https://www.w3schools.com/jquery/",
      "title" : "W3Schools jQuery Tutorial"
    });
  });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" title="some title"
id="w3s">W3Schools.com</a></p>

<button>Change href and title</button>

<p>Mouse over the link to see that the href attribute has changed and a title
attribute is set.</p>

</body>
</html>
```

O método attr() também permite definir vários atributos ao mesmo tempo.O exemplo a seguir demonstra como definir os atributos href e title ao mesmo tempo:

Adicionar novo conteúdo HTML

São quatro métodos jQuery usados para adicionar novo conteúdo:

- append() - Insere conteúdo no final dos elementos selecionados
- prepend() - Insere o conteúdo no início dos elementos selecionados
- after() - Insere conteúdo após os elementos selecionados
- before() - Insere conteúdo antes dos elementos selecionados

Método jQuery append()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").append(" <b>Appended text</b>.");
  });

  $("#btn2").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>

</body>
</html>
```

O método jQuery append() insere o conteúdo NO FINAL dos elementos HTML selecionados.

Método jQuery prepend()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").prepend("<b>Prepended text</b>. ");
  });
  $("#btn2").click(function(){
    $("ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Prepend text</button>
<button id="btn2">Prepend list item</button>

</body>
</html>
```

O método jQuery prepend() insere o conteúdo NO INÍCIO dos elementos HTML selecionados.

Adicionar vários novos elementos com append() e prepend()

Nos dois exemplos anteriores, apenas inserimos algum texto/HTML no início/fim dos elementos HTML selecionados.

No entanto, os métodos append() e prepend() podem ter um número infinito de novos elementos como parâmetros. Os novos elementos podem ser gerados com texto/HTML, com jQuery ou com código JavaScript e elementos DOM.

Adicionar vários novos elementos com append() e prepend()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
function appendText() {
  var txt1 = "<p>Text.</p>";         // Criar texto com HTML
  var txt2 = $("<p></p>").text("Text.");  // Criar texto com jQuery
  var txt3 = document.createElement("p");
  txt3.innerHTML = "Text.";          // Criar texto com DOM
  $("body").append(txt1, txt2, txt3);    // Append new elements
}
</script>
</head>
<body>

<p>This is a paragraph.</p>
<button onclick="appendText()">Append text</button>

</body>
</html>
```

Os elementos são criados com HTML, jQuery e JavaScript/DOM. Em seguida, anexamos os novos elementos ao texto com o método append().

Métodos jQuery after () e before ()

O método jQuery after() insere o conteúdo APÓS os elementos HTML selecionados.

O método jQuery before() insere o conteúdo ANTES dos elementos HTML selecionados.

Métodos jQuery after () e before ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("img").before("<b>Before</b>");
  });

  $("#btn2").click(function(){
    $("img").after("<i>After</i>");
  });
});
</script>
</head>
<body>

<img src="/images/w3jquery.gif" alt="jQuery" width="100" height="140"><br><br>

<button id="btn1">Insert before</button>
<button id="btn2">Insert after</button>

</body>
</html>
```

Adicionar vários novos elementos com after() e before()

Além disso, os métodos after()e before()podem ter um número infinito de novos elementos como parâmetros. Os novos elementos podem ser gerados com texto / HTML (como no exemplo anterior), com jQuery ou com código JavaScript e elementos DOM.

Adicionar vários novos elementos com after() e before()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
function afterText() {
  var txt1 = "<b>I </b>";           // Create element with HTML
  var txt2 = $("<i></i>").text("love ");  // Create with jQuery
  var txt3 = document.createElement("b");   // Create with DOM
  txt3.innerHTML = "jQuery!";
  $("img").after(txt1, txt2, txt3);     // Insert new elements after img
}
</script>
</head>
<body>

<img src="/images/w3jquery.gif" alt="jQuery" width="100" height="140">

<p>Click the button to insert text after the image.</p>

<button onclick="afterText()">Insert after</button>

</body>
</html>
```

Os elementos são criados com HTML, jQuery e JavaScript/DOM. Em seguida, anexamos os novos elementos ao texto com o método after().

Remover elementos / conteúdo

Para remover elementos e conteúdo, existem principalmente dois métodos jQuery:

- remove() - Remove o elemento selecionado (e seus elementos filhos)
- empty() - Remove os elementos filhos do elemento selecionado

Método jQuery remove ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").remove();
  });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid
black;background-color:yellow;">

This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>

</div>
<br>

<button>Remove div element</button>

</body>
</html>
```

O método jQuery remove() remove os elementos selecionados e seus elementos filhos.

Método jQuery empty()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").empty();
  });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid
black;background-color:yellow;">

This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>

</div>
<br>

<button>Empty the div element</button>

</body>
</html>
```

O método jQuery empty() remove os elementos filhos dos elementos selecionados.

Filtrar os elementos a serem removidos

O método jQuery remove() também aceita um parâmetro, que permite filtrar os elementos a serem removidos.

O parâmetro pode ser qualquer uma das sintaxes do seletor jQuery.

Filtrar os elementos a serem removidos

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").remove(".test");
  });
});
</script>
<style>
.test {
  color: red;
  font-size: 20px;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p class="test">This is another paragraph.</p>
<p class="test">This is another paragraph.</p>

<button>Remove all p elements with class="test"</button>

</body>
</html>
```

O exemplo a seguir remove todos os elementos <p> com class="test":

jQuery Manipulando CSS

O jQuery possui vários métodos para manipulação de CSS. Veja os seguintes métodos:

- addClass() - Adiciona uma ou mais classes aos elementos selecionados
- removeClass() - Remove uma ou mais classes dos elementos selecionados
- toggleClass() - Alterna entre adicionar/remover classes dos elementos selecionados
- css() - Define ou retorna o atributo style

Método jQuery addClass ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").addClass("blue");
    $("div").addClass("important");
  });
});
</script>
<style>
.important {
  font-weight: bold;
  font-size: xx-large;
}

.blue {
  color: blue;
}
</style>
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<div>This is some important text!</div><br>

<button>Add classes to elements</button>

</body>
</html>
```

O exemplo a seguir mostra como adicionar atributos de classe a diferentes elementos.

Método jQuery addClass ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").addClass("important blue");
  });
});
</script>
<style>
.important {
  font-weight: bold;
  font-size: xx-large;
}

.blue {
  color: blue;
}
</style>
</head>
<body>

<div id="div1">This is some text.</div>
<div id="div2">This is some text.</div>
<br>

<button>Add classes to first div element</button>

</body>
</html>
```

É possível também especificar várias classes dentro do método addClass():

Método jQuery removeClass ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").removeClass("blue");
  });
});
</script>
<style>
.blue {
  color: blue;
}
</style>
</head>
<body>

<h1 class="blue">Heading 1</h1>
<h2 class="blue">Heading 2</h2>

<p class="blue">This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Remove class from elements</button>

</body>
</html>
```

O exemplo a seguir mostra como remover um atributo de classe específico de diferentes elementos:

Método jQuery toggleClass ()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").toggleClass("blue");
  });
});
</script>
<style>
.blue {
  color: blue;
}
</style>
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Toggle class</button>

</body>
</html>
```

Este método alterna entre adicionar / remover classes dos elementos selecionados:

Retornar uma propriedade CSS

Para retornar o valor de uma propriedade CSS especificada, use a seguinte sintaxe:

```
css("propertyname");
```

Retornar uma propriedade CSS

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert("Background color = " + $("p").css("background-color"));
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<button>Return background-color of p</button>

</body>
</html>
```

O exemplo a seguir retornará o valor da cor de plano de fundo do PRIMEIRO elemento correspondente:

Definir uma propriedade CSS

Para definir uma propriedade CSS especificada, use a seguinte sintaxe:

```
css("propertyname","value");
```

Definir uma propriedade CSS

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css("background-color", "yellow");
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set background-color of p</button>

</body>
</html>
```

O exemplo a seguir definirá o valor da cor de plano de fundo para TODOS os elementos correspondentes:

Definir várias propriedades CSS

Para definir várias propriedades CSS, use a seguinte sintaxe:

```
css({"propertyname":"value","propertyname":"value",...});
```

Definir várias propriedades CSS

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css({"background-color": "yellow", "font-size": "200%"});
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set multiple styles for p</button>

</body>
</html>
```

O exemplo a seguir definirá uma cor de plano de fundo e um tamanho de fonte para TODOS os elementos correspondentes:

Referências

https://www.w3schools.com/jquery/jquery_intro.asp