



APRENDENDO A UTILIZAR O JQUERY

Resumo:

Esse material tem como objetivo apresentar um framework Javascript, o JQuery. Abordando a definição, o contexto histórico, seletores, métodos, eventos e o JQuery UI.

Yorras Gomes de Lima

yorrasgomeslima@gmail.com

Sumário

1. O que é o JQuery?	3
2. Contexto histórico	4
3. DOM	6
4. Inclusão do JQuery	7
4.1 Criando um projeto no NetBeans	7
4.2 CDN..	8
4.3 Download...	9
5. Sintaxe	11
6. Seletores	12
7. Métodos	17
8. Métodos de eventos	19
9. Eventos	20
10. AJAX	21
10.1 Estado de prontidão	21
10.2 Código de status	21
10.3 Método ajax do JQuery	22
11. JQuery UI	23
11.1 Elementos de interação com o mouse	23
11.2 Widgets	27
11.3 Efeitos	40

1. O que é o JQuery?

O JQuery é uma biblioteca Javascript de código aberto que foi desenvolvida para simplificar e diminuir a quantidade de código Javascript. O JQuery é capaz de adicionar interatividade e dinamismo às páginas web com objetivo de fazer isso de forma simplificada, o que é aderente ao seu lema que é Write less, do more, ou seja, escreva menos, faça mais. A biblioteca foi criada dentro dos Padrões Web estipulados pela W3C, por este motivo é uma biblioteca multi-plataforma, ou seja, é compatível com qualquer navegador de internet.

O JQuery inclui recursos e simplifica a manipulação de:

- Documentos HTML;
- Propriedades CSS;
- DOM;
- Seletores;
- Eventos;
- Efeitos e animações;
- Interações com AJAX.

Quanto ao tipo de licença, o JQuery possui licença dual:

- **Licença MIT:** A licença MIT permite que o software seja tratado sem restrições para o uso, modificação e distribuição. Desta forma, pode ser utilizada tanto em projetos de software livre, quanto em projeto de software proprietário. No texto desta licença não existe copyright, desta forma outros grupos podem modificar a licença, com o objetivo de atender as suas necessidades.
- **GNU General Public License versão 2:** A GNU General Public License ou GPL é a licença para software livre com maior utilização. Ela se baseia em 4 liberdades: (1) a liberdade de executar o programa, para qualquer propósito; (2) a liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades; (3) a liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo; e (4) a liberdade de aperfeiçoar o programa, e liberar suas modificações, de modo que toda a comunidade se beneficie delas. O acesso ao código-fonte é um pré-requisito para seus objetivos serem atingidos. Com a garantia destas liberdades, a GPL permite que os programas sejam distribuídos e reaproveitados, mantendo, porém, os direitos do autor.

2. Contexto histórico

Era o ano de 2006, e John Resig era um desenvolvedor web que trabalhava em seus próprios projetos. Ele estava frustrado com a dificuldade de escrever JavaScript que funcionasse em vários navegadores, e decidiu escrever sua própria biblioteca Javascript para resolver o problema. Essa biblioteca seria chamada de JQuery.

Em 16 de janeiro de 2006, John Resig deu uma palestra sobre sua nova biblioteca em uma BarCamp local, para um pequeno grupo de desenvolvedores web, e publicou isso em seu blog:

Finalmente, anunciei meu segundo novo lançamento da noite: **jQuery: a nova geração do Javascript**. Em poucas palavras, este código revoluciona a maneira como você pode fazer o Javascript interagir com HTML – é um conjunto de códigos realmente impressionante, e dediquei muito tempo e esforços para deixá-lo como imaginei. Estou trabalhando na documentação para o site neste momento, que deverá ficar pronta daqui a dois dias.

A página web do JQuery, na época, era desse jeito:



 **jQuery**
Nova geração do Javascript

O jQuery é um novo tipo de biblioteca Javascript. Não se trata de uma estrutura enorme e abarrotada que promete o melhor em AJAX, nem apenas um conjunto de melhorias complexas e desnecessárias. **O jQuery foi desenvolvido para mudar a maneira com que você escreve Javascript.**

Novidade: a [lista de endereçamento do jQuery](#) já está funcionando – inscreva-se e participe das discussões!

Novidade: o [blog do jQuery](#) acabou de ser lançado. [Cadastre-se agora!](#)

O que é o jQuery?

jQuery é uma biblioteca Javascript que leva o seguinte lema a sério: **Programar usando Javascript deve ser algo divertido**. O jQuery atinge este objetivo eliminando todas as marcações desnecessárias de tarefas comuns e repetitivas e encurtando-as, deixando-as inteligentes e compreensíveis.

Como é o jQuery? **Veja um exemplo rápido:**

```
$(\"p.surpresa\") .addClass(\"minhanossa\") .show(\"devagar\");
```

 » Executar

O trecho de código acima localiza todos os parágrafos com uma classe "surpresa", adiciona a classe "minha nossa" a eles e, então, revela-os lentamente. Clique no botão "Executar" para ver isso na prática!

Outras informações:

- O jQuery é compatível com [CSS 1-3](#) e com [X-Path](#) básico.
- O jQuery tem cerca de [10 KB](#).
- O jQuery funciona no Internet Explorer, Firefox, Safari e Opera.

Muitos desenvolvedores web gostaram da simplicidade e o poder do jQuery, e a biblioteca de John logo ficou popular.

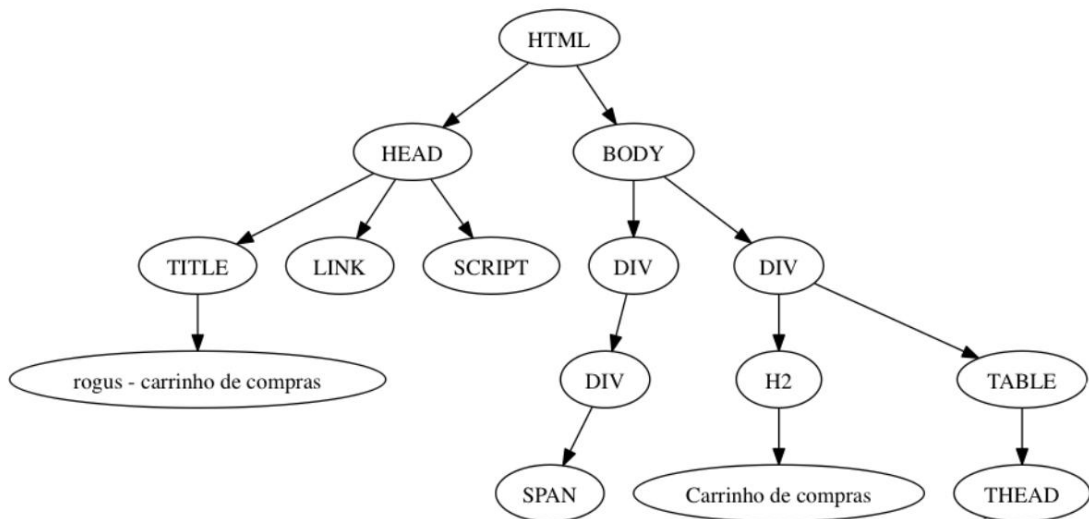
Hoje, jQuery é a biblioteca JS mais popular e é mantida pela Fundação jQuery, uma grande equipe de voluntários.

Além disso, as maiores empresas da web usam JQuery, como:

- Google;
- Microsoft;
- IBM;
- Netflix.

3. DOM

DOM significa Document Object Model ou Objeto Modelo do Documento. É um padrão do W3C que define como documentos HTML, XHTML e XML são acessados. Esses documentos são acessados através de árvore, ou seja, o navegador cria uma representação interna desses documentos ou de qualquer página web em forma de árvore, como a representada abaixo:



A árvore representa o código abaixo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>rogus - carrinho de compras</title>
    <link>
    <script></script>
  </head>
  <body>
    <div>
      <div>
        <span></span>
      </div>
    </div>
    <div>
      <h2>Carrinho de compras</h2>
      <table>
        <thead></thead>
      </table>
    </div>
  </body>
</html>
```

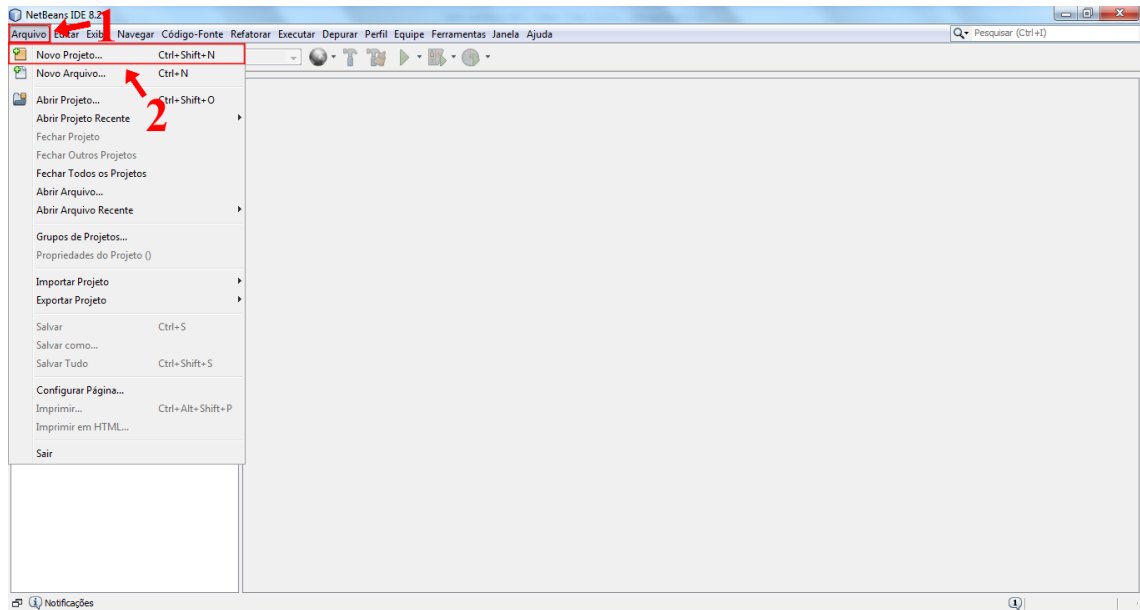
Quando utilizamos JavaScript para ler ou escrever dados numa página HTML, estamos na verdade manipulando o DOM.

4. Inclusão do JQuery

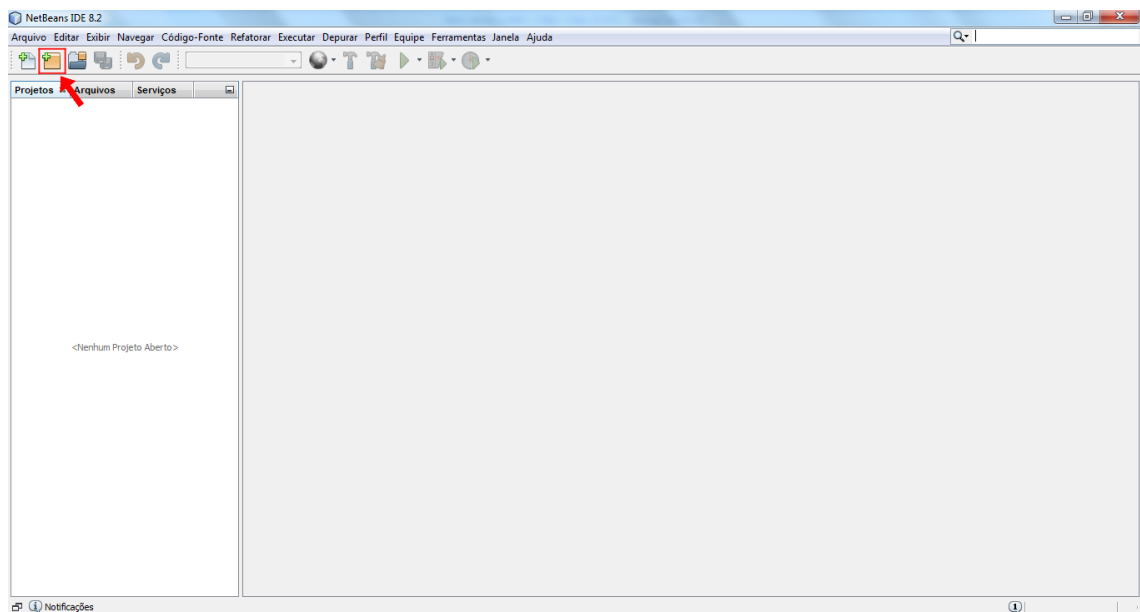
O JQuery pode ser incluído nos projetos Web de duas formas diferentes: uma forma é através de CDN (Content Delivery Network) e a outra forma é através do download da biblioteca diretamente no site do JQuery.

4.1. Criando um projeto no NetBeans

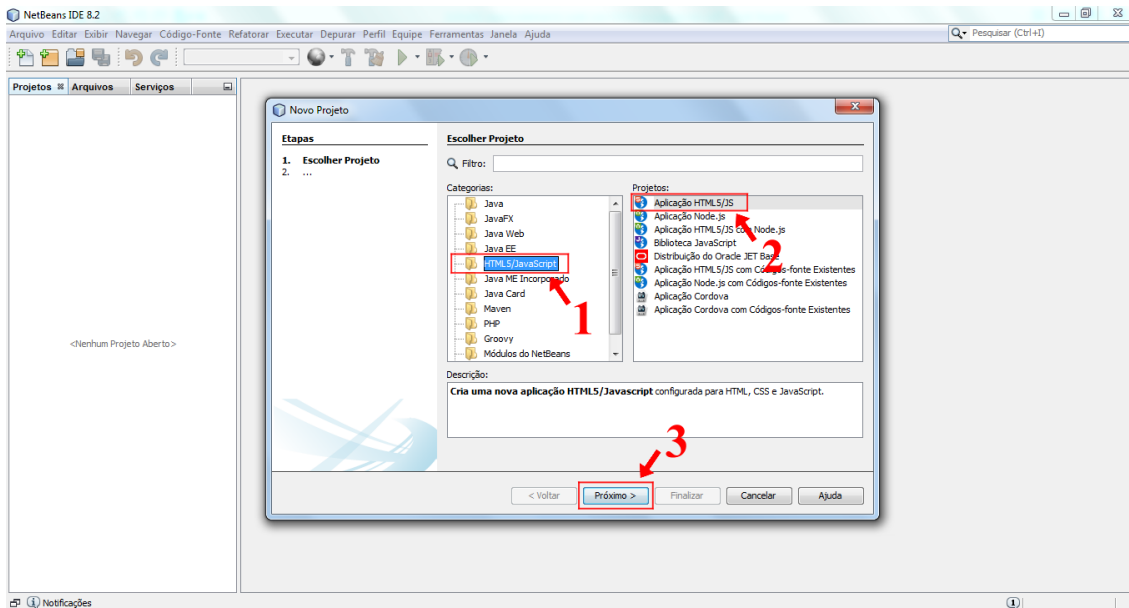
Abra o Netbeans e crie um projeto chamado jQuery.



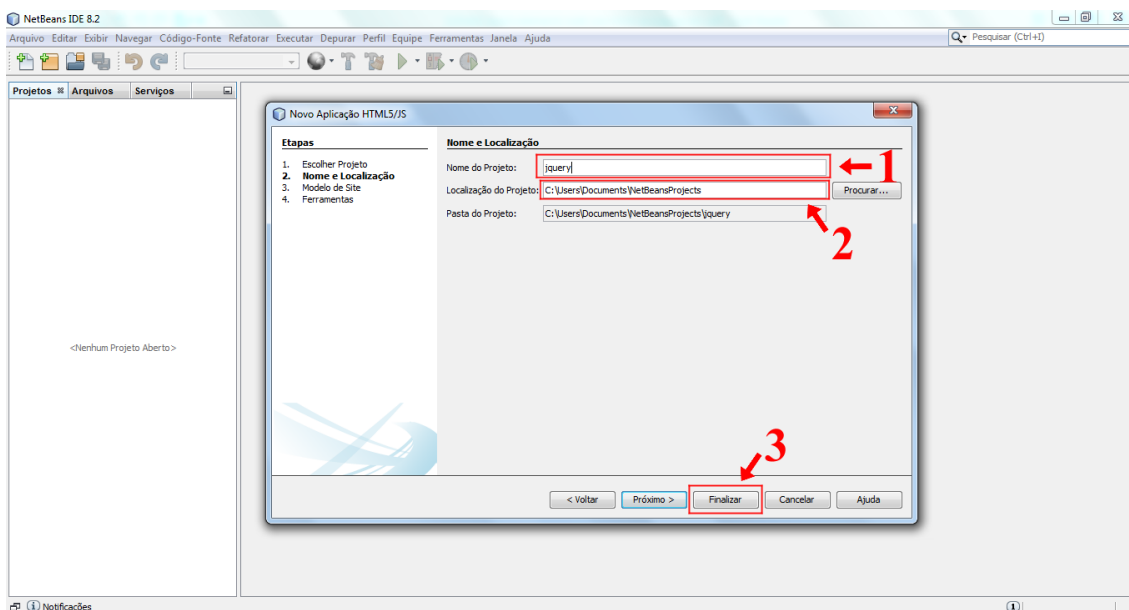
1. No Netbeans, clique em Arquivo.
2. Clique em Novo Projeto.



Ou, no Netbeans, clique no ícone mostrado acima.



1. Agora, clique em HTML5/JavaScript.
2. Em seguida, clique em Aplicação HTML5/JS.
3. Clique em Próximo.



1. Por último, defina o nome do projeto.
2. A localização do projeto.
3. Clique em finalizar para criar um projeto no Netbeans.

4.2. CDN

Uma forma de incluir o JQuery em um projeto é através de CDN (Content Delivery Network), que é uma rede de computadores conectados a internet focada na distribuição de conteúdo. Algumas opções de CDN são:

- CDN do JQuery (MaxCDN).

```
<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
```

- Google CDN.

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

- Microsoft CDN.

```
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery -  
1.10.2.min.js"></script>
```

Adicione uma dessas CDN na página HTML, conforme o código abaixo:

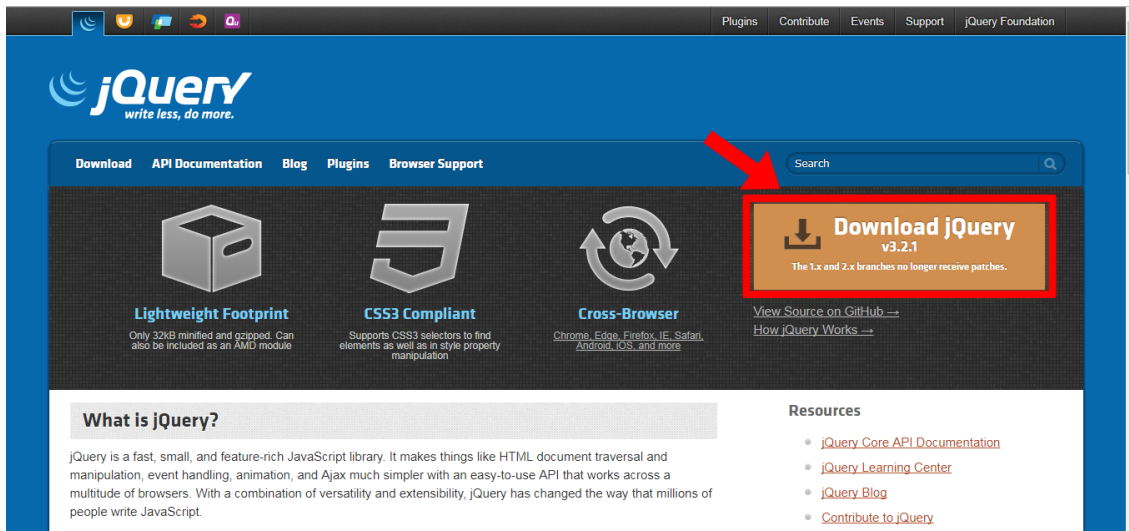
```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <title>jQuery</title>  
    <meta charset="UTF-8">  
    <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>  
  </head>  
  <body>  
    <div></div>  
  </body>  
</html>
```

4.3. Download

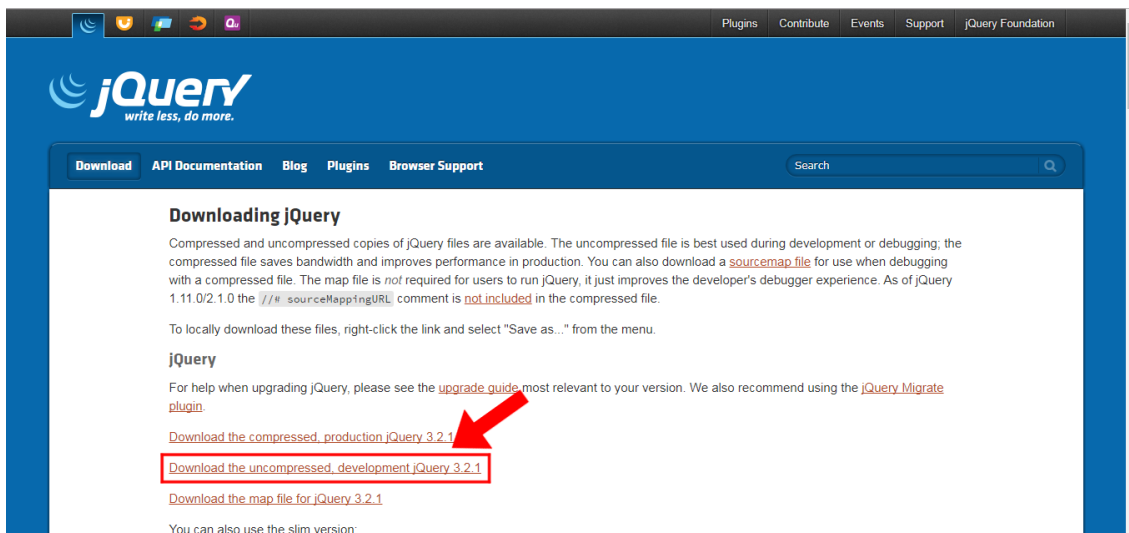
Através de download, para utilizar o JQuery, basta adicionar a biblioteca do JQuery no documento HTML. A biblioteca pode ser obtida no site <http://jquery.com/> e está disponível em duas versões: Compressed e Uncompressed. A primeira versão deve ser utilizada para não sobrecarregar a transferência de dados entre o servidor e os navegadores. Já para desenvolvimento, podemos utilizar a segunda versão, pois ela possui um código bem mais legível o que facilita a depuração.

Para utilizar essa forma siga os passos abaixo:

Acesse o site do JQuery.



Clique em Download jQuery.



Baixe a versão descompactada.

Adicione o arquivo baixado no projeto e faça o link no documento HTML, conforme o código abaixo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery</title>
    <meta charset="UTF-8">
    <script src="jquery-3.2.1.js"></script>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

5. Sintaxe

- \$("seletor").ação();

A sintaxe do JQuery é bastante simples. Primeiramente deve-se colocar o símbolo \$ que representa o JQuery, para em seguida definir qual seletor usar entre aspas e parênteses, e por último definir a ação que se deseja executar.

Exemplo:

```
1 $("body").click(function(){
2   alert("Exemplo");
3 });
```

No exemplo, a ação é o evento click que após um clique na página é exibida a mensagem Exemplo.

6. Seletores

Seletor	Descrição	Exemplo
"*"	Seleciona todos os elementos.	Selecionando todos os elementos. \$("*")
"#id"	Seleciona o elemento com o id especificado.	Selecionando o elemento com id igual a conteudo. \$("#conteudo")
".class"	Seleciona todos os elementos da classe especificada.	Selecionando os elementos da classe titulo. \$(".titulo")
"tipo"	Seleciona todos os elementos do tipo especificado.	Selecionando os divs. \$("div")
"s1 s2"	Seleciona todos os elementos que combinam com o seletor s2 e são descendentes de elementos que combinam com o seletor s1.	Selecionando os spans descendentes de um div. \$("div span")
"s1, s2, s3"	Seleciona todos os elementos que combinam com pelo menos um dos seletores especificados.	Selecionando os divs ou os elementos da classe titulo. \$("div, .titulo")
[atributo]	Seleciona os elementos que possuem o atributo especificado.	Selecionando os divs que possuem id. \$("div[id]")
[atributo="valor"]	Seleciona os elementos tal que o valor de um determinado atributo é igual ao valor especificado.	Selecionando os imgs com title igual a logo. \$("img[title='logo']")
[atributo!="valor"]	Seleciona os elementos tal que o valor de um determinado atributo não é igual ao valor especificado.	Selecionando os imgs com title diferente de logo. \$("img[title!='logo']")
[atributo^="valor"]	Seleciona os elementos tal que o valor de um determinado atributo começa com o valor especificado.	Selecionando os imgs com title iniciando com a string logo. \$("img[title^='logo']")
[atributo\$="valor"]	Seleciona os elementos tal que o valor de um determinado atributo termina com o valor especificado.	Selecionando os imgs com title terminando com a string logo. \$("img[title\$='logo']")
[atributo*="valor"]	Seleciona os elementos tal que o valor de um determinado atributo contém o valor especificado como substring.	Selecionando os imgs com title contendo a string logo. \$("img[title*='logo']")

[atributo~="valor"]	Seleciona os elementos tal que o valor de um determinado atributo contém o valor especificado como palavra.	Selecionando os imgs com title contendo a palavra logo. \$("img[title~='logo']")
[atributo =“valor”]	Seleciona os elementos tal que o valor de um determinado atributo é igual ao valor especificado ou inicia com esse valor seguido do caractere “-”.	Selecionando os imgs com title iniciando com o prefixo logo. \$("img[title =‘logo’]”)
[f1][f2]...[fn]	Seleciona os elementos de acordo com os filtros de atributo especificados.	Selecionando os inputs com id e type igual a text. \$("input[id][type=‘text’]”)
“s1 > s2”	Seleciona todos os elementos que combinam com o seletor s2 e são filhos de elementos que combinam com o seletor s1.	Selecionando os spans filhos de um div. \$("div > span")
“s1 + s2”	Seleciona todos os elementos que combinam com o seletor s2 e são irmãos sucessores adjacentes de elementos que combinam com o seletor s1.	Selecionando os ps sucessores adjacentes de um h1. \$("h1 + p")
“s1 ~ s2”	Seleciona todos os elementos que combinam com o seletor s2 e são irmãos sucessores de elementos que combinam com o seletor s1.	Selecionando os ps sucessores de um h1. \$("h1 ~ p")
:first	Seleciona o primeiro elemento da lista dos elementos que combinam com o seletor utilizado.	Selecionando o primeiro div. \$("div:first")
:last	Seleciona o último elemento da lista dos elementos que combinam com o seletor utilizado.	Selecionando o último div. \$("div:last")
:eq(n)	Seleciona o n-ésimo elemento da lista dos elementos que combinam com o seletor utilizado.	Selecionando o quarto div. \$("div:eq(3)")
:gt(n)	Da lista de elementos que combinam com o seletor utilizado, seleciona do (n+1)-ésimo elemento até o	Selecionando do quarto ao último div. \$("div:gt(2)")

	último.	
:lt(n)	Da lista de elementos que combinam com o seletor utilizado, seleciona do primeiro elemento até o (n-1)-ésimo.	Selecionando do primeiro ao décimo div. \$("div:lt(10)")
:even	Da lista de elementos que combinam com o seletor utilizado, seleciona todos os elementos de índice par.	Selecionando os divs de índice 0, 2, 4, 6... \$("div:even")
:odd	Da lista de elementos que combinam com o seletor utilizado, seleciona todos os elementos de índice ímpar.	Selecionando os divs de índice 1, 3, 5, 7... \$("div:odd")
:first-child	Seleciona os elementos que não possuem um irmão antecessor.	Selecionando os divs que não possuem um irmão antecessor. \$("div:first-child")
:first-of-type	Seleciona os elementos que não possuem um irmão antecessor do mesmo tipo.	Selecionando os divs que não possuem um irmão div antecessor. \$("div:first-of-type")
:last-child	Seleciona os elementos que não possuem um irmão sucessor.	Selecionando os divs que não possuem um irmão sucessor. \$("div:last-child")
:last-of-type	Seleciona os elementos que não possuem um irmão sucessor do mesmo tipo.	Selecionando os divs que não possuem um irmão div sucessor. \$("div:last-of-type")
:nth-child(n)	Seleciona os elementos que possuem (n - 1) irmãos antecessores.	Selecionando os divs que possuem exatamente 3 irmãos antecessores. \$("div:nth-child(4)")
:nth-last-child(n)	Seleciona os elementos que possuem (n - 1) irmãos sucessores.	Selecionando os divs que possuem exatamente 3 irmãos sucessores. \$("div:nth-last-child(4)")
:nth-of-type(n)	Seleciona os elementos que possuem (n - 1) irmãos antecessores do mesmo tipo.	Selecionando os divs que possuem exatamente 3 irmãos div antecessores. \$("div:nth-of-type(4)")
:nth-last-of-type(n)	Seleciona os elementos que possuem (n - 1) irmãos sucessores do mesmo tipo.	Selecionando os divs que possuem exatamente 3 irmãos div sucessores. \$("div:nth-last-of-type(4)")
:only-child	Seleciona todos os elementos que não possuem irmãos.	Selecionando os divs que não possuem irmãos. \$("div:only-child")
:only-of-type	Seleciona todos os elementos que não possuem irmãos do mesmo tipo.	Selecionando os divs que não possuem irmãos div. \$("div:only-of-type")
:parent	Seleciona todos os elementos que possuem filhos.	Selecionando os divs que possuem filho. \$("div:parent")
:hidden	Seleciona os elementos	Selecionando os divs hidden.

	considerados hidden. Os elementos que não ocupam espaço na página são considerados hidden.	<code>\$("div:hidden")</code>
<code>:visible</code>	Seleciona os elementos considerados visible. Os elementos que ocupam espaço na página são considerados visible.	Selecionando os divs visible. <code>\$("div:visible")</code>
<code>:disabled</code>	Seleciona todos os elementos desabilitados.	Selecionando os inputs desabilitados. <code>\$("input:disabled")</code>
<code>:enabled</code>	Seleciona todos os elementos habilitados.	Selecionando os inputs habilitados. <code>\$("input:enabled")</code>
<code>:contains("texto")</code>	Seleciona todos os elementos que possuem o texto especificado em seu conteúdo.	Selecionando os divs que contêm o texto exemplo. <code>\$("div:contains('exemplo'))"</code>
<code>:empty</code>	Seleciona todos os elementos sem conteúdo.	Selecionando os divs que não possuem conteúdo. <code>\$("div:empty")</code>
<code>:has(selector)</code>	Seleciona os elementos que possuem pelo menos um descendente que combina com o seletor passado como parâmetro.	Selecionando os divs que possuem pelo menos um img. <code>\$("div:has(img))"</code>
<code>:not(selector)</code>	Seleciona os elementos que não combinam com o seletor especificado.	Selecionando os elementos que não são div. <code>\$("div:not(div))"</code>
<code>:button</code>	Seleciona todos os elementos button ou com <code>type="button"</code> .	Selecionando os inputs com <code>type="button"</code> . <code>\$("input:button")</code>
<code>:file</code>	Seleciona todos os elementos com <code>type="file"</code> .	Selecionando os inputs com <code>type="file"</code> . <code>\$("input:file")</code>
<code>:image</code>	Seleciona todos os elementos com <code>type="image"</code> .	Selecionando os inputs com <code>type="image"</code> . <code>\$("input:image")</code>
<code>:password</code>	Seleciona todos os elementos com <code>type="password"</code> .	Selecionando os inputs com <code>type="password"</code> . <code>\$("input:password")</code>
<code>:radio</code>	Seleciona todos os elementos com <code>type="radio"</code> .	Selecionando os inputs com <code>type="radio"</code> . <code>\$("input:radio")</code>
<code>:checkbox</code>	Seleciona todos os elementos com <code>type="checkbox"</code> .	Selecionando os inputs com <code>type="checkbox"</code> . <code>\$("input:checkbox")</code>
<code>:reset</code>	Seleciona todos os elementos com	Selecionando os inputs com <code>type="reset"</code> .

	type="reset".	<code>\$("input:reset")</code>
:text	Seleciona os elementos com type="text".	Selecionando os inputs com type="text". <code>\$("input:text")</code>
:submit	Seleciona os elementos com type="submit".	Selecionando os inputs com type="submit". <code>\$("input:submit")</code>
:lang(linguagem)	Seleciona todos os elementos com lang igual à linguagem especificada.	Selecionando os divs com lang="pt-br" <code>\$("div:lang(pt-br)")</code>
:animated	Seleciona todos os elementos que estão com alguma animação em andamento.	Selecionando os divs com animação em andamento. <code>\$("div:animated")</code>

7. Métodos

add

Adiciona elementos.

Exemplo: adicionando um div ao body da página.

```
$("body").add("div");
```

each

Permite executar uma função para cada elemento de um conjunto.

Exemplo: exibe a largura de todos os divs da página.

```
$("div").each(function(){  
    alert($(this).width());  
});
```

css

Esse método é utilizado para dois propósitos.

O primeiro é recuperar os valores das propriedades CSS do primeiro elemento selecionado.

Exemplo: recuperando a propriedade CSS color do body da página.

```
var color = $("body").css("color");
```

O segundo é alterar os valores das propriedades CSS de todos os elementos selecionados.

Exemplo: alterando as propriedades color, width e height do body da página.

```
$("body").css({  
    color: "blue",  
    width: "500px",  
    height: "500px"  
});
```

html

Esse método é utilizado para dois propósitos.

O primeiro é recuperar o conteúdo HTML do primeiro elemento selecionado.

Exemplo: recuperando o conteúdo HTML do body da página.

```
var conteudo = $("body").html();
```

O segundo é alterar o conteúdo HTML de todos os elementos selecionados.

Exemplo: alterando o conteúdo HTML do body da página.

```
$("body").html("<p>Exemplo</p>");
```

text

Esse método é utilizado para dois propósitos.

O primeiro é recuperar o texto contido no corpo do primeiro elemento selecionado.

Exemplo: recuperando o texto contido no corpo do primeiro div.

```
var texto = $("div").text();
```

O segundo é alterar o texto contido no corpo de todos os elementos seleccionados.
Exemplo: alterando o texto contido no corpo do div.

```
$("div").text("Exemplo");
```

8. Métodos de eventos

on

O tratamento de eventos podem ser definidos com o método on. No exemplo abaixo, o primeiro parâmetro é o nome do evento que será tratado e o segundo parâmetro a função que tratará o evento.

```
$("#body").on("click", function(){  
    alert("click");  
});
```

O tratamento de diversos eventos pode ser definido como uma única chamada do método on.

```
$("#body").on("click mouseenter mouseleave", function(){  
    alert("click mouseenter mouseleave");  
});
```

off

O tratamento de um evento pode ser eliminado com o método off. A seguir, mostramos algumas formas de utilização desse método.

Removendo todos os tratamentos associados aos eventos dos parágrafos.

```
$("#p").off();
```

Removendo todos os tratamentos associados ao evento de clique dos parágrafos.

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

Removendo um tratamento específico associado ao evento de clique dos parágrafos.

```
$("#p").off("click", tratamento);
```

Atalhos

O tratamento de um evento pode ser definido através de métodos que funcionam como atalhos para o método on. No exemplo abaixo, utilizamos o método click que é um atalho para on("click", função).

```
$("#body").click(function(){  
    alert("click");  
});
```

Analogamente, para cada evento, há um método de atalho com o mesmo nome.

9. Eventos

Evento	Descrição
ready	Esse evento é disparado após o carregamento da árvore de elementos do documento HTML.
resize	Esse evento é disparado quando a janela do navegador é redimensionada.
scroll	Esse evento é disparado quando a barra de rolagem de um elemento é movimentada.
focus	Esse evento é disparado quando um elemento ganha o foco.
focusin	Esse evento é disparado quando um elemento ou um dos seus descendentes ganha o foco.
blur	Esse evento é disparado quando um elemento perde o foco.
focusout	Esse evento é disparado quando um elemento ou um dos seus descendentes perde o foco.
select	Esse evento é disparado quando o valor de um elemento é selecionado.
change	Em determinados casos, esse evento é disparado depois de um elemento ter o seu valor modificado e em seguida perder o foco. Em outros casos, esse evento é disparado depois de um elemento ter o seu valor modificado e selecionado nessa ordem.
keydown	Esse evento é disparado quando uma tecla é pressionada.
keypress	Esse evento é semelhante ao evento keydown. Contudo, ele não é disparado para teclas que não representam caracteres (exemplo: SHIFT, CTRL, ALT, entre outros). O keypress é disparado depois do keydown.
keyup	Esse evento é disparado quando uma tecla é solta.
click	Esse evento é disparado quando o usuário clica sobre um elemento com o botão esquerdo ou com o botão do meio do mouse.
dblclick	Esse evento é disparado quando o usuário realiza um duplo clique sobre um elemento com o botão esquerdo ou com o botão do meio do mouse.
mousedown	Esse evento é disparado quando o usuário pressiona um botão do mouse sobre um determinado elemento.
mouseup	Esse evento é disparado quando o usuário solta um botão do mouse sobre um determinado elemento.
mouseenter	Esse evento é disparado no momento em que o ponteiro do mouse passa a estar sobre um determinado elemento.
mouseleave	Esse evento é disparado no momento em que o ponteiro do mouse passa a não estar sobre um determinado elemento.
hover	Esse evento é disparado no mouseenter e mouseleave.
mouseover	Esse evento é disparado no momento em que o ponteiro do mouse passa a estar sobre um determinado elemento ou sobre os seus descendentes.
mouseout	Esse evento é disparado no momento em que o ponteiro do mouse passa a não estar sobre um determinado elemento ou sobre os seus descendentes.
mousemove	Esse evento é disparado quando o ponteiro do mouse se move sobre um determinado elemento ou sobre os seus descendentes.
submit	Esse evento é disparado quando um formulário é enviado.

10. AJAX

Durante o desenvolvimento de sites e sistemas web em geral, é comum a necessidade de consumir recursos do servidor sem que a página precise ser carregada novamente. Para atender esse tipo de situação, podemos utilizar o conceito de Ajax, definido, em poucas palavras, como um conjunto de técnicas que utilizam JavaScript para carregar informações de forma assíncrona.

Ao realizar chamadas assíncronas, o fluxo do código não é interrompido até que a resposta seja obtida. Ao invés disso, após realizar a requisição, a resposta é obtida em um momento posterior, de forma independente, e tratada por meio de uma função (chamada função de callback).

AJAX significa Asynchronous JavaScript e XML. Com AJAX a página inteira não precisa ser recarregada, permitindo maior interatividade, velocidade de processamento e usabilidade. Ele pode enviar bem como receber informações em uma variedade de formatos como HTML, XML, PHP, TXT, entre outros.

Os recursos utilizados pelo AJAX são:

- Fazer requisições para o servidor sem recarregar a página.
- Receber e trabalhar com dados do servidor.

Nota-se que Ajax não é uma linguagem de programação, mas uma nova forma de usar os padrões existentes.

10.1. Estado de prontidão

Quando o AJAX faz solicitações de recursos para o servidor, ele passa por determinados estados até que a resposta do servidor esteja concluída. Cada um desses estados são chamados de estado de prontidão e existem os seguintes estados de prontidão possíveis:

- Estado de prontidão 0: quando se é criado o objeto de solicitação.
- Estado de prontidão 1: quando a solicitação sabe como e com o que se conectar.
- Estado de prontidão 2: a solicitação está em progresso.
- Estado de prontidão 3: os dados estão sendo baixados no objeto de solicitação.
- Estado de prontidão 4: a resposta do servidor está concluída e todos os dados estão no objeto de solicitação.

10.2. Código de status

Outro retorno importante do servidor é o código de Status. O código de Status é um código gerado pelo servidor onde o servidor informa se encontrou o recurso de solicitação e o que esse recurso retornou ao acessá-lo.

É sempre importante lembrar do caminho percorrido entre o navegador e o servidor. O navegador cria um objeto de solicitação e envia uma solicitação ao servidor informando o script que ele deve executar. O servidor por sua vez tenta localizar esse script na url da solicitação passada pelo navegador. Às vezes o servidor não consegue encontrar esse programa ou tem problemas para acessar o recurso solicitado. Assim, o servidor retorna um código de Status informando a situação atual.

Os status mais importantes são:

- Código 404: o servidor não conseguiu encontrar um recurso.
- Código 200: indica que não houve nenhum problema ao acessar o recurso.

Outros status existentes são:

- Código 400: Bad Request
- Código 401 Unauthorized
- Código 403 Forbidden
- Código 405 Method Not Allowed
- Código 406 Not Acceptable
- Código 408 Request Timeout

10.3. Método ajax do JQuery

No JQuery, existe o método ajax, que com uma sintaxe bastante simples, permite enviar e tratar o resultado de requisições assíncronas.

Exemplo:

Nome...:

Salário...:

Dados enviados, João

```
$.ajax({
  url: "cadastrar.php",
  type: "post",
  data: {
    nome: "João",
    salario: "4000"
  },
  beforeSend: function(){
    $("#resultado").html("ENVIANDO...");
  }
})
.done(function(msg){
  $("#resultado").html(msg);
})
.fail(function(jqXHR, textStatus, msg){
});

$.ajax({
  url: "listarDados.php",
  success: function(resultado){
    $("#resultado").html(resultado);
  }
});
```

11. JQuery UI

O JQuery UI é uma ferramenta de interação construída em cima do JQuery que permite animar os diferentes elementos HTML, lhes dando impressionantes interações front-end com relativa facilidade. Ou seja, é uma rica e extensível biblioteca de componentes gráficos criada para maximizar o uso e poder do JQuery. Totalmente baseada no framework jQuery, é usada para fornecer uma melhor interação entre o usuário e o cliente (web browser), com recursos ricos como animação, efeitos, componentes estilizáveis, entre outros.

O pacote de interface é basicamente uma coleção de funções de interface de usuário que estão relacionadas e podem ser divididas em três módulos principais:

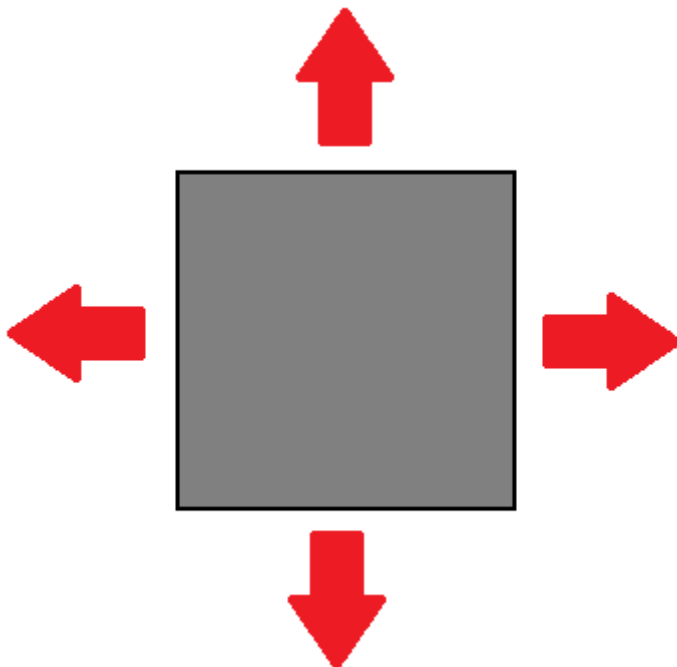
- Elementos de interação com o mouse.
- Widgets, que contêm interfaces de usuário pré-construídas e customizáveis.
- Efeitos, que são animações muito simples e fáceis que podem ser feitas para um elemento de uma página.

11.1. Elementos de interação com o mouse

Quando um elemento HTML se arrasta em detrimento de outro ou quando o mesmo está por cima de outro elemento, pode existir uma interação forte entre os dois. Isso constitui uma interação.

Drag and Drop

O Drag and Drop permite que elementos sejam movidos usando o mouse.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Draggable</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">

<script src="jquery.js"></script>
<script src="jquery-ui.js"></script>

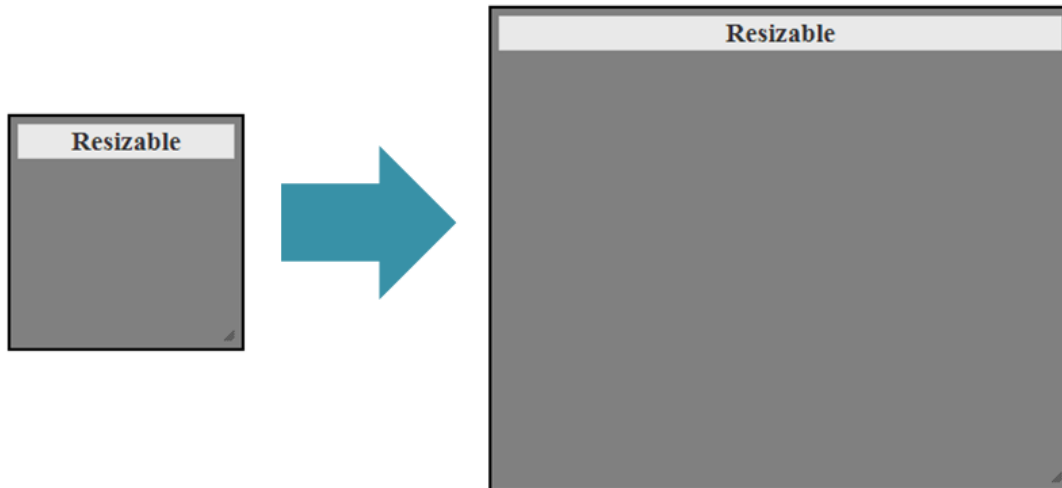
<style>
  #draggable {
    border: 2px solid black;
    background-color: gray;
    width: 150px;
    height: 150px;
    padding: 5px;
  }
</style>

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

</head>
<body>
  <div id="draggable">
    <p>Drag and Drop</p>
  </div>
</body>
</html>
```

Resizable

O Resizable muda o tamanho do elemento usando o mouse.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Resizable</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

```
    <link rel="stylesheet"
```

```
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
    <link rel="stylesheet" href="/resources/demos/style.css">
```

```
    <script src="jquery.js"></script>
```

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

```
  <style>
```

```
    #resizable {
```

```
      border: 2px solid black;
```

```
      background-color: gray;
```

```
      width: 150px;
```

```
      height: 150px;
```

```
      padding: 5px;
```

```
    }
```

```
    #resizable h3 {
```

```
      text-align: center;
```

```
      margin: 0;
```

```
    }
```

```
  </style>
```

```

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

</head>
<body>
  <div id="resizable">
    <h3 class="ui-widget-header">Resizable</h3>
  </div>
</body>
</html>

```

Sortable

O Sortable reordena os elementos de uma lista ou grid usando o mouse.



```
<!DOCTYPE html>
```

```

<html>
  <head>
    <title>Sortable</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
    <link rel="stylesheet" href="/resources/demos/style.css">

    <script src="jquery.js"></script>
    <script src="jquery-ui.js"></script>

    <style>

```

```

#sortable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
#sortable li { margin: 0 3px 3px 3px; padding: 0.4em; padding-left:
1.5em; font-size: 1.4em; height: 18px; }
#sortable li span { position: absolute; margin-left: -1.3em; }
</style>

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

</head>
<body>
<ul id="sortable">
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 1</li>
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 2</li>
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 3</li>
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 4</li>
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 5</li>
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 6</li>
<li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-
2-n-s"></span>Item 7</li>
</ul>
</body>
</html>

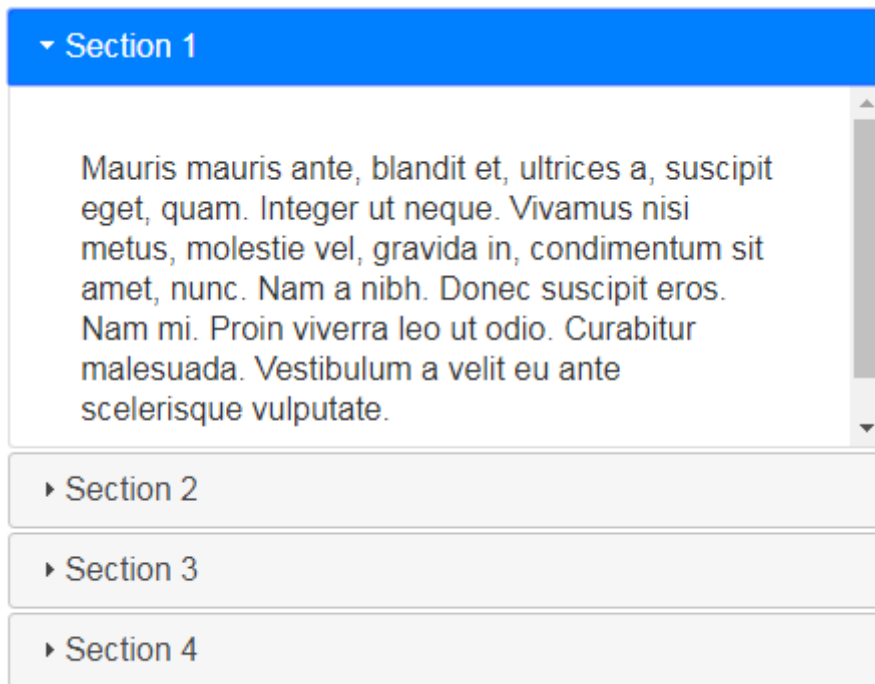
```

11.2. Widgets

Widget é um termo técnico criado para definir componentes ou pedaços de interface gráfica que proveem funcionalidades básicas como: botões, listas, ícones, janelas, etc. Estes são muito comuns em Sistemas Operacionais para representar pequenos aplicativos que pairam pelo desktop com funcionalidades específicas.

Accordion

O Accordion exibe painéis de conteúdo flexíveis para apresentar informações em uma quantidade limitada de espaço.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Accordion</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

```
    <link rel="stylesheet"
```

```
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
    <link rel="stylesheet" href="/resources/demos/style.css">
```

```
    <script src="jquery.js"></script>
```

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

```
    <script>
```

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

```
        $("#accordion").accordion();
```

```
      });
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
    <div id="accordion">
```

```
      <h3>Section 1</h3>
```

<div>

<p>

Integer Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam.
ut neque. Vivamus nisi metus, molestie vel, gravida in,
condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin
viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante
scelerisque vulputate.

</p>

</div>

<h3>Section 2</h3>

<div>

<p>

Sit amet Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum
porttitor purus. Vivamus hendrerit, dolor at aliquet laoreet, mauris turpis
quam. In velit, faucibus interdum tellus libero ac justo. Vivamus non
suscipit faucibus urna.

</p>

</div>

<h3>Section 3</h3>

<div>

<p>

Quisque lobortis. Nam enim risus, molestie et, porta ac, aliquam ac, risus.

Phasellus pellentesque purus in massa. Aenean in pede.

Phasellus ac libero

magna quis ac tellus pellentesque semper. Sed ac felis. Sed commodo,

venenatis dui. lacinia ornare, quam ante aliquam nisi, eu iaculis leo purus

</p>

List item one

List item two

List item three

</div>

<h3>Section 4</h3>

<div>

<p>

netus Cras dictum. Pellentesque habitant morbi tristique senectus et
 primis in et malesuada fames ac turpis egestas. Vestibulum ante ipsum
 lacinia faucibus orci luctus et ultrices posuere cubilia Curae; Aenean
 mauris vel est.

</p>
 <p>
 consequat lectus. Suspendisse eu nisl. Nullam ut libero. Integer dignissim
 nostra, per Class aptent taciti sociosqu ad litora torquent per conubia
 inceptos himenaeos.

</p>
 </div>
 </div>
 </body>
 </html>

Autocomplete

O Autocomplete fornece sugestões enquanto é digitado no campo especificado.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Autocomplete</title>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

```
<link rel="stylesheet"
```

```
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
<link rel="stylesheet" href="/resources/demos/style.css">
```

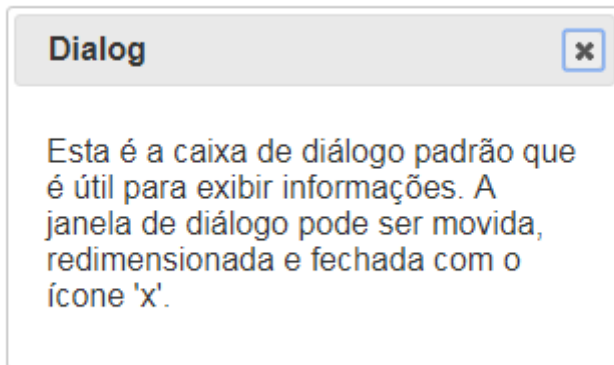
```
<script src="jquery.js"></script>
<script src="jquery-ui.js"></script>
```

```
<script>
  $( function() {
    var availableTags = [
      "ActionScript",
      "AppleScript",
      "Asp",
      "BASIC",
      "C",
      "C++",
      "Clojure",
      "COBOL",
      "ColdFusion",
      "Erlang",
      "Fortran",
      "Groovy",
      "Haskell",
      "Java",
      "JavaScript",
      "Lisp",
      "Perl",
      "PHP",
      "Python",
      "Ruby",
      "Scala",
      "Scheme"
    ];
    $( "#tags" ).autocomplete({
      source: availableTags
    });
  });
</script>
```

```
</head>
<body>
  <div class="ui-widget">
    <label for="tags">Tags: </label>
    <input id="tags">
  </div>
</body>
</html>
```

Dialog

O Dialog abre o conteúdo em uma janela de diálogo sobreposta na página.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Dialog</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

```
    <link rel="stylesheet"
```

```
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
    <link rel="stylesheet" href="/resources/demos/style.css">
```

```
    <script src="jquery.js"></script>
```

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

```
    <script>
```

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

```
        $("#dialog").dialog();
```

```
      });
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
    <div id="dialog" title="Dialog">
```

```
      <p>Esta é a caixa de diálogo padrão que é útil para exibir informações.  
A janela de diálogo pode ser movida, redimensionada e fechada com o ícone  
'x'.</p>
```

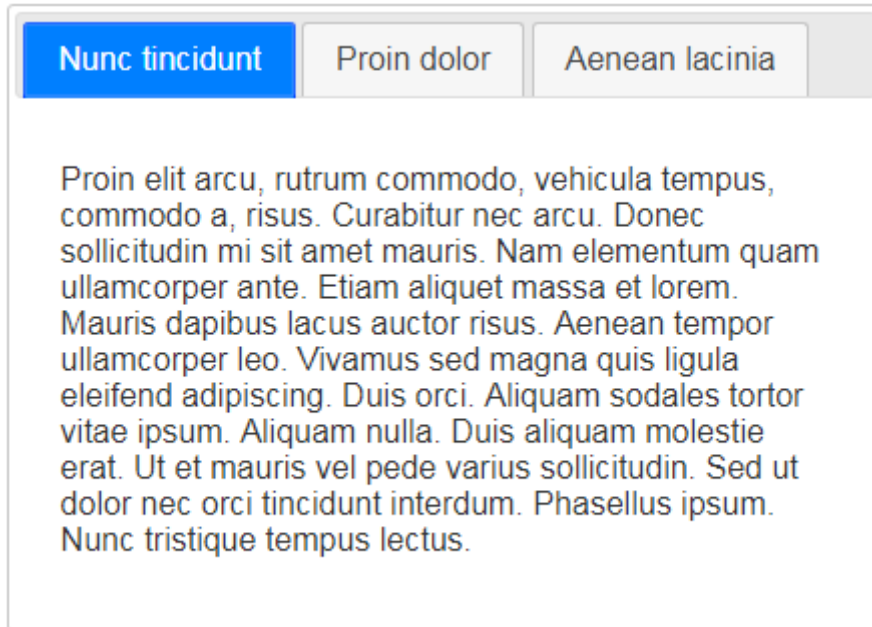
```
    </div>
```



```
</body>
</html>
```

Tabs

Tabs permite trocar o conteúdo entre as guias utilizando a mesma área.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Tabs</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
    <link rel="stylesheet"
```

```
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
    <link rel="stylesheet" href="/resources/demos/style.css">
```

```
    <script src="jquery.js"></script>
```

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

```
    <script>
```

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

```
        $("#tabs").tabs();
```

```
      });
```

```
    </script>
```

```
</head>
```

```
<body>
```

```
<div id="tabs">
```

```
<ul>
```

```
<li><a href="#tabs-1">Nunc tincidunt</a></li>
```

```
<li><a href="#tabs-2">Proin dolor</a></li>
```

```
<li><a href="#tabs-3">Aenean lacinia</a></li>
```

```
</ul>
```

```
<div id="tabs-1">
```

```
<p>Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.</p>
```

```
</div>
```

```
<div id="tabs-2">
```

```
<p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem. Suspendisse sed ligula in ligula suscipit aliquam. Praesent in eros vestibulum mi adipiscing adipiscing. Morbi facilisis. Curabitur ornare consequat nunc. Aenean vel metus. Ut posuere viverra nulla. Aliquam erat volutpat. Pellentesque convallis. Maecenas feugiat, tellus pellentesque pretium posuere, felis lorem euismod felis, eu ornare leo nisi vel felis. Mauris consectetur tortor et purus.</p>
```

```
</div>
```

```
<div id="tabs-3">
```

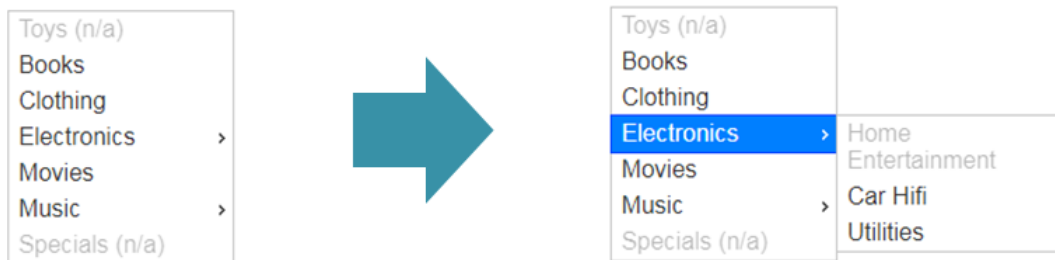
```
<p>Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate, pede vel vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem. Vestibulum non ante. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce sodales. Quisque eu urna vel enim commodo pellentesque. Praesent eu risus hendrerit ligula tempus pretium. Curabitur lorem enim, pretium nec, feugiat nec, luctus a, lacus.</p>
```

```
<p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna. Nullam ac lacus. Nulla facilisi. Praesent viverra justo vitae neque. Praesent blandit adipiscing velit. Suspendisse potenti. Donec mattis, pede vel pharetra blandit, magna ligula faucibus eros, id euismod lacus dolor eget odio. Nam scelerisque. Donec non libero sed nulla mattis commodo. Ut sagittis. Donec nisi lectus, feugiat porttitor, tempor ac, tempor vitae, pede. Aenean vehicula velit eu tellus interdum rutrum. Maecenas commodo.
```

Pellentesque nec elit. Fusce in lacus. Vivamus a libero vitae lectus hendrerit hendrerit.</p></div></div></div></body></html>

Menu

Cria um menu com interações e teclado para navegação.



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Menu</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

```
    <link rel="stylesheet"
```

```
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
    <link rel="stylesheet" href="/resources/demos/style.css">
```

```
    <script src="jquery.js"></script>
```

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

```
    <style>
```

```
      .ui-menu { width: 150px; }
```

```
    </style>
```

```
    <script>
```

```
      $(function(){  
        $("#menu").menu();
```

```
      });
```

```
    </script>
```

```

</head>
<body>
  <ul id="menu">
    <li class="ui-state-disabled"><div>Toys (n/a)</div></li>
    <li><div>Books</div></li>
    <li><div>Clothing</div></li>
    <li><div>Electronics</div>
      <ul>
        <li class="ui-state-disabled"><div>Home Entertainment</div></li>
        <li><div>Car Hifi</div></li>
        <li><div>Utilities</div></li>
      </ul>
    </li>
    <li><div>Movies</div></li>
    <li><div>Music</div>
      <ul>
        <li><div>Rock</div>
          <ul>
            <li><div>Alternative</div></li>
            <li><div>Classic</div></li>
          </ul>
        </li>
        <li><div>Jazz</div>
          <ul>
            <li><div>Freejazz</div></li>
            <li><div>Big Band</div></li>
            <li><div>Modern</div></li>
          </ul>
        </li>
        <li><div>Pop</div></li>
      </ul>
    </li>
    <li class="ui-state-disabled"><div>Specials (n/a)</div></li>
  </ul>
</body>
</html>

```

Button

Melhora os estilos de botões e âncoras.



```

<!DOCTYPE html>

<html>
  <head>
    <title>Button</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
    <link rel="stylesheet" href="/resources/demos/style.css">

    <script src="jquery.js"></script>
    <script src="jquery-ui.js"></script>

    <script>
      $(function(){
        $(".widget input[type=submit], .widget a, .widget button").button();
      });
    </script>

  </head>
  <body>
    <div>
      <h1>Elementos normais</h1>
      <button>Elemento botão</button>

      <input type="submit" value="Botão submit">

      <a href="#">Âncora</a>
    </div>
  </body>
</html>

```

```
</div>
<div class="widget">
  <h1>Widget Buttons</h1>
  <button>Elemento botão</button>

  <input type="submit" value="Botão submit">

  <a href="#">Âncora</a>
</div>
</body>
</html>
```

Selectmenu

Melhora os estilos de selects.

Selecione um número

1	▼
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>Selectmenu</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
    <link rel="stylesheet" href="/resources/demos/style.css">

    <script src="jquery.js"></script>
    <script src="jquery-ui.js"></script>

    <style>
      .select {
        width: 200px;
      }
    </style>

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

  </head>
  <body>
    <div class="select">
      <form action="#">
        <label for="numero">Selecione um número</label>
        <select name="numero" id="selectmenu">
          <option>1</option>
          <option>2</option>
          <option>3</option>
          <option>4</option>
          <option>5</option>
          <option>6</option>
          <option>7</option>
          <option>8</option>
          <option>9</option>
          <option>10</option>
          <option>11</option>
          <option>12</option>
          <option>13</option>
        </select>
      </form>
    </div>
  </body>
</html>
```

```

        <option>14</option>
        <option>15</option>
        <option>16</option>
        <option>17</option>
        <option>18</option>
        <option>19</option>
    </select>
</form>
</div>
</body>
</html>

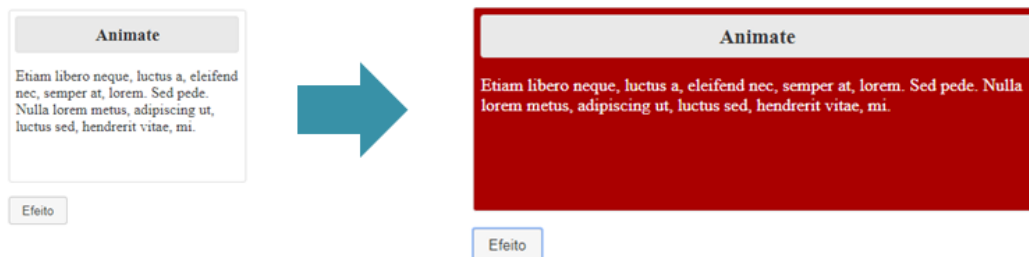
```

11.3. Efeitos

A partir de efeitos é possível dinamizar pedaços de HTML criando resultados bem interessantes. Aqui, serão listados alguns efeitos mais usados e sua forma de uso junto ao JQuery UI.

Color Animation

Anima as propriedades dos elementos usando cores.



```
<!DOCTYPE html>
```

```

<html>
  <head>
    <title>Color Animation</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
    <link rel="stylesheet" href="/resources/demos/style.css">

    <script src="jquery.js"></script>
    <script src="jquery-ui.js"></script>

```



```

<style>
  .toggler { width: 500px; height: 200px; position: relative; }
  #button { padding: .5em 1em; text-decoration: none; }
  #effect { width: 240px; height: 170px; padding: 0.4em; position:
relative; background: #fff; }
  #effect h3 { margin: 0; padding: 0.4em; text-align: center; }
</style>

<script>
  $(function(){
    var state = true;
    $("#button").on("click", function(){
      if(state){
        $("#effect").animate({
          backgroundColor: "#aa0000",
          color: "#fff",
          width: 500
        }, 1000);
      }
      else{
        $("#effect").animate({
          backgroundColor: "#fff",
          color: "#000",
          width: 240
        }, 1000);
      }
      state = !state;
    });
  });
</script>

</head>
<body>
  <div class="toggler">
    <div id="effect" class="ui-widget-content ui-corner-all">
      <h3 class="ui-widget-header ui-corner-all">Animate</h3>
      <p>
        Etiam libero neque, luctus a, eleifend nec, semper at, lorem. Sed
pede. Nulla lorem metus, adipiscing ut, luctus sed, hendrerit vitae, mi.
      </p>
    </div>
  </div>

  <button id="button" class="ui-state-default ui-corner-
all">Efeito</button>

```

```
</body>  
</html>
```