



jQuery / AJAX

Professor Vicente Paulo de Camargo



jQuery / Ajax

Ajax

É o acrônimo de Asynchronous JavaScript and XML

Não é uma tecnologia, mas um conjunto de tecnologias

Permite a comunicação com o servidor de forma assíncrona

Essa comunicação não causa o reload da página, apenas atualiza parte da página

O tratamento dos dados, seu formato e exibição são responsabilidade desse script

O termo Ajax foi inventado por Jesse James Garrett, mas o objeto que permite essa comunicação foi criado pela Microsoft

Esse objeto é o XMLHttpRequest

Inicialmente o XML era o responsável pela comunicação. Hoje é o formato JSON



jQuery / Ajax

Ajax

A sincronia entre duas tarefas ocorre quando a tarefa1 depende da finalização da tarefa2

Neste caso, as duas tarefas são síncronas

No entanto, quando duas tarefas são executadas uma independente da outra, ocorre o processo assíncrono

Neste caso, há um ganho de tempo, pois as duas podem ser executadas paralelamente

Além de possuir recursos para acesso à árvore de elementos do DOM, o Ajax permite o processamento assíncrono das informações, principalmente entre o cliente e o servidor



jQuery / Ajax

jQuery

- É uma biblioteca de funções em JavaScript
- Foi criada por John Resig em 2006, na cidade de New York
- A sua sintaxe permite:
 - simplificar a navegação em documentos HTML
 - selecionar elementos DOM
 - manipular eventos
 - criar animações
 - suporta o uso de Ajax (acrônimo de Asynchronous JavaScript and XML)

jQuery / Ajax

Acesse o jquery.com e efetue o download da última versão dessa biblioteca

Há opção para se selecionar a versão minimizada ou não

A versão minimizada ocupa menos espaço e seu carregamento é mais rápido

Para utilizar essa biblioteca basta referenciá-la:

via CDN ou via arquivo dentro da aplicação

Para que as ações de JavaScript ocorram, é necessário que toda a página esteja carregada

Caso utilize
JavaScript
puro, o
normal é
utilizar um
código
semelhante a:

Caso utilize
jQuery,
utilizar um
código
semelhante a:

```
$(document).ready(function() {  
    //código a executar quando o DOM estiver  
});
```

```
window.onload = function () {  
    //comandos  
}
```

jQuery / Ajax

Assim:

Para verificar o evento **click** sobre o componente de um link, utiliza-se algo como:

```
$(document).ready(function() {  
    $("a").click(function(evento) {  
        //código correspondente ao evento click  
    });  
});
```

Os códigos e funções devem ficar entre as chaves da função **ready** do jQuery

jQuery / Ajax

A biblioteca jQuery possui as funções `addClass()` e `removeClass()` que permite adicionar ou remover uma classe de um objeto HTML.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset='utf-8' lang='pt-br'>
5     <style>
6       .teste{
7         background:green;
8         font-weight:bold;
9       }
10    </style>
11  </head>
12  <body>
13    <a href="google.com.br">Google</a>
14    <div id="camada">
15      conteudo de exemplo
16    </div>
17
18    <script src="jquery-3.5.1.min.js"></script>
19    <script>
20      $(document).ready(function() {
21        $("a").mouseover(function(event) {
22          $("#camada").addClass("teste");
23        });
24        $("a").mouseout(function(event) {
25          $("#camada").removeClass("teste");
26        });
27      });
28    </script>
29  </body>
30 </html>
```

Adiciona e
remove a
classe css do
elemento
div camada

jQuery / Ajax

O método `css(...)` permite acessar propriedades de CSS via jQuery. Observe o código do exemplo:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset='utf-8' lang='pt-br'>
5     <style>
6       .teste{
7         background:green;
8         font-weight:bold;
9       }
10    </style>
11  </head>
12  <body>
13    <a href="google.com.br">Google</a>
14    <button id="b1">Visualizar</button>
15    <button id="b2">Esconder</button>
16
17    <script src="jquery-3.5.1.min.js"></script>
18    <script>
19      $(document).ready(function() {
20        $("#b1").click(function(event) {
21          $("#a").css("display","block")
22        });
23        $("#b2").click(function(event) {
24          $("#a").css("display","none")
25        });
26      });
27    </script>
28  </body>
29 </html>
```

Esconde e
visualiza o
elemento a

jQuery / Ajax

O jQuery oferece recursos de efeitos sobre os elementos HTML.
Um exemplo (do site do jQuery)

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>animate demo</title>
6   <style>
7     div {
8       background-color: #bca;
9       width: 100px;
10      border: 1px solid green;
11    }
12  </style>
13  <script src="https://code.jquery.com/jquery-3.5.0.js"></script>
14 </head>
15 <body>
16
17  <button id="go">&raquo; Run</button>
18  <div id="block">Hello!</div>
19
20  <script>
21    // Using multiple unit types within one animation.
22
23    $( "#go" ).click(function() {
24      $( "#block" ).animate({
25        width: "70%",
26        opacity: 0.4,
27        marginLeft: "0.6in",
28        fontSize: "32px",
29        borderWidth: "10px"
30      }, 1500 );
31    });
32  </script>
33
34 </body>
35 </html>
```



jQuery / Ajax

A biblioteca jQuery oferece recurso para carregar conteúdo para uma determinada página.

Isso é possível com o uso de **load()**, que possui a seguinte sintaxe:

load(url [, parâmetros][,callback]), sendo que os elementos entre colchetes são opcionais.

Onde:

url : A url que é solicitada para requisição

Parâmetros : Um objeto com os respectivos campos e valores, usando POST como padrão de envio

Callback : Uma função que é ativada logo após a realização da chamada

jQuery / Ajax

Considere o seguinte exemplo:

A página **usandoajaxload.html** carrega o conteúdo da página **paginaLoad.html** quando o botão **btnGo** é clicado. O alert **“preparando dados”** será apresentado logo após a chamada do **load()**.

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>animate demo</title>
6   <script src="https://code.jquery.com/jquery-3.5.0.js"></script>
7 </head>
8 <body>
9   <div id="conteudo"></div>
10  <button id="btnGo">Carregar</button>
11
12 <script>
13   // Using multiple unit types within one animation.
14   $(document).ready(function() {
15     $("#btnGo").click(function(event) {
16       $("#conteudo").load('paginaLoad.html',aviso());
17     });
18   });
19   function aviso(){
20     alert('conteúdo carregado');
21   }
22 </script>
23
24 </body>
25 </html>
```

Página **usandoajaxload.html**

```
1 <h1>Página Carregada</h1>
2 <p>Exemplo de uso do Load do jQuery</p>
```

Página **paginaLoad.html**

jQuery / Ajax

Para que o carregamento do conteúdo ocorra, é necessário utilizar um container web. Para isso, instale o WampServer (Wamp) ou Xampp, os quais utilizam o Apache como container web

No caso do WampServer, copie a pasta do seu projeto para o diretório www do Wamp

No caso do Xampp, copie a pasta do seu projeto para o diretório htdocs do Xampp

Geralmente, para esses dois servidores, informe a url:

<http://localhost:80/sua-pagina-html>

Assim, ao executar a página **usandoajaxload.html**, tem-se:

1

2

3

Página Carregada

Exemplo de uso do Load do jQuery

jQuery / Ajax

Outro Exemplo:

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4   <title>Ajax com jQuery</title>
5
6   <style type="text/css">
7     #box {border:1px solid #ccc; padding:5px}
8   </style>
9
10 </head>
11 <body>
12   <p><input type="button" value="Editar" /></p>
13   <p><input type="button" value="Novo" /></p>
14   <p><input type="button" value="Excluir" /></p>
15   <div id="box"></div>
16
17   <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js" type="text/javascript"></script>
18
19   <script>
20     $(document).ready(function() {
21       $("input[type=button]").click(function(event) {
22         var acao = $(this).attr("value");
23         if (acao == "Editar")
24           $("#box").load("loadEditar.html");
25         if (acao == "Novo")
26           $("#box").load("loadNovo.html");
27         if (acao == "Excluir")
28           $("#box").load("loadExcluir.html");
29       });
30     });
31   </script>
32
33 </body>
34 </html>
```

```
1 <h1>Excluir Dados</h1>
```

```
1 <h1>Editar Dados</h1>
```

```
1 <h1>Novos Dados</h1>
```

localhost/jquery/load01.html

Editar

Novo

Excluir

Excluir Dados

localhost/jquery/load01.html

Editar

Novo

Excluir

Editar Dados

localhost/jquery/load01.html

Editar

Novo

Excluir

Novos Dados

jQuery / Ajax

Complementos

O uso de `load` é interessante quando se deseja carregar uma página dentro de um elemento HTML, principalmente em um container **div**.

Para o envio e recebimento de informações via jQuery, o melhor é utilizar o **\$.ajax()**, que possui a seguinte sintaxe:

```
1 $.ajax(  
2 {  
3   url : "url_destino",  
4   type : 'get',  
5   data : {  
6     //estrutura no formato campo1 : valor1,...campoN : valorN  
7   },  
8   beforeSend : function()  
9   {  
10    //comandos-1  
11  },  
12  success: function(parâmetro1)  
13  {  
14    //comandos-2  
15  },  
16  error: function(parâmetro2)  
17  {  
18    //comandos-3  
19  }  
20 }  
21 );
```

Type pode ter **get** ou **post**.
O **beforeSend** executa a função logo antes de processar os dados.
A seção **success** executa a função caso o processamento seja bem sucedido.
A função de **error** será executada caso o processo apresente algum erro. Os parâmetros **parâmetro1** e **parâmetro2** são retornados para cada uma das respectivas funções

jQuery / Ajax

Complementos

Exemplo de \$.ajax()

```
$.excluir = function (msg) {  
    let codigo=msg;  
  
    $.ajax(  
        {  
            url : "http://localhost:8282/SistemaEstoque/servletexcluirporid",  
            type : 'get',  
            data : {  
                id:codigo  
            },  
            beforeSend : function(){  
                $("#mensagem").html("Analisando a solicitação...");  
            },  
            success: function(msg){  
                $("#mensagem").html(" ");  
                $.mostraMensagem(msg);  
                window.location.reload(true);  
            },  
            error: function(msg){  
                $("#mensagem").html(" ");  
                $.mostraMensagem(msg);  
            }  
        }  
    );  
};
```



jQuery / Ajax

- Conteúdo interessante para melhorar o seu conhecimento:
- <http://www.w3bai.com/pt/>
- <https://www.gocache.com.br/jquery/aprendendo-a-usar-jquery-exemplos/>



jQuery / Ajax

FIM