

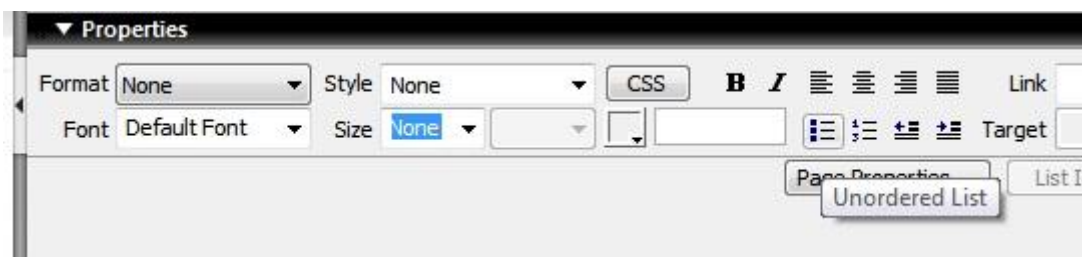
1º Vamos iniciar o desenvolvimento da nossa página, após abrir um documento html com o corpo pronto, adicionamos entre o corpo inicio do body e fim as Divs container, topo, menu, conteúdo e rodapé. Trabalhar com os Divs faz com que passe a ter estrutura o nosso sistema. Como adicionar então?

Trocamos o modo de Common para Layout, clicamos então em Insert Div Tag.



A 1ª Class chamará **container**, responsável por todo conteúdo do sistema. Dentro do corpo container, terá um outro div, com a class **topo**. Em baixo de topo teremos **menu**. A baixo do menu teremos **conteudo**, responsável por armazenar o conteúdo. E logo após o fim do corpo container adicionamos a class **rodape** como ultimo div.

2º No menu iremos por um ou lista como é chamada, como fazer? Modificamos o modo de edição para design, clicamos em *Content for class "menu" Goes Here*, em propriedades clicamos em Unordered list.



Apagamos a linha e digitamos Principal, será nossa primeira lista já incluindo também o link principal. Como fazer? Seleccionamos a palavra Principal e digitamos principal no campo link em propriedades. Alteramos para o modo código e vamos adicionar os demais itens do nosso menu. Copiamos e colamos 5 vezes a linha `Principal`. Por fim alteramos:

Serão eles após a linha home:

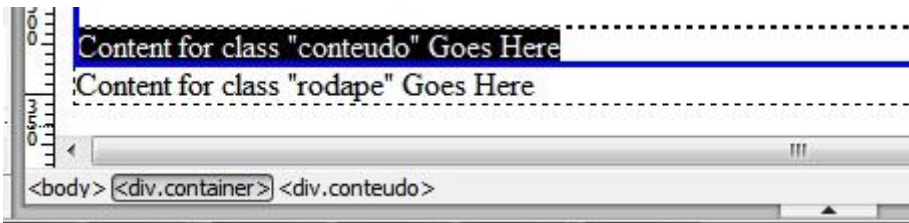
- link **jquery**, nome **jQuery**
- link **apis**, nome **Api's**
- link **download**, nome **Downloads**
- link **tutorial**, nome **Tutoriais**
- link **contato**, nome **Contatos**

3º Alteramos agora para o modo de edição design e vamos alimentar o topo.

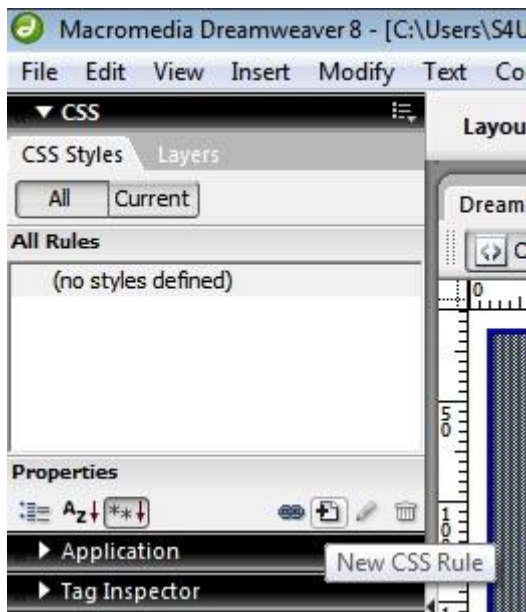
No topo, jogamos uma imagem, como?

Em files navegamos até a nossa pasta conteúdo e arrastamos o arquivo topo.gif para a linha *Content for class "topo" Goes Here*.

4º Vamos agora aplicar uma css, clicando na div.container:



Clicamos agora no menu CSS, depois em new CSS rule, o nome será **.container** mesmo e clicamos em this document only.



No menu background usaremos a cor **#FFFFFF**, no menu border em Style Top adicionamos **solid**, width **5**, e cor: **#99CCCC**.

No menu box vamos setar **780** em Width como padrão, **10** em padding e em margin **auto** para localizarmos o container no centro da tela de acordo com o monitor do usuário.

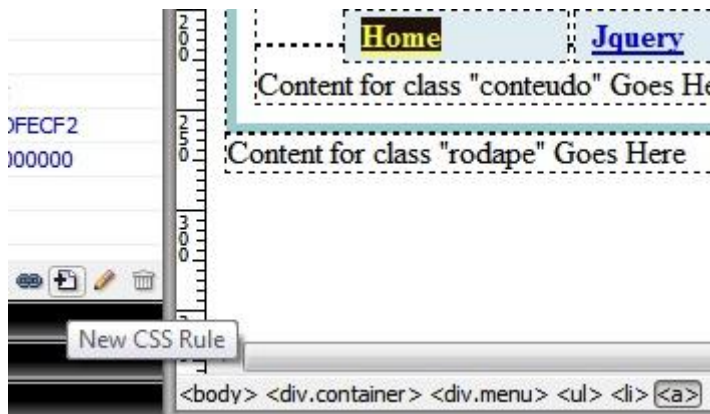
Agora selecionamos o li, selecionando apenas uma linha ou clicando em qualquer li, dará no mesmo resultado.



Adicionamos um **New CSS Rule**, retira o **.container** do nome e deixar apenas **.menu li**.

Vamos mexer na estrutura então, o objeto vai se comportar como um bloco então clicamos no menu Block e em Display escolhemos **block**. No menu box colocamos **2** em margem para definir os espaços entre os menus, em float **left** para que fique um ao lado do outro, e width largura **100**, em Pedding colocamos **3** para definirmos a altura padrão (desmarcamos **same for all**) em left espaço lateral de **10** pixels. Definimos agora no menu type a cor da fonte **#000000** ou preto, definimos negrito em Weight - **bold**, e por fim no menu background a cor de fundo do menu **#DFECF2**.

5º Vamos definir agora o link, selecionamos então o **<a>** para fazermos o CSS, como? Selecionando **Home** do menu aparecerá o **<a>** logo a baixo na barra, clicamos então em new CSS rule.



Renomeamos o container apenas para **.menu a**, em background coloque a mesma cor **#DFECF2**, no menu block, display optamos por **block**, no menu Type em Weight escolhemos **bold** para negrito, e em Color **preto**, marcamos **none** em decoration para não sublinhar.

6º Vamos fazer agora o loader do menu: Alteramos para o modo código, copiamos o **menu.a** que acabamos de criar e colamos logo abaixo do mesmo, adicionamos ao nome **.menu a: hover** ficando assim (**.menu a: hover**) que resultará ao passar o mouse sobre o link nós teremos a animação no botão. Resultado:

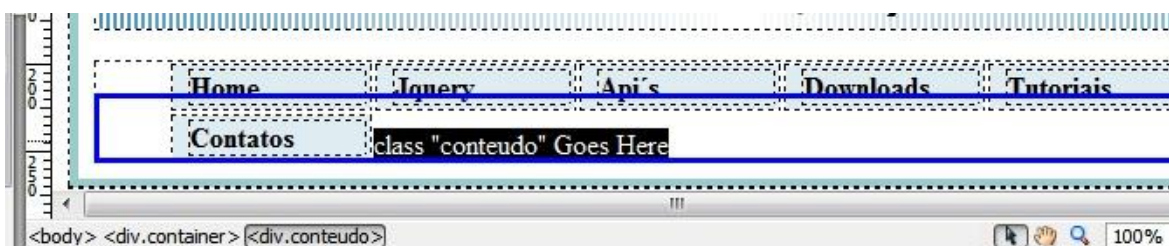
```
...  
.menu a {  
    background-color: #DFECF2;  
    display: block;  
    font-weight: bold;  
    color: #000000;  
    text-decoration: none;  
}  
.menu a: hover {  
    background-color: #DFECF2;  
    display: block;  
    font-weight: bold;  
    color: #000000;  
    text-decoration: none;  
}  
...
```

Voltamos para o modo design, para editar o **.menu a: hover**:



Modificamos o background para a cor **#99CCCC**, no menu box em Width colocamos **100**, em Padding top digitamos **3**, desmarcamos **same for all**, e em left **10**. Em Margin será o oposto do Padding para que não estoure o menu ao passar o mouse sobre o botão, então em top **-2**, desmarcamos também **Same for all**, e digitamos em Left **-10**, para que aconteça uma movie mix bem leve ou seja um efeito bem leve e bonito.

7º Vamos alimentar agora o conteúdo do site, onde o mesmo vai ser responsável por alimentar o conteúdo automaticamente.



Adicionamos uma nova CSS clicando em **New CSS Rule** e alteramos o nome apenas para **.conteudo**, no menu border em style escolhemos **solid** e Width **3**, a cor da borda será **branco**, no menu block escolhemos display **none**.

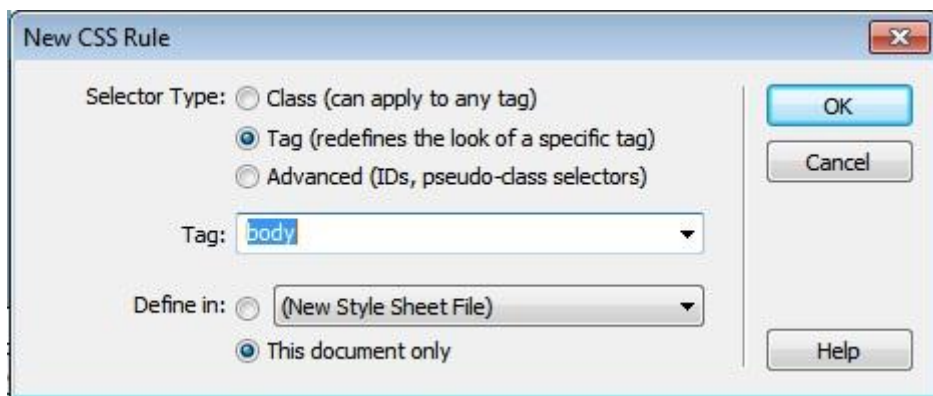
8º Vamos agora setar o rodapé. Criaremos então mais uma CSS, então clicamos no campo **rodape** e depois em **New CSS Rule** nomeando apenas como **.rodape**.

No menu background em color usamos a tonalidade **#FFFFFF**, no menu Block em text align setamos **center**, ele se comportará como um bloco, então vamos em display block, no menu Box colocamos uma largura de **780** em Width, em Padding os espaços internos será de **10** pixels, e em margin o topo de **5** para não ficar colado com o container, então desmarcamos a opção **same for all** em Right e Left escolhemos **Auto** para que assim fique centralizado na tela, igualmente ao container. No menu Border em Style será **solid**, em Width **5** e cor a mesma do container **#99CCCC**. Digitamos então **"PIBID - Programa Institucional de Bolsa de Iniciação à Docência"** no rodapé.

9º Agora selecionamos o elemento carregando, criando assim a CSS também para carregando. Do mesmo jeito do outros CSS, vamos em **New CSS Rule**, renomeamos apenas para **.carregando img**.

No menu Block, o alinhamento do texto será centralizado. No menu Background terá um fundo branco. No menu Borda Style:Solid e Espessura:2 com a cor padronizada do menu. Vamos ocultar ele da mesma forma que realizamos com o conteúdo, no menu Block, Display:none. Porque enquanto o div não estiver carregado ele será exibido, após estiver carregado ele será ocultado o limite. Porque esse div será exibido quando o conteúdo for carregado.

Vamos fazer mais uma css dessa vez do tipo **tag** nomeando **body** para que possamos definir o fundo da nossa página.



No menu background, vamos adicionar um fundo cinza escuro **#333333**.

O que devemos fazer para corrigir o erro do menu: Alteramos para o modo código, e adicionaremos um **br** entre o **** e o **</div>**.

```

<div class="menu">
  <ul>
    <li> <a href="home" class="menu">Home</a></li>
    <li><a href="jquery">Jquery</a></li>
    <li><a href="apis">Api's</a></li>
    <li><a href="download">Downloads</a></li>
    <li><a href="tutorial">Tutoriais</a></li>
    <li><a href="contato">Contatos</a></li>
  </ul>
  |
</div>

```

Digitamos então a linha `<br class="cancelar"/>`.

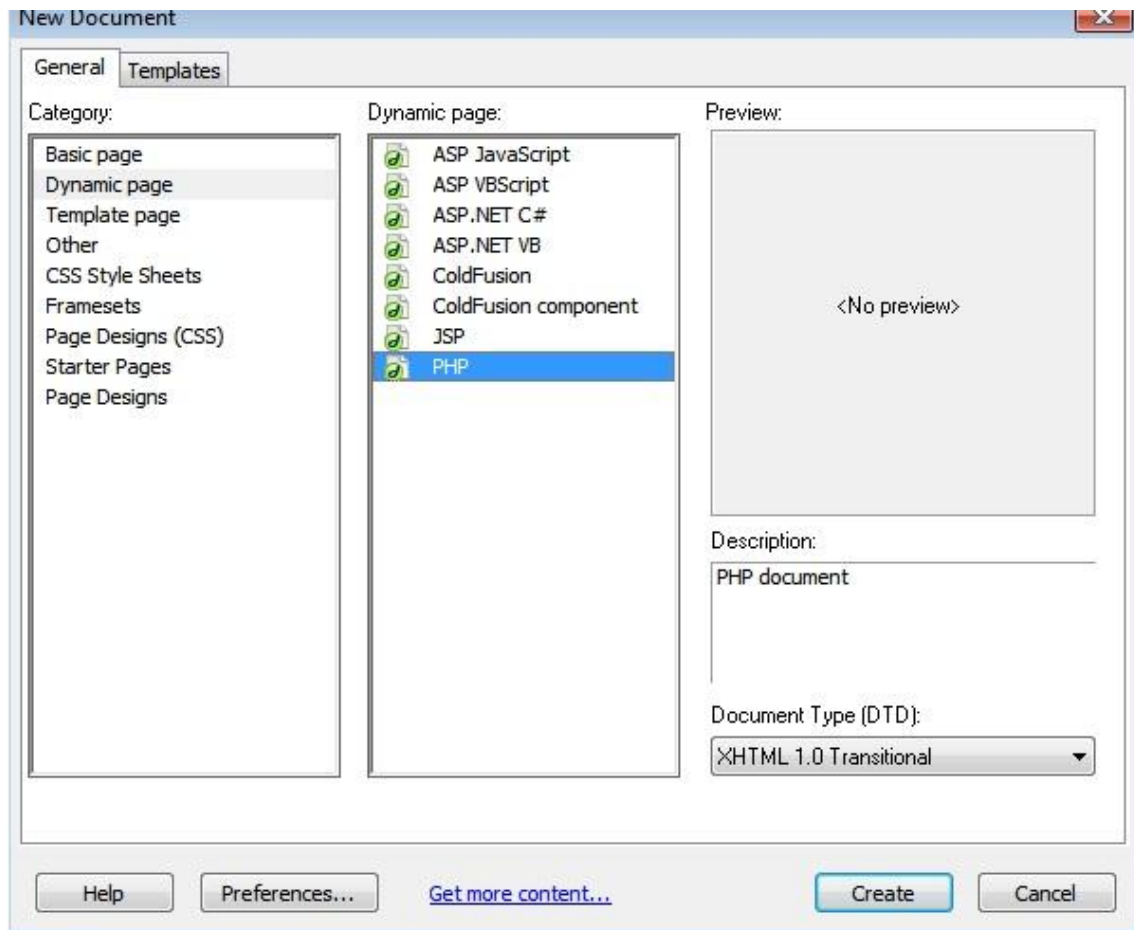
Agora podemos criar a class cancelar selecionando apenas (**class="cancelar"**) clicamos em **new CSS rule**, nomeando apenas **.cancelar**. Setamos apenas no menu Box em clear a opção **both** ou seja ela vai cancelar o left superior e inferior que houver.

Agora vamos selecionar o nosso menu clicando em `div.menu`. Que é o container onde o li se encontra.



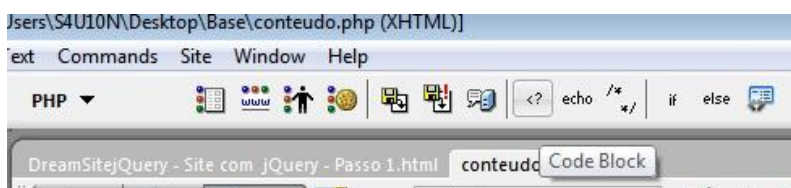
Clicamos em **new CSS rule** e renomeamos apenas para **.menu**. Vamos setar uma altura para esse menu, no menu Box Height digitamos **35** de altura (podendo assim prevenir algumas diferenças no posicionamento do menu da li dentro do nosso layout). Em Margin Top escolhemos **auto** desmarcamos a opção **Same for all** e digitamos **5** em Top e em bottom também **5**, para dar um espaço correto do menu.

10º Vamos criar agora nossa página dinâmica que será em php, crie um novo documento e salve como php.



Essa página conteúdo será requisitada através do ajax, salvando então como **conteudo.php**.

Passos: Modificamos o bloco de **common** ou **layout** para **php** e clicamos no menu botão **code block** adicionando assim o bloco php.



Vamos adicionar o nome da variável chamada **\$pagina=** e vamos pegar o **get** que será postado através de um link, então o get ele pega a variável da url postada que terá o nome de **pag**, resultando na linha seguinte:

```
$pagina= $_GET['pag'];
```

Vamos criar um outro bloco de php logo abaixo, com a condição, se a página requisitada for idêntico a pagina home, ou seja, home da nossa pagina libere a pagina para o usuário. Como então fazer? Vamos então digitar:

```
if($pagina=="principal"){ ?>
```

Dentro da condição serão adicionados os códigos html, vamos adicionar novamente outro bloco php para fechar a condição, com seguinte linha:

```
<?php }?>
```

Agora vamos copiar e colar logo abaixo, como é uma condição vamos alterar apenas as condições, fazendo o seguinte, na nossa pagina principal que estávamos desenvolvendo, copiamos o nome do link de cada botão no caso de **jquery** - *jquery*, **Api's** - *apis*, **Downloads** - *download*, **Tutoriais** - *tutorial* e **Contatos** - *contato* que não são ainda uma variável mais serão no futuro através do jQuery.

Colamos então na pagina php, substituindo pelas cinco condições principal, ficando nossa estrutura assim:

```
<?php $pagina= $_GET['pag'];?>  
<?php if($pagina=="principal"){?>
```

```
//codigos html
```

```
<?php }?>
```

```
<?php if($pagina=="jquery"){?>
```

```
//codigos html
```

```
<?php }?>
```

```
<?php if($pagina=="apis"){?>
```

```
//codigos html
```

```
<?php }?>
```

```
<?php if($pagina=="download"){?>
```

```
//codigos html
```

```
<?php }?>
```

```
<?php if($pagina=="tutorial"){?>
```

```
//codigos html
```

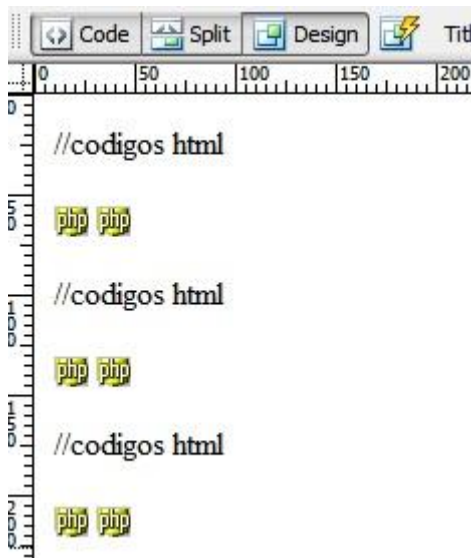
```
<?php }?>
```

```
<?php if($pagina=="contato"){?>
```

```
//codigos html
```

```
<?php }?>
```

Alteramos agora para o modo de edição design e organizarmos nossa pagina.



Após a organização podemos já substituir **//codigos html** pelo conteúdo de cada página.

Vamos agora limpar o código, pois vai ser exibido apenas um bloco por condição, apagando assim as linhas que contiver **<p>**, **</p>** e **<p> </p>**.

Após criar o bloco de PHP condicional:

Vamos testar e entender como será visualizado nossa home page através das condicionais, ao salvarmos, apertamos F12 para testar nossa página. Ao final do link da página vamos adicionar **?pag=principal**.

11º Vamos arredondar agora os cantos dos divs, para dar uma aparência melhor ao nosso layout. Embora esse plug-in não arredonda imagens, vamos trabalhar com um método que arredonde nesse caso vamos utilizar o jQuery. Como fazer?

Para isso vamos para o modo código da nossa pagina, copiamos a linha do conteúdo jQuery logo no inicio do nosso código e colamos logo abaixo do mesmo, ficando assim:

```
<script type="text/javascript" src="js/jquery-1.2.2.js"></script>
<script type="text/javascript" src="js/jquery-1.2.2.js"></script>
<script type="text/javascript" src="js/jquery-1.2.2.js"></script>
<script type="text/javascript">
```

Selecionamos a parte do conteúdo copiado (**js/jquery-1.2.2.js**) e clicamos com o botão direito sobre o mesmo, opção **Code Hint Tools - URL browser**, em browser selecionar o plug-in **arredonda_excanvas.js**, logo após vamos selecionar o mesmo conteúdo da linha a baixo e realizar o mesmo procedimento, dessa vez em browser vamos escolher o arquivo **arredondar_cantos.js**.

Agora que temos todos os arquivos (nosso framework jQuery e os plug-ins), vamos iniciar nosso trabalho com jQuery, setamos então da seguinte forma, abaixo a linha:

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

```

3 <head>
4 <script type="text/javascript" src="js/jquery-1.2.2.js"></script>
5 <script type="text/javascript" src="js/arredonda_excanvas.js"></script>
6 <script type="text/javascript" src="js/arredonda_cantos.js"></script>
7 <script type="text/javascript">
8 //início da sintaxe acima com head e script (responsável pela biblioteca jQuery
9
10
11 $(document).ready(function(){ // encarregado de encontrar elemento do html no (
12 um construtor) OBS: pode gerar conflito pois outras bibliotecas JavaScript tam
13 construtor, então uma solução seria usar o jQuery() no lugar do $().
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

digitamos:

```
$("#div.container").corner("30pxtop")
```

Vamos aplicar abaixo o mesmo conceito, copiamos então e colamos a linha de código e modificamos a linha para essa:

```
$("#div.topo").corner("20px")
```

Vamos trabalhar agora em menu onde existem as li, com a seguinte linha:

```
$("#div.menu li").corner("10px")
```

Agora o rodapé:

```
$("#div.rodape").corner("30pxbottom")
```

Onde **bottom** refere-se aos cantos inferiores do rodapé.

12º Vamos trabalhar agora dentro do campo definido logo no início:

```
//início do conteúdo home
```

```
//fim do conteúdo home
```

Setamos agora um padrão, assim que a página for carregada, vai setar o padrão de conteúdo que vamos postar aqui através do GET com a seguinte linha:

```
$.get("conteudo.php",
```

```
//O get vai para a página conteudo.php
```

```
{pag:"home"}, function(conteudo){
```

```
//passando o parametro pag=home e retornando uma função conteúdo
```

```
$(".conteudo").html(conteudo).show("slow")  
})
```

```
//o conteudo.php será exibido dentro do container através do efeito “slow”
```

Vamos trabalhar dentro do campo do conteúdo ajax:

```
//início do conteúdo ajax
```

```
//fim do conteúdo ajax
```

Digitamos:

```
$("#div.menu li a").click (function(){
```

```
//setamos o click..
```

```
link=$(this).attr("href")
```

```
//através da variável
```

```
$.ajax({method:"get",
```

```
//carregamento ajax através de get
```

```
url: "conteudo.php",
```

```
//carregando a url conteudo.php
```

```
data: "pag="+link,
```

```
//passando parâmetros do link
```

```
beforeSend: function(){
```

```
// beforeSend: antes de enviar execute a função exibir o carregando
```

```
$("#div.carregando").show()
```

```
// show div.carregando: para exibir o div oculto (carregando)
```

```
$(".conteudo").slideUp("slow");
```

```
// slideUp .conteudo: ocultar o conteúdo
```

```
complete: function(){
```

```
//quando a requisição estiver completa...
```

```
    $("div.carregando").slideUp("slow")),  
  
    //o div carregando ficará oculto através do slideUp  
  
    success: function(html){  
  
    //quando estiver tudo suficiente ou carregado,  
  
        $(".conteudo").html(html).show("slow")  
  
    //passa a variável interna html preenchendo e exibindo o conteúdo através da variável interna  
    //retornando o false no link, logo a baixo  
  
        })  
  
    //fim do $.ajax{  
  
        return false  
  
    //invalidamos o link clicado  
  
        })  
  
    //fim do $("div.menu li a").click  
  
        })
```