Conteúdo

[:: Modelo CSS das caixas - Box Model :: 2](#_Toc323793399)

**[Um código para nosso tutorial](#_Toc323793400)** [2](#_Toc323793400)

**[Introdução - entendendo como CSS manipula os elementos HTML](#_Toc323793401)** [3](#_Toc323793401)

**[Caixas CSS - entendendo as caixas](#_Toc323793402)** [4](#_Toc323793402)

[:: Modelo CSS das caixas - Box Model :: 6](#_Toc323793403)

**[Características de cada uma das quatro áreas da caixa](#_Toc323793404)** [6](#_Toc323793404)

**[As regras CSS para cada uma das quatro áreas da caixa](#_Toc323793405)** [7](#_Toc323793405)

**[As caixas "bloco" e as caixas "inline"](#_Toc323793406)** [7](#_Toc323793406)

**[Extensão do conceito de caixas](#_Toc323793407)** [8](#_Toc323793407)

[:: Modelo CSS das caixas - Box Model :: 10](#_Toc323793408)

**[Os blocos de conteúdo](#_Toc323793409)** [10](#_Toc323793409)

[:: Posicionar um objeto ou página no centro da tela :: 14](#_Toc323793410)

**[Introdução](#_Toc323793411)** [14](#_Toc323793411)

**[Posicionando uma imagem](#_Toc323793412)** [14](#_Toc323793412)

**[As regras CSS para posicionamento.](#_Toc323793413)** [14](#_Toc323793413)

**[Posicionando no centro da tela](#_Toc323793414)** [17](#_Toc323793414)

**[Aplicando a técnica para uma página web](#_Toc323793415)** [17](#_Toc323793415)

**[Conclusões](#_Toc323793416)** [18](#_Toc323793416)

[:: Posicionando relativamente o absoluto 18](#_Toc323793417)

**[O que há com os](#_Toc323793418)**[DIVs](#_Toc323793418)**[aninhados ?](#_Toc323793418)** [18](#_Toc323793418)

**[Blocos de conteúdos (](#_Toc323793419)*[Contexto de posicionamento](#_Toc323793419)*[)](#_Toc323793419)** [19](#_Toc323793419)

**[Posicionamento dinâmico](#_Toc323793420)** [20](#_Toc323793420)

**[O limite inferior da barra de navegação](#_Toc323793421)** [21](#_Toc323793421)

[Diferença entre position: static, relative, absolute e fixed 22](#_Toc323793422)

**[position: static](#_Toc323793423)** [22](#_Toc323793423)

**[position: relative](#_Toc323793424)** [23](#_Toc323793424)

**[position: absolute](#_Toc323793425)** [23](#_Toc323793425)

**[position:relative + position: absolute](#_Toc323793426)** [23](#_Toc323793426)

**[position: fixed](#_Toc323793427)** [24](#_Toc323793427)

:: Modelo CSS das caixas - Box Model ::

Informação importante:

Se você pretende projetar uma página, um documento ou um site inteiro baseado em CSS e sem uso de tabelas complexas o perfeito entendimento deste tutorial vai fornecer-lhe os conceitos básicos de posicionamento CSS.

Este assunto iniciará você nas técnicas avançadas de posicionamento CSS e em consequência para seu perfeito entendimento é necessário que você domine com destreza o uso das propriedades CSS.

**Um código para nosso tutorial**

Para melhor visualização e exemplificação dos conceitos contidos neste tutorial vamos criar uma página web bem simples constituída de:

* um cabeçalho título;
* um cabeçalho subtítulo;
* dois parágrafos;

Abaixo o código completo da página, em [XHTML 1.0](http://www.maujor.com/tutorial/xhtml.php" \o "Tutorial XHTML) modo Transitional:

[view source](http://www.maujor.com/tutorial/csscaixa.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/csscaixa.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/csscaixa.php" \l "about" \o "?)

01.<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0

02.Transitional//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1->

03.transitional.dtd">

04.<**html** xmlns="<http://www.w3.org/1999/xhtml>">

05.<**head**>

06.<**meta** http-equiv="Content-Type" content="text/html;

07.charset=iso-8859-1" />

08.<**title**>Box Model</**title**>

09.</**head**>

10.<**body**>

11.

12.<**h1**>Folhas de estilo em cascata</**h1**>

13.

14.<**h3**>A importância do Box Model</**h3**>

15.

16.<**p**>

17.Para projetar um layout CSS, seja para uma página

18.web ou um site inteiro é fundamental que o

19.desenvolvedor entenda com detalhes como as folhas

20.de estilo em cascata tratam os elementos HTML< **para** fins

21.de apresentação em uma mídia do usuário

22.por exemplo: a tela de um monitor).

23.</p>

24.

25.<**p**>

26.Todo elemento HTML é uma "caixa" retangular a ser

27.apresentada na tela com as estilizações determinadas

28.pelas regras CSS. As caixas sõo empilhadas uma após a

29.outra e constituídas de margens, bordas, espaçamentos

30.e o conteúdo propriamente dito.

31.</**p**>

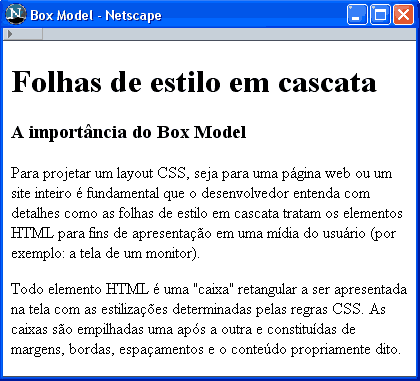
32.

33.</**body**>

34.</**html**>

E na figura a seguir uma visualização da página renderizada no browser.

Notar o título em um cabeçalho <h1>, o subtítulo em um cabeçalho <h3> e dois parágrafos <p>



Você certamente ao olhar para a figura mostrada não tem qualquer dificuldade em entender e "ver" dois cabeçalhos e dois parágrafos em uma sequência igual a do código HTML.

Mas, para facilitar o entendimento do Box Model, olhe novamente para a figura e "veja-a" como se fossem 04 (quatro) caixas empilhadas.

Em cima <h1>, depois <h3>, a seguir <p> e em baixo de todas outro <p>.

**Introdução - entendendo como CSS manipula os elementos HTML**

As CSS para **montar o layout** de um documento, consideram e tratam todos os elementos HTML como se fossem caixas (ou mais propriamente em duas dimensões: um retângulo) a serem exibidas em uma mídia visual (a tela do monitor ou uma folha de papel impressa por exemplo).

As caixas são formatadas, estilizadas e dispostas pelas CSS segundo uma seqüência, na qual cada elementoHTML ocupa uma caixa.

Desta forma, cabeçalhos (<h1>,<h2>,<h3>..), parágrafos (<p>), listas (<ul>, <ol>), formulários (<form>), divisões (<div>), e em fim qualquer elemento HTML (tag) é representado por uma caixa. Podemos então dizer que a caixa assim como foi definida é a **unidade básica de formatação CSS**. As caixas podem conter ou estar contidas dentro de outras caixas.

As técnicas de formatação são simples, contudo o projetista web deve conhecê-las a fundo para bem manipular suas regras.

**Caixas CSS - entendendo as caixas**

As caixas CSS são constituídas por 04 (quatro) áreas retangulares, listadas abaixo:

1. conteúdo;
2. espaçamentos (padding);
3. bordas (border);
4. margens (margin).

que se desenvolvem de dentro para fora, na ordem listada acima.

Uma caixa consiste primeiramente do seu **conteúdo** que é aquilo ou o que vai ser exibido.

A caixa pode conter 4 **bordas,**que vem a ser uma linha decorativa em volta da caixa.

A distância entre a borda e o conteúdo é o**espaçamento**.

A distância que separa uma caixa da outra é a**margem.**

Vamos ao nosso código XHTML inicial e acrescentemos as regras CSS abaixo:

[view source](http://www.maujor.com/tutorial/csscaixa.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/csscaixa.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/csscaixa.php" \l "about" \o "?)

01.h1 {

02.**background-color**: #cc9;

03.**border**:10px solid #f00;

04.**padding**: 5px;

05.}

06.

07.h3 {

08.**background-color**: #fc9;

09.**border**:20px solid #039;

10.**padding**: 15px;

11.}

12.

13.p.um {

14.**background-color**: #ff9;

15.**border**:5px solid #f0f;

16.**padding**: 5px;

17.**text-align**:justify;

18.}

19.

20.p.dois {

21.**background-color**: #cff;

22.**border**:2px solid #039;

23.**padding**: 10px;

24.**text-align**:justify;

25.}

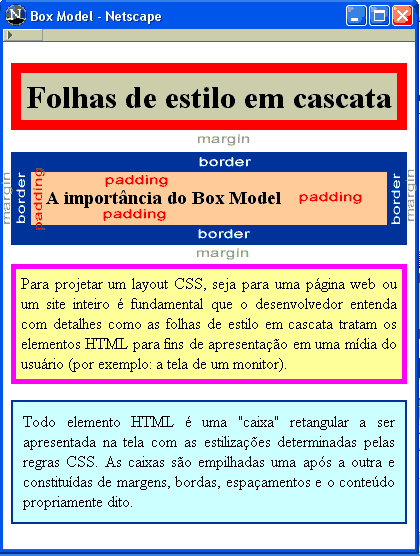
26.

27.Nota: As classes .um e .dois foram aplicadas

28.nos primeiro e segundo parágrafos

Eis as quatro caixas agora coloridas e bem visíveis.

Nota: na caixa que representa o elemento <h3> nomeei as quatro áreas da caixa: **margin, border, padding e conteúdo,**para facilitar a visualização daquelas áreas que compõem as caixas



:: Modelo CSS das caixas - Box Model ::

**Características de cada uma das quatro áreas da caixa**

Agora que você já conhece a anatomia de uma caixa, vamos mostrar quais as peculiaridades de cada área que a compõe.

1. As**margens (**margin**)** são sempre transparentes e cada elemento HTML tem uma margem default determinada pelo browser, isto é embora as especificações CSS preconizem uma margem default 0 (zero) para todos os elementos HTML, cada fabricante de browser adota um valor default para a margem. Você com certeza já notou que em um documento HTML a distância entre dois cabeçalhos, entre dois parágrafos, entre um cabeçalho e um parágrafo, é diferente. Estas distâncias são as margens daqueles elementos.
2. As **bordas (**border**)**são transparentes e de espessuras iguais a zero por default;
3. Os**espaçamentos (**padding**)** são transparentes e de espessuras iguais a zero por default;
4. O fundo (background) de uma caixa está por trás do conteúdo e do espaçamento.

**As regras CSS para cada uma das quatro áreas da caixa**

Com regras CSS você pode controlar a aparência e dimensões de cada uma das quatro áreas que compõem a caixa.

1. As espessuras das margens, das bordas e dos espaçamentos podem ser definidas com valores diferentes para cada um dos quatro lados da caixa;
2. Se as espessuras forem definidas com valor 0 (zero) elas não serão exibidas;
3. As cores e o tipo das margens podem ser definidas com valores diferentes para cada um dos quatro lados da caixa;
4. A cor ou uma imagem de fundo pode ser definida para ser colocada por trás do conteúdo e do espaçamento.

Link para os tutoriais das propriedades citadas:

* [Tutorial Margens](http://www.maujor.com/tutorial/margintut.php)
* [Tutorial Bordas](http://www.maujor.com/tutorial/bordertut.php)
* [Tutorial Espaçamentos](http://www.maujor.com/tutorial/paddingtut.php)

**As caixas "bloco" e as caixas "inline"**

Existem dois tipos de caixas: caixas nível de bloco (block-level) e caixas inline a exemplo dos elementos HTML que também podem ser bloco ou inline.

Lembrando o conceito visto em HTML que diz que elementos nível de bloco criam uma quebra de linha (por exemplo <p>), enquanto elementos inline não geram quebra de linha (por exemplo <span>)

Cada elemento HTML cria uma caixa. Assim, um parágrafo (<p>), um cabeçalho (<h1>,< h2>, etc..) ou uma divisão (div) todos eles criam uma caixa quando são exibidos.

Se um elemento está contido dentro de outro elemento, então sua caixa será exibida dentro da caixa do elemento em que esta contido (elemento "pai"). Por exemplo, um parágrafo <p> dentro de uma divisão <div>.

**Extensão do conceito de caixas**

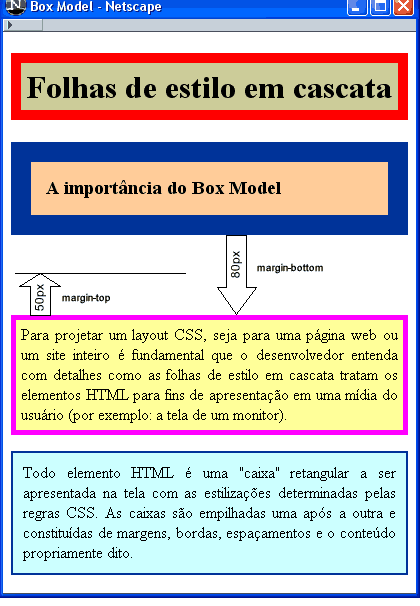
Um bloco de conteúdo é uma área retangular em um documento HTML na qual uma ou mais caixas são colocadas. Caixas normalmente estão confinadas dentro dos limites de um bloco de conteúdo.

O mais importante bloco de conteúdo é a área do navegador onde é exibido o conteúdo da página (viewport).

Se um bloco contém várias caixas, elas são colocadas uma em baixo da outra. Caixas assim colocadas têm suas MARGENS SOBREPOSTAS. Isto significa que se duas caixas são colocadas uma após a outra, a margem inferior da caixa de cima se sobrepõe a margem superior da caixa de baixo e a margem final entre as caixas será igual a maior delas.

A figura abaixo ilustra este conceito.

Foi definida uma margem inferior de 80 px para <h3> e uma margem superior de 50px para o primeiro parágrafo. Notar que a margem resultante não é de 80px + 50px = 130px e sim 80px



Sobreposição das margens:

A distância ilustrada com 80px representa a margem inferior da caixa de cima ( o elemento <h3>).

A distância ilustrada com 50px representa a margem superior da caixa de baixo ( o elemento <p>).

A margem inferior da primeira caixa (<h3>) é maior do que a margem superior da segunda caixa (<p>), em consequência a margem resultante será de 80px. Veja que as margens não se somam, elas se sobrepõem prevalecendo a maior delas.

:: Modelo CSS das caixas - Box Model ::

**Os blocos de conteúdo**

Já sabemos que cada elemento nível de bloco cria uma caixa. Também já vimos que no centro de cada caixa existe o conteúdo.

Se o elemento contém elementos inline, então estes são exibidos dentro deste conteúdo.

Se ele contém outros elementos nível de bloco estes serão exibidos dentro da caixa "pai", como caixas "filho".

Para exibir um documento HTML o navegador cria uma caixa para o elemento <body> e coloca esta caixa no bloco de conteúdo viewport (a tela do monitor por exemplo).

Esta caixa com suas margens, bordas e espaçamentos é o bloco de conteúdo para todos os elementos nível de bloco que estão contidos no elemento <body>.

As caixas são colocadas uma após a outra em sequência. Se qualquer elemento tem um elemento nível de bloco "filho" este será colocado dentro do elemento "pai", repetindo-se esse processo até que não haja mais elemento para ser exibido.

Para exemplificar este conceito acompanhe o exemplo abaixo.

**O elemento**<body>**contém:**

* **um elemento cabeçalho <h3>;**
* **um elemento <div>;**
  + **um elemento parágrafo 1;**
  + **um elemento parágrafo 2 ;**
* **um elemento cabeçalho <h4>;**
* **um elemento parágrafo 3;**
  + **um elemento <span>;**

Abaixo o código HTML:

<body>

<h3>Conteúdo do cabeçalho</h3>

<div>

<p>Conteúdo do primeiro parágrafo.</p>

<p>Conteúdo do segundo parágrafo.</p>

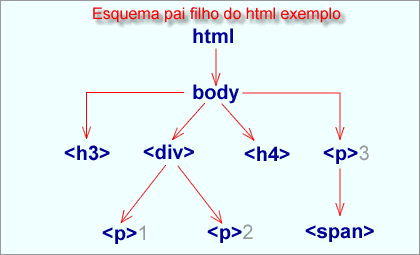
</div>

<h4> Conteúdo do cabeçalho</h4>

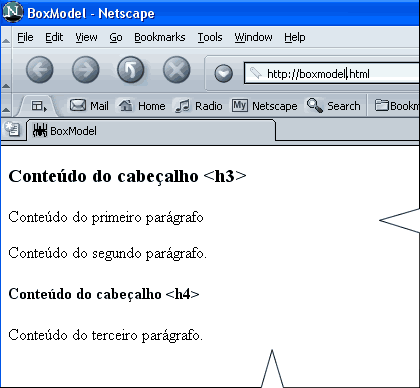
<p>Conteúdo do <span>terceiro</span> parágrafo.<p>

</body>

Observe um esquema do html na figura abaixo:



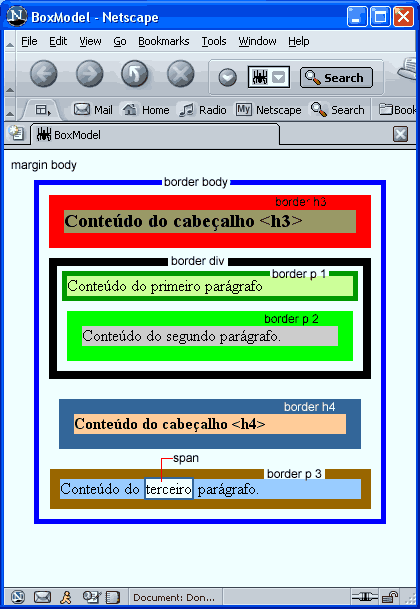
E a seguir como o browser renderiza o código HTML:



**E agora o grande resumo final de tudo que foi estudado nestas três páginas do tutorial.**

Acresentei propriedades CSS de bordas e margens coloridas nos elementos HTML do código, para você visualizar como as caixas se distribuem na aplicação do usuário (no caso o browser). Note caixas dentro de caixas e caixas após caixas como foi explicado no tutorial.

Se você entender com detalhes a figura abaixo, estará em condições de prosseguir com os assuntos de Posicionamento CSS e dar seu primeiro passo para o CSS avançado.



:: Posicionar um objeto ou página no centro da tela ::

**Introdução**

Neste tutorial vou mostrar a técnica CSS para posicionar um objeto ou uma página com altura definida no meio da tela do monitor do usuário, ou seja centrar na horizontal e na vertical, independente da resolução. Ao final da leitura desse tutorial você estará capacitado a posicionar com uso das CSS um objeto em qualquer lugar na tela, do monitor do usuário.

**Posicionando uma imagem**

Vou iniciar mostrando como posicionar uma imagem na tela. Usarei a imagem abaixo que nomeei de logo.gif:



O código HTML que coloca a imagem na tela é mostrado a seguir.

[view source](http://www.maujor.com/tutorial/meio_tela.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/meio_tela.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/meio_tela.php" \l "about" \o "?)

1.<**body**>

2.<**img** src="logo.gif" width="220" height="80" alt="logo do site Maujor" />

3.</**body**>

[Página mostrando a renderização do código acima](http://www.maujor.com/tutorial/imagens_meio_tela/ex1_meiotela.html)

Como você observou, a imagem está posicionada no canto superior esquerdo da tela com as margens superior (aproximadamente 20px) e esquerda (aproximadamente 10px) padrão, variando conforme o navegador usado.

Vamos a seguir atribuir um id para a imagem e estilizar o posicionamento da id. Escolhi o nome **imgpos** para oid da imagem e o HTML agora é:

[view source](http://www.maujor.com/tutorial/meio_tela.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/meio_tela.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/meio_tela.php" \l "about" \o "?)

1.<**body**>

2.<**img** src="logo.gif" width="220" height="80" alt="logo do site Maujor" id="imgpos" />

3.</**body**>

**As regras CSS para posicionamento.**

A propriedade CSS que possibilita posicionar um elemento qualquer (no nosso caso o elemento img que tem a id="imgpos") é a propriedade position complementada pelas propriedades left e top. O valor absolute para a propriedade position quando declarado para um elemento X controla a posição do elemento tomando como origem do sistema de coordenadas o canto superior esquerdo do primeiro box ancestral de X que tenha sido posicionado com os valores fixed, relative ou absolute (Ver o tutorial [Posicionando relativamente o absoluto](http://www.maujor.com/tutorial/absrel.php)). Caso não exista um elemento ancestral posicionado, a origem do sistema de coordenadas será o canto superior esquerdo da tela.

A regra CSS mostrada a seguir posiciona de forma absoluta em relação ao canto superior esquerdo da tela o elemento com id="imgpos" .

[view source](http://www.maujor.com/tutorial/meio_tela.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/meio_tela.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/meio_tela.php" \l "about" \o "?)

1.#imgpos {

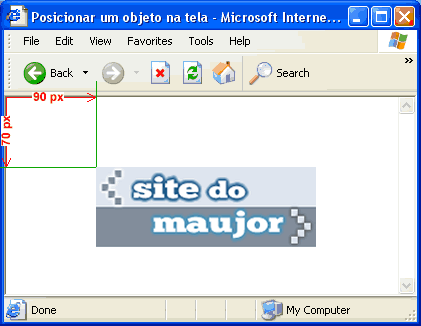
2.**position**: absolute;

3.**left**: 90px; /\* posiciona a 90px para a esquerda \*/

4.**top**: 70px; /\* posiciona a 70px para baixo \*/

5.}

Observe a figura explicativa.



Notar que regra CSS a definição da posição absolute e das coordenadas (left ) que define o quanto a imagem dista da margem esquerda e (top) que define o quanto a imagem dista do topo.

A referência, como já foi dito acima é o canto superior esquerdo da tela e o ponto que ocupa a posição definida é o canto superior esquerdo da figura ou elemento posicionado.

**IMPORTANTE:** Você poderá alterar as referências da tela e consequentemente da figura ou elemento posicionado, mudando as propriedades para right e bottom . E valem as combinações para os quatro cantos da tela.

left e top: referência é o canto superior esquerdo;

left e bottom: referência é o canto inferior esquerdo;

right e top : referência é o canto superior direito;

right e bottom: referência é o canto inferior direito.

Qualquer unidade de medida CSS pode ser usada para definir as coordenadas, inclusive medidas relativas como, por exemplo, porcentagens.

Assim é válido declarar:

[view source](http://www.maujor.com/tutorial/meio_tela.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/meio_tela.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/meio_tela.php" \l "about" \o "?)

1.#imgpos {

2.**position**: absolute;

3.**left**: 50%;

4.**top**: 50%

5.}

**Desafio:**

Você seria capaz de interpretar a regra acima e imaginar qual a posição ocupada pelo logo na tela já que a regra diz: posicione 50% para a esquerda e 50% para baixo na tela.

[Página mostrando a posição ocupada pelo logo](http://www.maujor.com/tutorial/imagens_meio_tela/ex2_meiotela.html)

Era esta a posição que você imaginou? Ou imaginou a figura no centro da tela?

Lembre-se o ponto que está a 50% do topo e 50% para a esquerda é o canto *superior esquerdo* da figura.

A figura não está no centro da tela. Volte lá e confira!

**Posicionando no centro da tela**

Conseguimos posicionar o canto superior esquerdo da figura no meio da tela com os valores declarados em porcentagens. E agora, para que o CENTRO da figura ocupe o meio da tela basta arrastar a figura para a esquerda e para cima de um valor igual a:

**para a esquerda:** metade da largura da figura;

**para cima:** metade da altura da figura.

A figura é um retângulo de 220 x 80 e os valores para arrastar são:

**para a esquerda:**220px/2 = 110px;

**para cima:**80/2 = 40px.

A propriedade CSS que determina o afastamento em volta dos elementos é margin.

Como a figura deve ser arrastada para a esquerda basta declarar uma margin-left **negativa!** E o arrastar para cima consegue-se com uma margin-top negativa!

[Página mostrando o logo no centro da tela.](http://www.maujor.com/tutorial/imagens_meio_tela/ex3_meiotela.html)

E o código completo da página para seus estudos.

[view source](http://www.maujor.com/tutorial/meio_tela.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/meio_tela.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/meio_tela.php" \l "about" \o "?)

01.<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01

02.Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">

03.<**html**>

04.<**head**>

05.<**meta** http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

06.<**title**>Posicionar um objeto na tela</**title**>

07.<**style** type="text/css">

08.#imgpos {

09.position:absolute;

10.left:50%;

11.top:50%;

12.margin-left:-110px;

13.margin-top:-40px;

14.}

15.</**style**>

16.</**head**>

17.<**body**>

18.<**img** src="logo.gif" width="220" height="80"

19.alt="logo de 220 por 80 pixel" id="imgpos">

20.</**body**>

21.</**html**>

**Aplicando a técnica para uma página web**

Você poderá aplicar esta técnica para posicionar uma página inteira no centro da tela. Isto só faz sentido prático quando a página tem altura reduzida e menor do que a altura total da tela. Se você tem uma página de entrada no site do tipo "Clique aqui para entrar" e que normalmente tem um design de dimensões reduzidas, poderá colocá-la no meio da tela com esta técnica.

**Nota: A página que você vai visualizar foi desenvolvida com tabelas à maneira ultrapassada de projetar web sites. Usei um código HTML "ruim" apenas como ilustração da técnica, no entanto, ela é válida para sites desenvolvidos segundo os Padrões Web.**

[Página de código ruim, exemplificando a técnica de centralização](http://www.maujor.com/tutorial/imagens_meio_tela/pg_meio.html)

O código fonte da página, mostra o uso da técnica.

**Conclusões**

Posicionar um elemento no centro da tela é uma tarefa bem simples com uso das CSS. Como você já deve ter concluído, usando esta técnica pode-se posicionar um elemento HTML em qualquer lugar na tela do usuário, tanto usando medidas absolutas como valendo-se de medidas relativas.

:: Posicionando relativamente o absoluto

Autor: [Douglas Bowman](http://www.stopdesign.com/" \o "Site do autor deste artigo)  
URL do original: [http://www.stopdesign.com/articles/absolute/](http://www.stopdesign.com/articles/absolute/" \o "Site onde se encontra o original em ingles)  
Título original: Making the Absolute, Relative  
Traduzido com autorização expressa do autor

**Blocos de conteúdos (*Contexto de posicionamento*)**

Colocar um elemento posicionado de forma absoluta dentro de outro elemento posicionado de forma relativa é uma técnica mágica, frequentemente usada. O mais comum é ouvir-se exatamente o inverso: um elemento posicionado de forma relativa ocupa posição *em volta* de um elemento posicionado absolutamente.

Segundo as especificações das CSS2 um [elemento posicionado de forma absoluta](http://www.w3.org/TR/CSS2/visuren.html" \l "positioning-scheme) ocupa a posição de acordo com o [bloco que o contém](http://www.w3.org/TR/CSS2/visuren.html" \l "containing-block). Qualquer elemento é considerado “*posicionado*” se o valor para posição for relative, absolute, ou fixed (qualquer valor que não seja static). “Static” é um dos possíveis valores para a propriedade position. É também o valor padrão para qualquer elemento que não tenha sua posição explicitamente declarada. Static significa que o posicionamento do elemento não foi modificado e ele estará localizado no [fluxo normal](http://www.w3.org/TR/CSS2/visuren.html" \l "normal-flow) do documento, inserido no contexto normal de seus elementos vizinhos e dos blocos que o contém.

Se um elemento posicionado de forma absoluta não está contido em qualquer elemento posicionado, ou seja, quando nenhum de seus elementos-ancestrais não forem *posicionados*, ele será posicionado em relação aos limites da página (chamada de *bloco de conteúdo inicial*). Assim um elemento estilizado com regras como mostrado a seguir:

[view source](http://www.maujor.com/tutorial/absrel.php" \l "viewSource" \o "view source)

[print](http://www.maujor.com/tutorial/absrel.php" \l "printSource" \o "print)[?](http://www.maujor.com/tutorial/absrel.php" \l "about" \o "?)

1.#topleft {

2.**position**:absolute;

3.**top**:0;

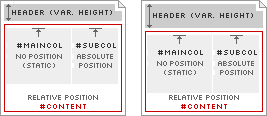
4.**left**:0;

5.}

será posicionado no canto superior esquerdo (top-left) da página. Contudo, se qualquer elemento-ancestral (elemento do qual ele seja descendente) for posicionado, então este ancestral estabelecerá um bloco de conteúdo para todos os seus descendentes. Se um elemento ancestral for *posicionado*, estabelecerá um novo contexto para posicionamento de todos seus descendentes (um novo bloco de conteúdo). Neste caso, os valores de [box offsets](http://www.w3.org/TR/CSS2/visuren.html" \l "position-props) estabelecidos na regra CSS mostrada posicionará o elemento #topleft no canto superior esquerdo do seu ancestral posicionado e não no canto superior esquerdo da página.

**Posicionamento dinâmico**

Para o stopdesign.com, (à época que escrevi este artigo) eu uso um posicionamento absoluto para a *coluna da direita* (#subcol). Quando eu projetei os templates eu queria que a coluna da direita ficasse **logo abaixo**do *topo* (#header) sem espaçamentos ou sobreposições entre os dois blocos. O topo contém o logo e a barra de navegação principal. Como a barra de navegação é constituída de texto, ela poderá ser redimensionada quando o usuário redimensionar os tamanhos de fontes no browser. Isto afeta diretamente a altura do topo. Poderia ter escolhido posicionar a coluna da direita em relação aos limites da página escolhendo um valor *top* para a propriedade do posicionamento, igual a altura do topo. Mas, como o topo varia em altura de acordo com o tamanho de fonte adotado pelo usuário em seu navegador, isto poderia fazer com que a coluna da direita ficasse em cima do topo quando este aumentasse sua altura. Ou um espaçamento poderia surgir quando o topo tivesse sua altura reduzida abaixo da medida estabelecida inicialmente.

*Diagrama 1*

Para fazer as adaptações necessárias devido a esta variação de altura do topo, eu posicionei o elemento de conteúdo (ancestral) (#content). Este elemento aparece na marcação do código imediatamente embaixo do topo e posicionado com position:relative;. Ele se posiciona, no fluxo do documento, sempre abaixo do topo, independentemente da maior ou menor altura assumida pelo topo. Em razão de #content ter sido *posicionado*, ele cria um novo contexto para seus elementos descendentes. A coluna da direita (#subcol) está contida dentro de #content, desta forma ela permanecerá posicionada no canto superior direito de #content, e não no canto superior direito da página. Isto também faz com que a coluna direita se adapte na posição de acordo com a altura assumida pelo topo, como mostrado no *Diagrama 1*.

**O limite inferior da barra de navegação**

Diagram 2-A mostra a navegação com o tamanho padrão para textos. *Diagrama 2-A*  
  
Diagram 2-B mostra a navegação com o tamanho aumentado para textos e mesma altura do topo. *Diagrama 2-B*

Para concluir, voltemos ao site Adaptive Path e sua barra de navegação que demonstra os diferentes usos desta técnica de *fazer o absoluto, relativo*. Eu poderia ter definido um valor fixo para margin-top da barra de navegação e pronto. Mas naquele caso eu queria que o topo tivesse sua altura constante independentemente de como o usuário dimensionasse a fonte de seu browser. Em outras palavras, eu não queria que a borda verde escura embaixo da barra navegação tivesse qualquer deslocamento vertical em relação ao logotipo, como mostrado nos *Diagramas 2-A* and *2-B*.

O texto da barra de navegação é livre para ser redimensionado de acordo com as preferências de tamanho de texto do usuário, mas sua linha de base (NT: a borda verde escura) permanece fixa na posição. O texto da barra de navegação sempre tocará na borda verde escura sem contudo nunca ultrapassá-la. Ao posicionar o elemento#nav usando position:relative; e definindo uma altura fixa (72px), para seu antecessor este estabelecerá um novo contexto de posicionamento para seu DIV “interior”. O DIV interior agora poderá ser absolutamente posicionado em relação a**parte inferior** de seu ancestral #nav usando-se bottom:0;.

Certamente existem outros usos para esta técnica. Não é incomum desejar-se redefinir um contexto para posicionamento de modo a permitir um posicionamento absoluto outro que não em relação aos limites da página. As razões pelas quais eu faço o que faço, frequentemente são óbvias para mim. Mas, devemos nos lembrar que o óbvio para nós nem sempre o é para os outros. Talvez esta técnica seja agora melhor entendida por alguns (talvez mesmo, óbvia)

Diferença entre position: static, relative, absolute e fixed

**position: static**

Esse é o valor default para todo elemento HTML, o que significa que o elemento não tem posicionamento definido e seu fluxo ocorre normalmente no documento HTML.

[view source](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "viewSource" \o "view source)[print](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "printSource" \o "print)[?](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "about" \o "?)

1.#div1 {position: static}

Não é necessário definir position:static para os elementos da página pois esse é o valor default.

**position: relative**

Especificando position:relative agora você pode utilizar top ou bottom, right ou left para posicionar os elementos na página em relação ao lugar que ele ocuparia no fluxo do documento.

[view source](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "viewSource" \o "view source)[print](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "printSource" \o "print)[?](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "about" \o "?)

1.#div1 {

2.position: relative;

3.top: 40px;

4.left: 20px;

5.}

Nesse exemplo posicionamos o elemento a 40px do topo e a 20px da esquerda em relação ao lugar que ele ocuparia no fluxo do documento. Vale lembrar que utilizando position:relative todo esse espaço que foi definido ainda continua sendo ocupado pelo elemento na página.[Exemplo](Position%20Relative.html" \o "Exemplo de position:relative).

**position: absolute**

Utilizando position: absolute agora podemos posicionar exatamente nosso elemento em relação ao primeiro elemento pai com posição diferente de “static” numa sequência até parar no body.

[view source](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "viewSource" \o "view source)[print](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "printSource" \o "print)[?](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "about" \o "?)

1.#div1 {

2.position: absolute;

3.top: 40px;

4.right:0;

5.}

Vale lembrar que a patir do momento que utilizamos position: absolute ele perde a ordem no fluxo do documento e o espaço ocupado passa a não existir mais. [Exemplo](Position%20Absolute.html" \o "Exemplo de position:absolute).

**position:relative + position: absolute**

Observem o seguinte código HTML:

[view source](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "viewSource" \o "view source)[print](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "printSource" \o "print)[?](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "about" \o "?)

1.<div id="div1">

2.<div id="div2"></div>

3.</div>

Se definirmos position: relative para a div1 e position: absolute para a div2 a partir de agora a div2 se posicionará absolutamente em relação a div1, não mais em ralação ao body da página. [Exemplo](Position%20Relative+Absolute.html" \o "Exemplo de position:relative + position:absolute).

**position: fixed**

[view source](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "viewSource" \o "view source)[print](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "printSource" \o "print)[?](http://www.cssnolanche.com.br/diferenca-entre-position-static-relative-absolute-e-fixed/" \l "about" \o "?)

1.#div1 {

2.position: fixed;

3.top:40px;

4.left:10px;

5.}

Posiciona um elemento fixadamente na página em relação a janela, ou seja, independente de a pagina conter um scroll ou não, o elemento sempre ficará visíve ao usuário em relação as medidas definidas. Vale lembrar que esse tipo de posicionamento não funciona no Internet Explorer. [Exemplo](Position%20Fixed.html).