

Autoria Web

Aula 07 - CSS: modelo de caixas, bordas,
dimensões e posicionamento

Apresentação

Nesta aula, você aprenderá a criar caixas e bordas para os elementos (X)HTML. Você também verá que é possível dimensionar os elementos e dispô-los um ao lado do outro ou um sobre o outro. Com isso, você começará a aprender propriedades indispensáveis para criação de layouts.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você deverá ser capaz de:

- Definir menus e títulos por meio do modelo de caixas de CSS.
- Definir diferentes tamanhos e posições para os elementos.
- Dispor os elementos um com relação ao outro.

Modelo de Caixas

O termo caixa é utilizado no contexto de design e layout. Caixas são retângulos a serem exibidos (na tela ou impressos) que consistem de margens, bordas, espaçamentos (padding) e conteúdo.

Todos os elementos (X)HTML podem ser estilizados como caixas. Isso possibilita colocar bordas nesses elementos e espaçá-los de maneira uniforme, criando um layout geral para o documento.

Para você entender melhor, veja um exemplo na **Figura 1**. Os retângulos azuis são alguns exemplos de caixas que compõem esse documento (X)HTML.

Figura 01 - Ilustração de caixas em um documento (X)HTML

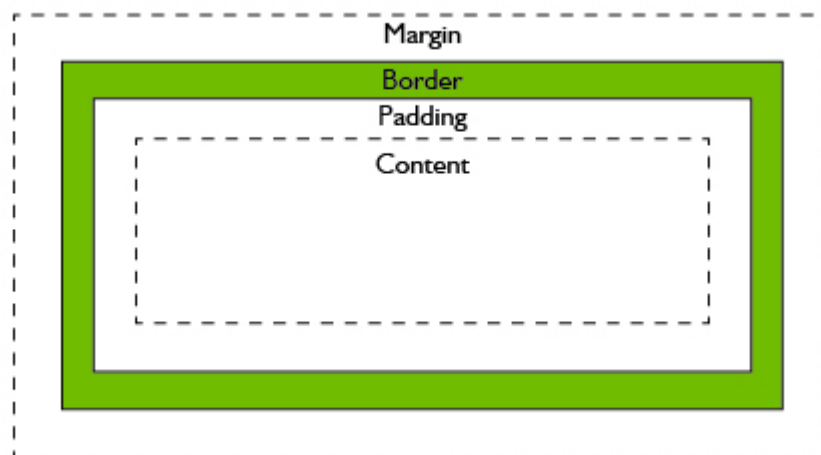


Fonte: <http://maxdesign.com.au/> Acesso em: 28 set. 2015.

As caixas podem ser representadas pelos seguintes elementos especificados a seguir e ilustrados na **Figura 2**:

- **Margin** – especifica uma área ao redor da borda, ou seja, determina a distância entre caixas. As margens são transparentes e cada elemento HTML tem uma margem default determinada pelo browser, ou seja, 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 (SILVA, 2010).
- **Border** – como você viu na estilização de tabelas e formulários, uma borda determina um tipo de linha decorativa ao redor de um elemento. Cada caixa pode ter **4** bordas, a saber: de cima, da direita, de baixo e da esquerda. Bordas são, por padrão (default), transparentes e de espessura zero.
- **Padding** – determina a distância entre a borda e o conteúdo. Esse elemento é afetado pela cor de plano de fundo da caixa, ou seja, essa distância assumirá o plano de fundo definido para o elemento.
- **Content** – indica o conteúdo da caixa; é onde o texto e imagens aparecem. É representado por qualquer elemento (X)HTML que descreve conteúdo, tais como títulos, links, listas, tabelas, formulários etc.

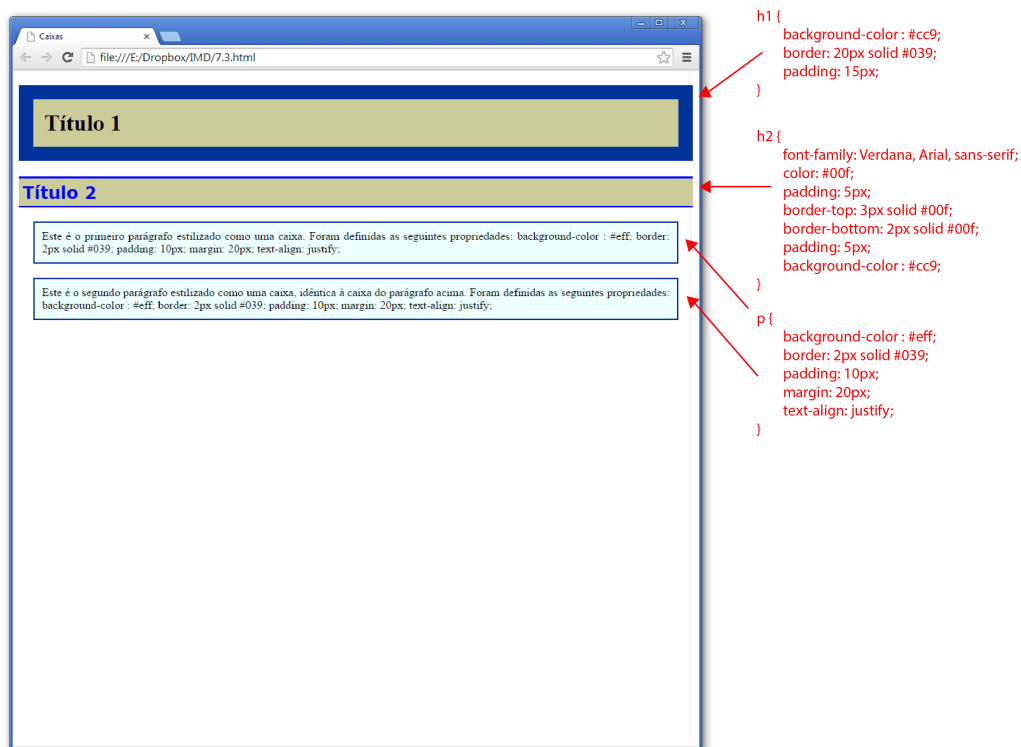
Figura 02 - Elementos que definem uma caixa



Fonte: http://www.w3schools.com/css/css_boxmodel.asp Acesso em: 28 set. 2015.

Veja um exemplo na **Figura 3**. Nesse exemplo, há dois títulos, de nível 1 e 2, e dois parágrafos. O CSS define bordas, espaçamento e plano de fundo para cada um desses elementos, veja o resultado.

Figura 03 - Exemplos de caixas



É possível definir uma cor ou uma figura para o plano de fundo das caixas, definir uma cor, uma espessura e um tipo de borda diferente para cada um dos seus quatro lados, bem como definir diferentes espessuras e espaçamentos para cada lado.

A seguir, mostramos quais as propriedades que compõem cada uma dessas partes constituintes das caixas e, em seguida, apresentamos como dimensionar os elementos (X)HTML.

Bordas

Como foi dito anteriormente, as bordas definem uma linha decorativa ao redor dos elementos (X)HTML. As propriedades para definir bordas são resumidas no quadro a seguir. Quando definirmos um valor como “tamanho” entenda que é possível definir como valor o tamanho em px, pt, cm, em, etc.

Propriedade	Descrição	Valores	Versão
border	Especifica 3 propriedades para bordas: espessura, estilo e cor	border-width border-style border-color	1
border-bottom	Especifica todas as propriedades para a borda inferior: espessura, estilo e cor	border-bottom-width border-bottom-style border-bottom-color	1
border-bottom-color	Especifica a cor da borda inferior	border-color	2
border-bottom-style	Especifica o estilo da borda inferior	border-style	2
border-bottom-width	Especifica a espessura da borda inferior	border-width	1
border-color	Especifica a cor das 4 bordas	color_name hex_number rgb_number transparent	1
border-left	Especifica todas as propriedades para a borda esquerda: espessura, estilo e cor	border-left-width border-left-style border-left-color	1

Propriedade	Descrição	Valores	Versão
border-left-color	Especifica a cor da borda esquerda	border-color	2
border-left-style	Especifica o estilo da borda esquerda	border-style	2
border-left-width	Especifica a espessura da borda esquerda	border-width	1
border-right	Especifica todas as propriedades para a borda direita: espessura, estilo e cor	border-right-width border-right-style border-right-color	1
border-right-color	Especifica a cor da borda direita	border-color	2
border-right-style	Especifica o estilo da borda direita	border-style	2
border-right-width	Especifica a espessura da borda direita	border-width	1
border-style	Especifica o estilo das 4 bordas	none hidden dotted dashed solid double groove ridge inset outset	1

Propriedade	Descrição	Valores	Versão
border-top	Especifica todas as propriedades para a borda superior: espessura, estilo e cor	border-top-width border-top-style border-top-color	1
border-top-color	Especifica a cor da borda superior	border-color	2
border-top-style	Especifica o estilo da borda superior	border-style	2
border-top-width	Especifica a espessura da borda superior	border-width	1
border-width	Especifica a espessura das 4 bordas	thin medium thick tamanho	1

Quadro 1 – Propriedades relativas a bordas.

Fonte: <http://www.w3schools.com/css/css_border.asp>. Acesso em: 28 set. 2015.



Vídeo 02 - Bordas

A **Figura 4** mostra 7 exemplos de definição de bordas, usando as propriedades border, border-style, border-top, border-bottom, border-width, e border-color. Observe cada um dos exemplos e veja que os efeitos são bem distintos, mesmo variando apenas algumas das propriedades.

Figura 04 - Exemplos de bordas

A propriedade `border-style`, a qual define o tipo de linha que circunda o elemento, pode ser especificada em termos de um até 4 valores, da seguinte maneira:

- **4 valores** indicam, respectivamente: a borda de cima, da direita, de baixo e da esquerda. Por exemplo: `border-style:dotted solid double dashed`.
- **3 valores** indicam, respectivamente: a borda de cima, as bordas da direita e esquerda (assumem o mesmo valor) e, por fim, a borda de baixo. Exemplo: `border-style:dotted solid double`.
- **2 valores** indicam, respectivamente, as bordas de cima e de baixo (assumem o mesmo valor) e as bordas da direita e da esquerda (com mesmo valor). Exemplo: `border-style:dotted solid`.
- **1 valor** indica que todas as quatro bordas assumirão o mesmo valor. Exemplo: `border-style:dotted`.

Outra maneira de fazer essa definição é usando as propriedades referentes ao estilo de cada lado, como no exemplo a seguir:

```
1 p
2 {
3   border-top-style:dotted;
4   border-right-style:solid;
5   border-bottom-style:dotted;
6   border-left-style:solid;
7 }
```

Então, como decidir quais propriedades usar? Fácil. Dê preferência por escrever o código de modo mais claro e sucinto que você puder. Isso serve para todas as outras propriedades CSS.

Margens

Como dito anteriormente, margens especificam uma área ao redor da borda, ou seja, determinam a distância entre caixas. As margens são definidas em termos das propriedades resumidas no quadro a seguir.

Propriedade	Descrição	Valores
margin	Especifica todas as propriedades de margem, nesta ordem: superior, direita, inferior e esquerda	margin-top margin-right margin-bottom margin-left
margin-bottom	Especifica a margem inferior	auto tamanho %
margin-left	Especifica a margem esquerda	auto tamanho %
margin-right	Especifica a margem direita	auto tamanho %
margin-top	Especifica a margem superior	auto tamanho %

Quadro 2 – Propriedades relativas a margens.

Fonte: <http://www.w3schools.com/css/css_margin.asp>. Acesso em: 28 set. 2015.

Mas, atenção: Quando são definidas várias caixas, uma embaixo da outra, e cada uma com suas respectivas margens, essas margens serão SOBREPOSTAS, ou seja, a margem inferior da caixa de cima se sobrepõe à margem superior da caixa de baixo, resultando numa distância que será igual à maior delas.

Por exemplo, se a margem inferior da caixa de cima for 80px e a margem superior da caixa de baixo for 30px, a distância entre elas será de 80px e não de 110px. Veja o exemplo da **Figura 5**, nele são definidos dois parágrafos: no primeiro,

a margem inferior é de 80px enquanto no segundo a margem superior é de 80px, resultando num espaçamento entre parágrafos de 80px, ao invés de ser 160px.

Figura 05 - Exemplos das propriedades relacionadas às margens



Vídeo 03 - Margem

Padding (espaçamentos)

Padding é a propriedade que determina a distância entre a borda e o conteúdo e assume o plano de fundo definido para o elemento. As propriedades que especificam esse espaçamento são listadas no quadro a seguir.

Propriedade	Descrição	Valores
Padding	Especifica todas as propriedades de espaçamento: espaçamento superior, da direita, inferior, e da esquerda	padding-top padding-right padding-bottom padding-left

Propriedade	Descrição	Valores
padding-bottom	Especifica o espaçamento inferior	tamanho (in pixels, pt, em etc.) %
padding-left	Especifica o espaçamento da esquerda	tamanho %
padding-right	Especifica o espaçamento da direita	tamanho %
padding-top	Especifica o espaçamento superior	tamanho %

Quadro 3 – Propriedades relacionadas a espaçamentos (padding).

Fonte: <http://www.w3schools.com/css/css_padding.asp>. Acesso em: 28 set. 2015.

No exemplo mostrado na **Figura 6**, o primeiro parágrafo está estilizado com a propriedade padding, enquanto o segundo parágrafo está estilizado com as propriedades padding-top, padding-right, padding-bottom, padding-left e padding-right.

Figura 06 - Exemplos das propriedades relacionadas a espaçamentos



Além de definir bordas, margens e espaçamentos, é possível também definir as dimensões de cada caixa (ou elemento).

Dimensões

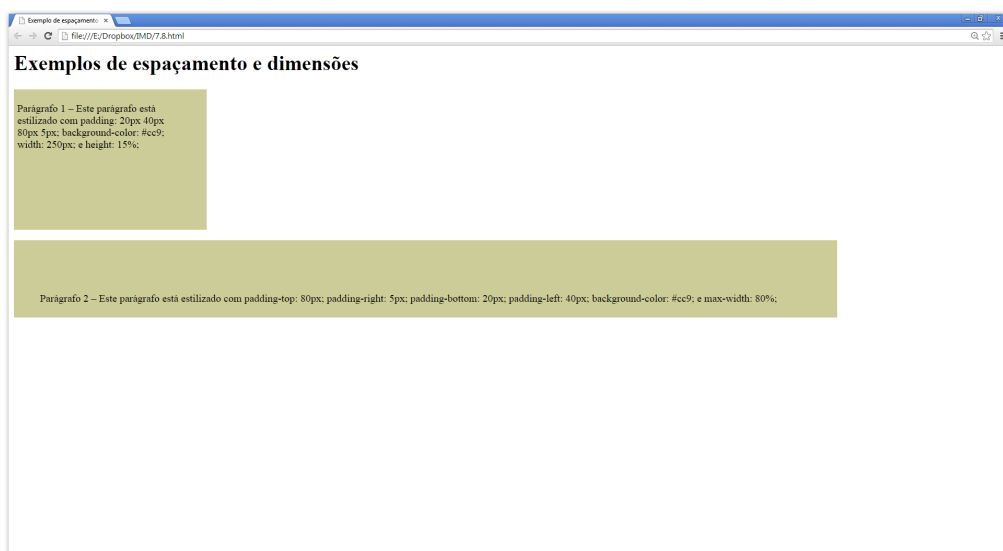


Vídeo 04 - Padding

É possível determinar as dimensões (altura e largura) de todos os elementos (X)HTML. Dessa forma, pode-se criar, por exemplo, caixas com qualquer tamanho na página, modificar somente a altura de um elemento, alterar a largura de uma tabela, etc.

Veja, na **Figura 7**, como fica o exemplo anterior (**Figura 6**) quando colocamos uma largura de 250px e uma altura de 15% para o primeiro parágrafo.

Figura 07 - Exemplo das propriedades associadas às dimensões



As dimensões são definidas pelas propriedades listadas no quadro a seguir.

Propriedade	Descrição	Valores	CSS
height	Especifica a altura	auto length %	1
max-height	Especifica uma altura máxima	none tamanho %	2
max-width	Especifica uma largura máxima	none tamanho %	2
min-height	Especifica uma altura mínima	tamanho %	2
min-width	Especifica uma largura mínima	tamanho %	2
width	Especifica a largura	auto tamanho %	1

Quadro 4 – Propriedades relacionadas às dimensões.

Fonte: <http://www.w3schools.com/css/css_dimension.asp>. Acesso em: 28 set. 2015.



Vídeo 05 - Dimensionamento

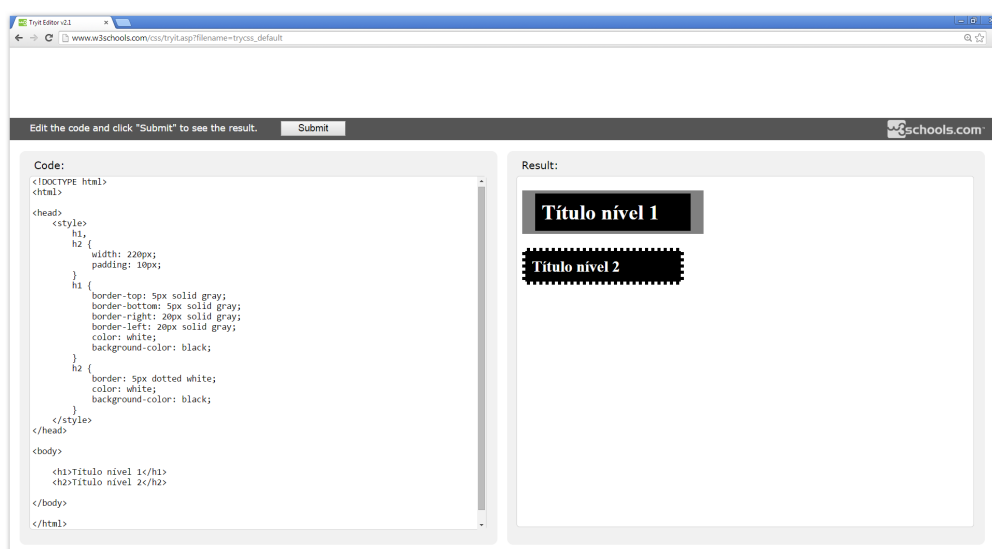
É necessário ter alguns cuidados ao utilizar essas propriedades, porque os espaçamentos definidos devem caber dentro das dimensões definidas.

Por exemplo, na **Figura 7**, você pode perceber que o primeiro parágrafo ultrapassa a caixa definida para ele e, além disso, o espaçamento de 80px que havia antes desapareceu (compare com a Figura 6). Isso ocorre porque a altura definida de 15% é insuficiente para caber o texto mais os espaçamentos de 20px superior e 80px inferior. Para solucionar um problema como este, é possível indicar no atributo de altura (height) o valor 'auto'. Com isto, caso o texto exceda o limite, o campo reservado para ele irá se adaptar para preencher o espaço que falta.

Dentre as inúmeras utilidades de se definir dimensões para os elementos, podemos citar a estilização de títulos e menus. Vamos ver um exemplo?

Na **Figura 8**, são ilustrados dois exemplos de títulos, nível 1 e nível 2, ambos foram estilizados para terem largura de 220px e espaçamento entre conteúdo e borda de 10px. Apesar de terem a mesma largura, o primeiro título aparenta ser mais largo porque possui bordas da direita e da esquerda com 20px, enquanto que o segundo título tem bordas com apenas 5px.

Figura 08 - Exemplo de estilização de títulos usando as propriedades relacionadas a caixas e dimensões



Vídeo 06 - Dimensionamento e margem

Posicionamento

Para posicionar um elemento um ao lado do outro, ou um por cima, ou por baixo do outro, deve-se utilizar as propriedades associadas ao posicionamento. Essas propriedades definem um método de posicionamento por meio da propriedade position, e a posição propriamente dita através das demais propriedades listadas no quadro a seguir.

Propriedade	Descrição	Valores Possíveis	CSS
bottom	Especifica a posição vertical inferior	auto tamanho %	2
clip	Especifica o tamanho de um elemento posicionado com o método absolute	rect (valor do topo, valor da direita, valor de baixo, valor da esquerda) auto	2
cursor	Especifica o tipo de cursor que é mostrado ao passar o mouse por cima do elemento	url auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	2

Propriedade	Descrição	Valores Possíveis	CSS
left	Especifica a posição horizontal esquerda	auto tamanho %	2
overflow	Especifica o que acontece se o conteúdo de um elemento é maior que a caixa dele	auto hidden scroll visible	2
position	Especifica o método de posicionamento	absolute fixed relative static	2
right	Especifica a posição horizontal direita	auto tamanho %	2
top	Especifica a posição vertical superior	auto tamanho %	2
z-index	Especifica se um elemento sobrepõe ou é sobreposto por outros	número auto	2

Quadro 5 – Propriedades relacionadas à posição dos elementos.

Fonte: <http://www.w3schools.com/css/css_positioning.asp>. Acesso em: 28 set. 2015.

Observação: Todas as propriedades associadas a posicionamento foram definidas na versão 2 de CSS, sendo assim, browsers antigos podem não mostrar, adequadamente, os elementos estilizados com essas propriedades.

Os possíveis métodos de posicionamento são: estático, fixo, relativo e absoluto. Esses métodos definem como cada elemento se posiciona na tela em relação aos outros elementos. Assim, todo elemento HTML usa alguma forma de se posicionar. Veja a definição de cada método a seguir

- **Estático (static)** – esse valor determina que o elemento não deve ser afetado pelas propriedades top, bottom, left e right, ou seja, eles aparecem um abaixo do outro, na ordem em que são escritos no documento (X)HTML, e nenhum elemento sobrescreve outro. Por default, todos os elementos (X)HTML são estáticos, ou seja, se você não definir nenhuma forma de posicionamento para um certo elemento, então, a propriedade position dele tem valor static.
- **Fixo (fixed)** – esse valor determina que o elemento seja posicionado em um lugar fixo na janela de tal forma que, mesmo que a janela seja rolada para cima-baixo ou esquerda-direita, aquele elemento continuará sendo visível no mesmo lugar. Dessa forma, elementos fixos podem sobrepor outros elementos e fazem com que os outros elementos se comportem como se ele não existisse.
- **Relativo (relative)** – esse valor determina que o elemento seja posicionado (mais para cima, baixo, esquerda ou direita) em relação à posição que ele ocupa no fluxo normal do (X)HTML. Dessa forma, o elemento pode ser empurrado para cima/baixo, esquerda/direita e sobrepor outros elementos, mas o espaço reservado para ele é preservado no fluxo normal descrito no (X)HTML.
- **Absoluto (absolute)** – esse valor determina que o elemento seja posicionado em relação ao seu elemento pai que tenha posição diferente de estática. Se não houver elemento pai com essa característica, então, o pai será o elemento e, então, o elemento será posicionado tomando como referência o topo esquerdo do browser. Esse método faz com que os outros elementos se comportem como se ele não existisse, ou seja, com que eles não respeitem o fluxo no qual eles foram descritos no documento HTML, podendo sobreporlos

Para você entender melhor, veja a **Figura 9**. Nela há 4 parágrafos posicionados segundo cada um dos métodos descritos acima: o parágrafo 1 usa o método fixed, o parágrafo 2 usa o método static, o parágrafo 3 utiliza o método absolute e o parágrafo 4 usa o método relative.

Observando o HTML, você verá que o parágrafo 1 é escrito antes do parágrafo 2, no entanto, na **Figura 9**, o parágrafo 1 está à direita do parágrafo 2, um ao lado do outro. Isso ocorre porque eles usam, respectivamente, as propriedades `right` e `left` e porque o método `fixed` faz com que outros elementos se comportem como se ele não existisse.

Observe, também, que os parágrafos 3 e 4 estão sobrepostos. Isso ocorre porque o método `absolute`, usado no parágrafo 3, faz com que outros elementos (que permitem sobreposição) ignorem-no. Outro detalhe a ser observado é que os parágrafos 3 e 4 aparecem abaixo do parágrafo 2 porque os métodos `absolute` e `relative` respeitam o fluxo descrito no documento (X)HTML. Rolando a página para baixo e para cima, você poderá perceber que o parágrafo 1, definido com o método `fixed`, não rola, ele fica fixo e é sobreposto pelo parágrafo 4 que estava abaixo.

Figura 09 - Exemplos de métodos de posicionamento



Vídeo 07 - Posicionamento absoluto

Como você viu, ao usar os métodos de posicionamento, é possível que alguns elementos sobreponham outros. Assim, como definir o elemento que ficará por cima e qual ficará por baixo?

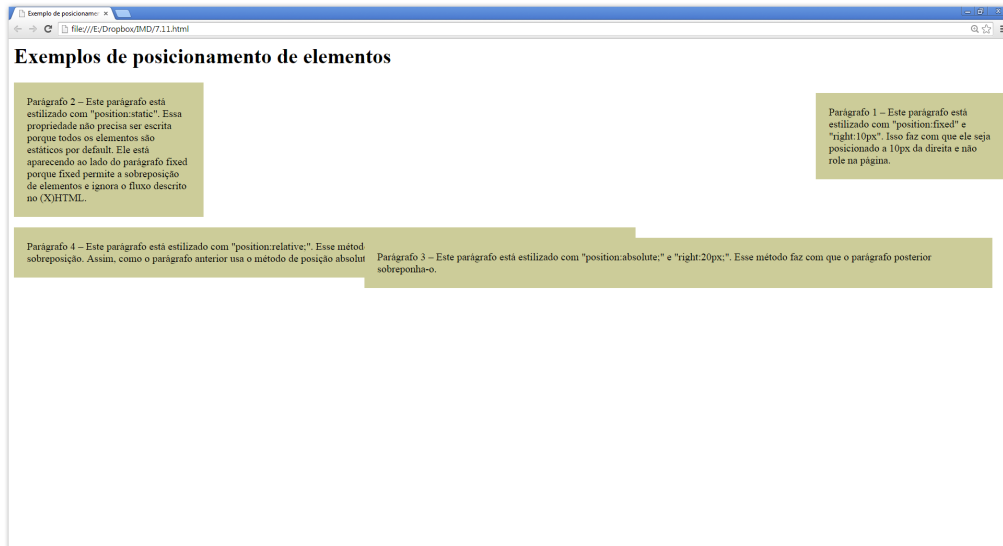
Sobreposição de Elementos

A propriedade `z-index` especifica se o elemento deve ser colocado na frente ou por trás de outros, quando eles usam um método de posicionamento que possibilita a sobreposição.

Essa propriedade pode assumir um inteiro que indicará a ordem de sobreposição, aquele que tiver maior valor ficará em cima dos que têm menor valor. Se os elementos descritos tiverem posições que se sobrepõem, mas não tiverem a propriedade `z-index`, então, o elemento que for escrito por último no HTML será posto em cima dos demais, como ocorre na **Figura 9** com os parágrafos 3 e 4.

Dessa forma, para fazer com que o parágrafo 3 apareça por cima do parágrafo 4, é necessário adicionar a propriedade `z-index=1` ao CSS relativo ao parágrafo 3, veja o resultado na **Figura 10**.

Figura 10 - Exemplo do uso da propriedade `z-index`



Enfim, agora você poderá colocar os elementos dispostos de várias maneiras e não um abaixo do outro como vinha fazendo. Entretanto, isso não é tudo, vamos fazer um exercício? Em seguida, nós continuamos com as propriedades referentes à posição dos elementos.

Atividade 01

1. Usando o exemplo da **Figura 9**, troque a ordem dos parágrafos 3 e 4 no documento (X)HTML. Explique por que o resultado no browser é diferente daquele mostrado na **Figura 9**.
2. Modificando o CSS para o documento da questão 1, desta atividade, posicione os quatro parágrafos um ao lado do outro, como mostra a **Figura 11**. Insira também uma figura no canto esquerdo por trás dos parágrafos. Estilize os textos e bordas e adicione conteúdo. Ao adicionar conteúdo, a largura da área destinada a cada parágrafo não deve mudar.

Figura 11 - Exemplo de posicionamento de parágrafos



Propriedade Display

Você concluiu o exercício anterior? Se sim, você deve estar pensando: “Será que não existe outro jeito de colocar um elemento do lado do outro?”

Sim, há mais duas maneiras. As propriedades display e float são responsáveis por isso. Vamos ver como funcionam essas propriedades?

Propriedade display

Os elementos (X)HTML são de dois tipos, a saber:

- Block – um elemento do tipo block pode ser definido como sendo aquele que tem uma largura definida (como padrão, a largura da janela do browser) e são finalizados com uma quebra de linha, por exemplo, os elementos relacionados a títulos, itens de listas, e parágrafos.
- Inline – elementos do tipo inline são aqueles que, em geral, não forçam uma quebra de linha, por exemplo, os elementos relacionados ao elemento span, links, figuras e elementos textuais, como negrito, itálico, dentre outros.

Entretanto, embora os elementos sejam definidos como sendo de um desses tipos, é possível mudar a forma como eles aparecem no browser ao utilizar a propriedade display. Essa propriedade pode assumir 3 valores:

- Inline – define que o elemento não terá a quebra de linha (antes e depois).
- Block – define que o elemento será apresentado com quebra de linha antes e depois dele.
- None – define que o elemento não será mostrado. Ele fica invisível como se não existisse.

A **Figura 12** ilustra um exemplo de lista, no qual os itens aparecem um ao lado do outro ao invés de um abaixo do outro. Para isso, foi utilizada a propriedade `display:inline`; em cada item da lista (ou seja, no elemento ``).

A propriedade `display` é muito utilizada para posicionar os itens de um menu um ao lado do outro, tal como mostrado na **Figura 12**, ou usando o valor `none`, para ocultar algum elemento, principalmente quando se está definindo um estilo para impressão (você verá um exemplo do uso do valor `none` na **Aula 8**).

Figura 12 - Exemplo do uso da propriedade `display`

Propriedade Float

A propriedade `float` possibilita que um elemento seja empurrado para direita ou esquerda e os demais elementos fiquem posicionados ao seu redor. Essa propriedade é, muitas vezes, usada para posicionar imagens e é extremamente útil na definição de layouts (você aprenderá a definir layouts na próxima aula).

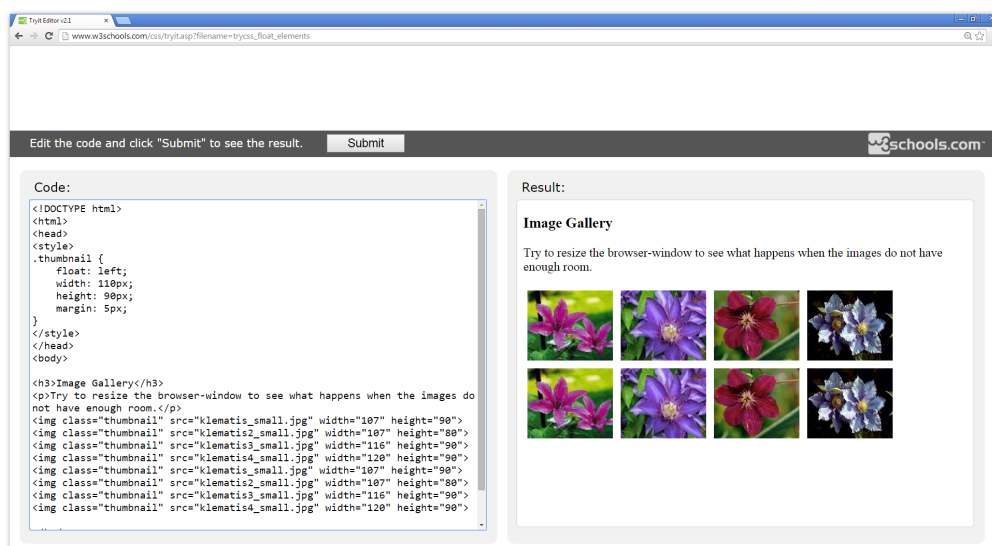
Quando essa propriedade é usada, o elemento com a propriedade `float` é posicionado à direita ou à esquerda e o elemento seguinte a ele é (imediatamente) colocado ao seu lado. Os elementos definidos antes de um elemento `float` não são afetados. Vamos ver um exemplo.

Na **Figura 13**, há dois exemplos de elementos posicionados com a propriedade `float:right`. O primeiro é uma figura e o segundo um parágrafo. Observe que o texto dos parágrafos 1, 2, 4 e 5 adapta-se para circundar a figura e o parágrafo 3.

Figura 13 - Exemplo de posicionamento usando a propriedade `float`

Assim, é possível posicionar uma figura ao lado da outra e criar uma galeria de imagens, simplesmente usando a propriedade float, como mostrado na **Figura 14**. Veja também, na **Figura 14**, que com o uso da propriedade float não foi necessário posicionar cada figura individualmente.

Figura 14 - Exemplo de posicionamento de figuras com a propriedade float



Fonte: http://www.w3schools.com/css/tryit.asp?filename=trycss_float_elements Acesso em: 28 set. 2015.

Entretanto, para evitar que um elemento definido após um elemento com a propriedade float seja afetado (ou seja, posicionado do lado dele), é necessário utilizar a propriedade clear:both;

Como exemplo, veja novamente a **Figura 14**, a posição do parágrafo que vem após a última imagem é afetada porque a última imagem é estilizada para se posicionar à esquerda do próximo elemento, assim, o próximo elemento, que é um parágrafo, fica à sua direita. Portanto, para posicionar o parágrafo abaixo das figuras, e não ao lado delas, é necessário usar a propriedade clear:both no parágrafo. Faça o teste!

No quadro a seguir, são listados os possíveis valores para clear e float.

Propriedade	Description	Values
clear	Especifica o lado de um elemento onde não é permitido posicionar um elemento float.	left right both none inherit
float	Especifica de que lado deve se posicionar um elemento.	left right none inherit

Quadro 6 – Propriedades relacionadas à float e clear.

Fonte: <http://www.w3schools.com/css/css_float.asp>. Acesso em: 28 set. 2015.

Agora, você sabe três formas de posicionar os elementos um do lado do outro, mas isso ainda não é tudo. Na próxima aula, você aprenderá a usar os elementos <div> e (que você viu, rapidamente, na **Aula 2** desta disciplina) para agrupar elementos diferentes e posicioná-los na janela de maneira muito mais elegante, por meio da criação de layouts.

Leitura Complementar

No link a seguir você encontrar o tutorial do Maujor para definição de menus – <<http://www.maujor.com/tutorial/cssmenu.php>>.

Resumo

Nesta aula, você conheceu o modelo de caixas do CSS, bem como as propriedades relacionadas à dimensão e ao posicionamento dos elementos. Com isso, você já consegue dispor os elementos em mais de uma coluna, realizar sobreposições e estilizar margens, espaçamentos e tamanho desses elementos.

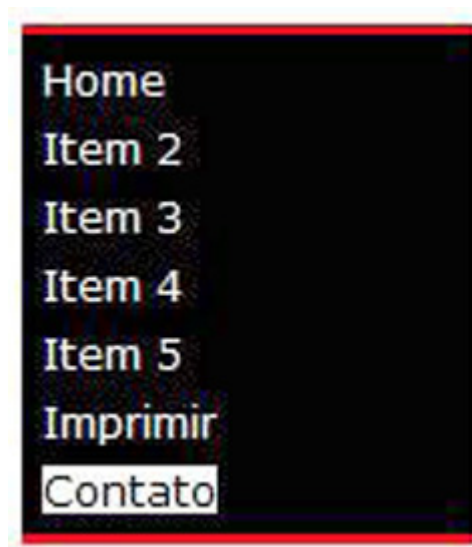
Autoavaliação

1. Defina caixa com suas palavras e defina os elementos padding, margin e border.
2. Explique qual a diferença entre os 4 métodos de posicionamento.
3. Explique qual o efeito de se aplicar a propriedade “display:inline;” aos itens de uma lista.
4. Com as propriedades associadas a caixas, listas e links crie o CSS referente ao título, barra de navegação e menu das figuras abaixo. Retorne à **Aula 5** e **6** para relembrar as propriedades associadas à estilização de listas e links. A seguir, você pode copiar o HTML referente a essas figuras.

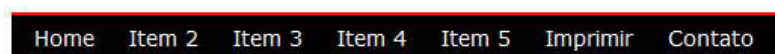
a. Título



b. Menu

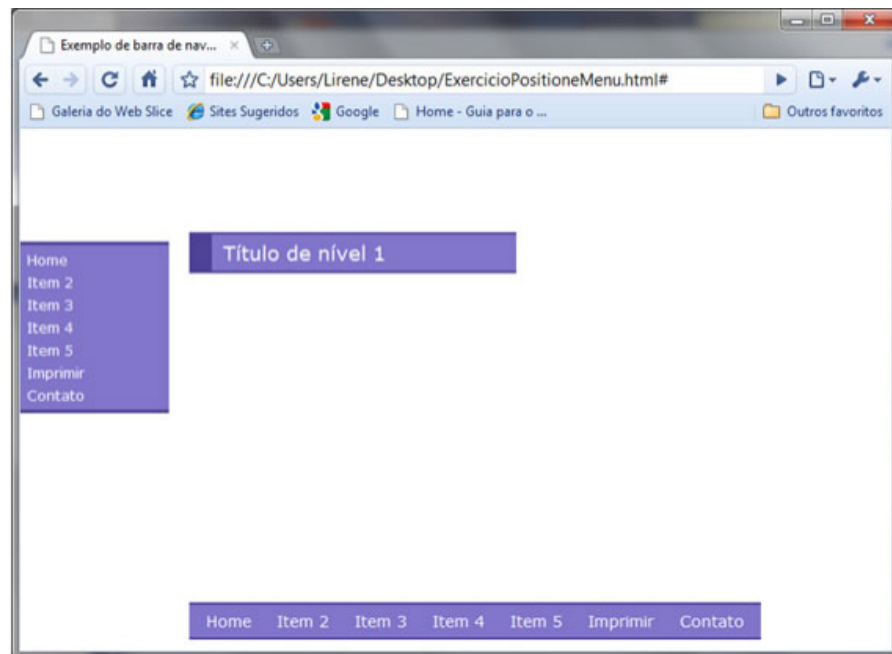


c. Barra de navegação



```
1 <html>
2   <head>
3     <title>Exemplo de menu</title>
4   </head>
5   <body>
6     <ul id="menu">
7       <li><a href="#" title="Página inicial">Home</a></li>
8       <li><a href="#" title="Item 2">Item 2</a></li>
9       <li><a href="#" title="Item 3">Item 3</a></li>
10      <li><a href="#" title="Item 4">Item 4</a></li>
11      <li><a href="#" title="Item 5">Item 5</a></li>
12      <li><a href="#" title="Imprimir">Imprimir</a></li>
13      <li><a href="#" title="Fale conosco">Contato</a></li>
14    </ul>
15  </body>
16 </html>
```

5. Use as propriedades referentes às caixas, dimensões e posicionamento para estilizar o site que você definiu para um colecionador (na aula passada). Coloque nesse site um menu e títulos similares aos que você criou na questão anterior e aplique uma única paleta de cores para todos os elementos desse site. Por exemplo, a figura a seguir ilustra uma forma de organizar os elementos da questão anterior.



Referências

Max Design. Disponível em: <<http://maxdesign.com.au/>>. Acesso em: 28 set. 2015.

CONSTRUÇÃO de menu de navegação. Disponível em: <http://www.maujor.com/tutorial/menu_vert.php>. Acesso em: 28 set. 2015.

LAYOUT CSS passo a passo. Disponível em: <<http://www.maujor.com/tutorial/layout-css-passo-a-passo.php>>. Acesso em: 28 set. 2015.