

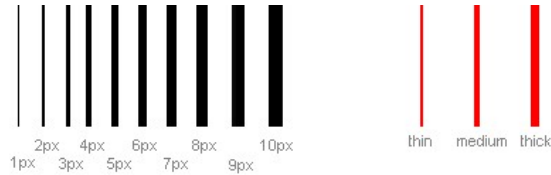
Lição 11: Bordas

As bordas podem ser utilizados para muitas coisas, por exemplo, como um elemento decorativo ou para sublinhar uma separação das duas coisas. CSS proporciona infinitas possibilidades de uso de bordas em suas páginas.

border-width
border-color
border-style
fronteira

A largura das bordas [border-width]

A largura das bordas é definida pela propriedade `fronteira largura`, que pode obter os valores de finos, médio, e de espessura, ou um valor numérico, indicado em pixels. A figura abaixo ilustra o sistema:



A cor das bordas [border-color]

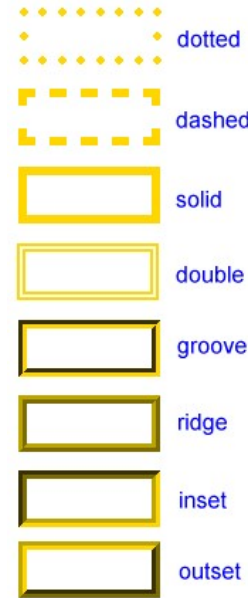


A propriedade `border-color` define a cor que a fronteira tem. Os valores são os normais cor-valores, por exemplo `"# 123456"`, `"rgb (123123123)"` ou `"amarelo"`.

Tipos de bordas [border-style]

Existem diferentes tipos de bordas disponíveis para escolha. A seguir apresentamos 8 tipos diferentes de fronteiras como o Internet Explorer interpreta-los. Todos os exemplos são mostrados com a cor "ouro" e a espessura "thick", mas pode naturalmente ser mostrada em outras cores e espessuras.

Os valores **none** ou **hidden** podem ser usados se você não quer qualquer fronteira.

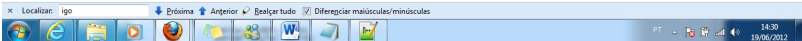
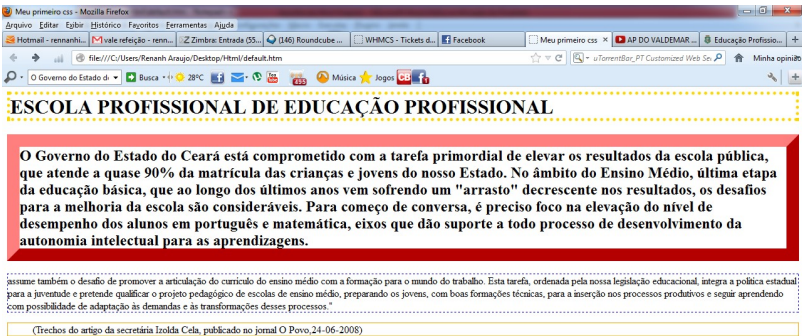


Exemplos de definição de fronteiras

As três propriedades descritas acima podem ser colocados juntos por cada elemento e resultam em diferentes bordas. Para ilustrar isso, vamos dar uma olhada em um documento onde se definir as fronteiras diferentes para `<h1>`, `<h2>`, `` e `<p>`. O resultado pode não ser tão bonito, mas ilustra algumas das muitas possibilidades:

```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}  
  
h2 {  
    border-width: 20px;  
    border-style: outset;  
    border-color: red;  
}  
  
p {
```

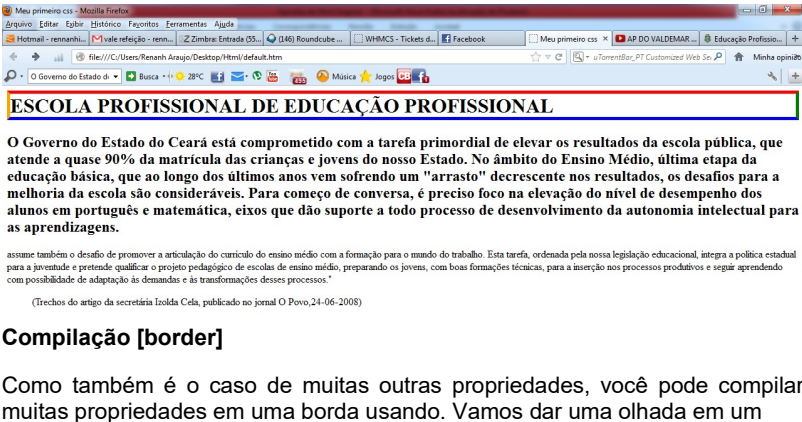
```
}  
  
ul {  
    border-width: thin;  
    border-style: solid;  
    border-color: orange;  
}
```



Também é possível definir propriedades especialmente para o alto, baixo, direita ou esquerda fronteiras. O exemplo a seguir mostra como:

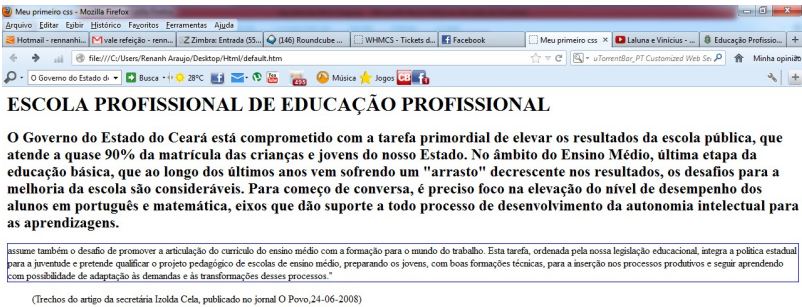
```
h1 {  
    border-top-width: thick;  
    border-top-style: solid;  
    border-top-color: red;  
  
    border-bottom-width: thick;  
    border-bottom-style: solid;  
    border-bottom-color: blue;
```

```
border-right-width: thick;  
border-right-style: solid;  
border-right-color: green;  
  
border-left-width: thick;  
border-left-style: solid;  
border-left-color: orange;
```



Exemplo:

```
p {  
    border-width: 1px;  
    border-style: solid;  
    border-color: blue;  
}
```



Pode ser compilado em:

```
p {  
    border: 1px solid blue;  
}
```

Resumo

Nesta lição você aprendeu sobre as infinitas opções de CSS dá de bordas em suas páginas.

Na próxima lição, veremos como você definir as dimensões do modelo de caixa - altura e largura.

Lição 12: Altura e largura

Até agora, nós não nos preocupamos muito com as dimensões dos elementos com os quais trabalhamos. Nesta aula, vamos dar uma olhada em como você pode facilmente definir a altura e a largura de um elemento.

largura

altura

Definir a largura [width]

Com a largura de propriedade, você pode definir a largura de um elemento.

O exemplo simples a seguir nos fornece um texto em que caixa pode ser digitado:

```
div.box {  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

Definir a altura [height]

No exemplo acima a altura da caixa é definido pelo conteúdo da caixa. Você pode afetar a altura de um elemento com a propriedade height. Como exemplo vamos tentar fazer a caixa no exemplo 500px altura:

```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

Resumo

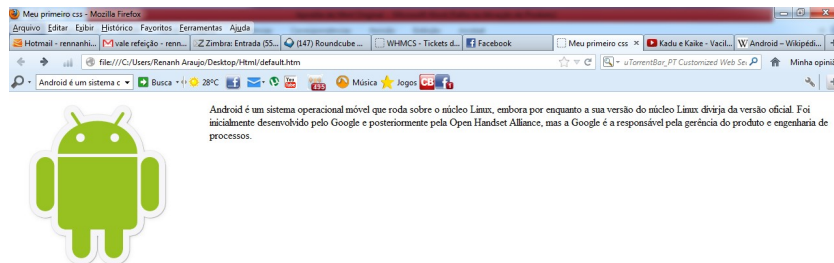
Aulas 9, 10, 11 e 12 deram-lhe uma introdução ao modelo de caixa em CSS. Como você provavelmente pode ver, o modelo de caixa lhe dá muitas opções novas. Você pode ter sido o uso de tabelas em HTML para criar seus layouts, até agora, mas com CSS e o box model você deve agora ser capaz de criar layouts mais elegantes, precisos e de acordo com as recomendações do W3C.

Lição 13: Flutuando elementos (floats)

Um elemento pode ser lançado para a direita ou para a esquerda, utilizando o flutuador propriedade. Isto quer dizer que a caixa com o seu conteúdo são deslocados para a direita ou para a esquerda em um documento (ou a caixa contendo) (ver lição 9 para uma descrição do modelo de Caixa). A figura a seguir ilustra o princípio:



Se, por exemplo, gostaria de ter uma quebra automática de texto em torno de uma imagem, o resultado seria assim:



Como é feito?

O código HTML para o exemplo acima, a seguinte aparência:

```
<div id="Android">
  
</div>
```

<p> Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gestão do produto e engenharia de processos. **</p>**

Para obter a imagem flutuando à esquerda e o texto a cercá-lo, você só tem que definir a largura da caixa que rodeia a imagem e, posteriormente, definir a propriedade float para a esquerda:

Outro exemplo: colunas

Flutuadores também pode ser usado para colunas de um documento. Para criar as colunas, você simplesmente tem que estruturar as colunas desejadas no código HTML com <div> como segue:

```
<div id="column1">
  <p>EEEE Miguel Gurgel</p>
</div>
```

```
<div id="column2">
  <p>EEEE José de Barcelos</p>
</div>
```

```
<div id="column3">
  <p>EEEE César Campelo</p>
</div>
```

Agora, a largura desejada das colunas é definido como por exemplo 33%, e então você simplesmente flutuar cada coluna para a esquerda, definindo a propriedade float:

```
#column1 {
  float:left;
  width: 33%;
}

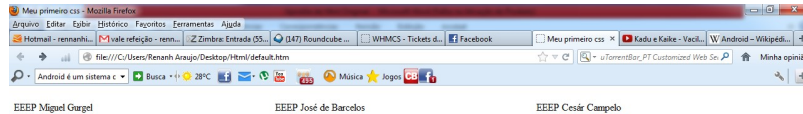
#column2 {
```

```

float:left;
width: 33%;
}

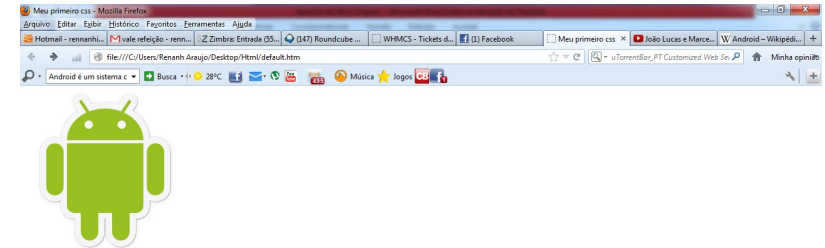
#column3 {
float:left;
width: 33%;
}

```



<h1>Android</h1>

<p class="floatstop">Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos.**</p>**



Android

Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos.



O flutuador pode ser definido como qualquer direito, esquerdo ou nenhum.

A propriedade clear

A propriedade clear é usado para controlar como os elementos se seguem aos elementos floats no documento deve se comportar.

Por padrão, os elementos subsequentes são movidos para cima para preencher o espaço disponível, que será libertado quando a caixa está flutuaram para um lado. Olhe para o exemplo acima que o texto é automaticamente movido para cima ao lado da foto de Bill Gates.

A propriedade clear pode assumir os valores left, right, ambos ou nenhum. O princípio é que, se claro, por exemplo, é definido como tanto para uma caixa, a margem superior da caixa será sempre sob a margem inferior dos possíveis caixas flutuantes que vêm de cima.

```

id="picture">

</div>

```



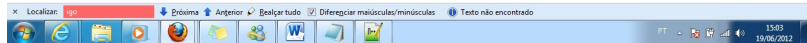
Para evitar que o texto flutue ao lado da imagem, podemos adicionar a seguinte regra CSS:

```

#picture {
float:left;
width: 100px;
}

.floatstop {
clear:both;
}

```



Resumo

Flutuadores são úteis em muitas situações e, muitas vezes, ser utilizados em conjunto com o posicionamento. Na próxima lição vamos dar uma olhada em como posicionar uma caixa, relativo ou absoluto.

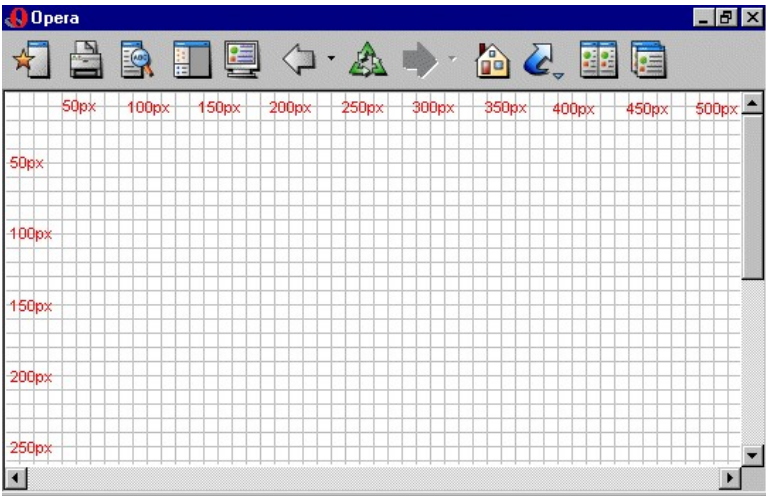
Lição 14: Posicionando elementos

Com posicionamento CSS, você pode colocar um elemento exatamente onde você quiser na sua página. Combinado com floats (ver Aula 13), o posicionamento abre muitas possibilidades para criar um layout avançado e preciso.

A seguir serão discutidos nesta lição:

- O princípio de posicionamento CSS
- O posicionamento absoluto
- O posicionamento relativo
- O princípio de posicionamento CSS

Imagine uma janela do navegador como um sistema de coordenadas:



O princípio de posicionamento CSS é que você pode posicionar qualquer caixa de qualquer lugar no sistema de coordenadas.

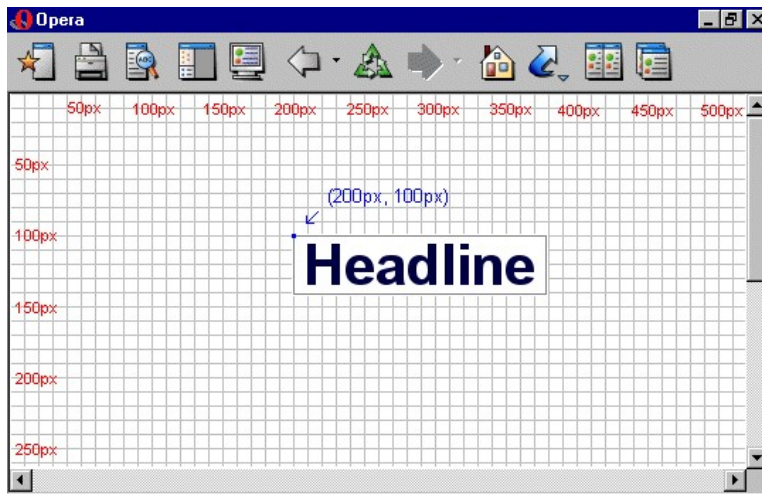
Vamos dizer que queremos posicionar um cabeçalho. Usando o modelo de caixa (ver aula 9) o título vai aparecer como segue:

Headline

Se quisermos o cabeçalho posicionado a 100px do topo do documento ea 200px à esquerda do documento, podemos usar o seguinte CSS

```
h1 {
    position: absolute;
    top: 100px;
    left: 200px;
}
```

O resultado será como se segue:



Como você pode ver, posicionar com CSS é uma técnica muito precisa para colocar elementos. É muito mais fácil do que tentar usar tabelas, imagens transparentes e tudo mais.

O posicionamento absoluto

Um elemento que está posicionado absoluto não obtém qualquer espaço no documento. Isto significa que não deixa um espaço vazio após ser posicionado.

Para posicionar um elemento de forma absoluta a propriedade position é definida como absoluta. Você pode então usar as propriedades esquerda, direita, superior, inferior e colocar a caixa.

Como exemplo de posicionamento absoluto escolhemos colocar 4 caixas em cada canto do documento:

```
#box1 {
    position: absolute;
    top: 50px;
    left: 50px;
}

#box2 {
    position: absolute;
    top: 50px;
    right: 50px;
}

#box3 {
    position: absolute;
    bottom: 50px;
    right: 50px;
}

#box4 {
    position: absolute;
    bottom: 50px;
    left: 50px;
}
```

Posicionamento relativo

Para posicionar um elemento relativamente a propriedade position deve ser definida para relative. A diferença entre o posicionamento absoluto e relativo é como a posição está sendo calculada.

A posição para um elemento que é posicionada relativamente é calculado a partir da posição original no documento. Isto significa que é mover o elemento para a direita, para a esquerda, para cima ou para baixo. Desta forma, o elemento ainda obtém um espaço no documento após ser posicionado.

Como um exemplo de posicionamento relativo, podemos tentar posicionar três imagens relativamente à sua posição original na página. Observe como as imagens deixam um espaço vazio nas suas posições originais no documento:

```
#dog1 {
    position:relative;
    left: 350px;
    bottom: 150px;
}
#dog2 {
    position:relative;
    left: 150px;
    bottom: 500px;
}

#dog3 {
    position:relative;
    left: 50px;
    bottom: 700px;
}
```

Resumo

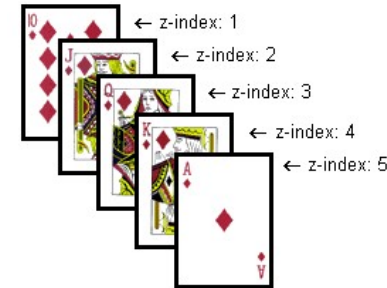
Nas duas aulas anteriores, você aprendeu como flutuar e posicionar elementos. Estes dois métodos oferecem muitas oportunidades para construir suas páginas sem ter que usar alguns dos métodos antiquados com tabelas e imagens transparentes no HTML. Use CSS vez. É mais preciso, dá-lhe mais vantagens, e é também muito mais fácil de manter.

Lição 15: camada sobre camada com z-index (Layers)

CSS opera em três dimensões - altura, largura e profundidade. Vimos as duas primeiras dimensões nas aulas anteriores. Nesta lição, vamos aprender como colocar elementos em camadas. Em resumo, isto significa que o fim do qual os elementos se sobrepõem um ao outro.

Para esse efeito, você pode atribuir a cada elemento um número (z-index). O sistema é que um elemento com um número maior sobrepõe um elemento com um número mais baixo.

Vamos dizer que estamos a jogar poker e ter um royal flush. As cartas podem ser apresentados de uma forma onde cada cartão tem um z-index:



Neste caso, os números seguir em outro (1-5) mas o mesmo resultado pode ser obtido usando 5 números diferentes. O importante é a sequência cronológica dos números (a ordem).

O código no exemplo a placa poderia ser assim:

```
#dez_ouro{
    position: absolute;
    left: 100px;
    top: 100px;
    z-index: 1;
}

#valet_ouro{
    position: absolute;
    left: 115px;
    top: 115px;
    z-index: 2;
}

#rainha_ouro {
    position: absolute;
    left: 130px;
    top: 130px;
    z-index: 3;
}

#rei_ouro {
    position: absolute;
```



```
    left: 145px;  
    top: 145px;  
    z-index: 4;  
}  
  
#az_ouro {  
    position: absolute;  
    left: 160px;  
    top: 160px;  
    z-index: 5;  
}
```

O método é relativamente simples, mas as possibilidades são várias. Você pode colocar imagens no texto, texto sobre texto etc.

Resumo

As camadas podem ser usadas em muitas situações. Por exemplo, tente usar z-index para criar efeitos em manchetes em vez de criar estes como elementos gráficos. Por um lado, é mais rápido para carregar o texto e, por outro, fornece um ranking potencialmente melhor nos motores de busca.

Lição 16: Web standards e validação

W3C é a World Wide Web Consortium, que é uma organização independente que administra as normas do código na web (HTML, CSS, XML e outros). Microsoft, Fundação Mozilla e muitos outros fazem parte da W3C e de acordo sobre o futuro desenvolvimento das normas.

Se você estiver trabalhando um pouco com web design, provavelmente sabe que não pode haver uma grande diferença na forma como uma página web é apresentada em diferentes navegadores. Pode ser muito frustrante e demorado para criar uma página web que pode ser visto no Mozilla, Internet Explorer, Opera e todo o resto dos navegadores existentes.

A idéia de ter padrões é acordar um denominador comum sobre como utilizar as tecnologias da web. Isto significa que, observando as normas, um desenvolvedor terá a certeza que o que ele ou ela não vai funcionar de uma forma mais apropriada em diferentes plataformas. Por isso, recomendamos que você faça backup o trabalho realizado pelo W3C e valide sua CSS, a fim de observar o padrão.

CSS validator

Se o validador não encontrar erros; essa imagem será mostrado, que você pode colocar no seu site para ilustrar que você está usando codificação validados:



CSS válido!

O validador pode também ser encontrado neste link: <http://jigsaw.w3.org/css-validator/>