

Apostila CSS

SUMÁRIO

| | |
|---|----|
| Introdução..... | 3 |
| Regras e Sintaxe | 3 |
| Método Incorporado, Inline e Externo..... | 4 |
| Estilo inline..... | 4 |
| Estilo incorporado | 5 |
| Estilo externo | 5 |
| Especificidade e Precedência..... | 6 |
| Seletores CSS no HTML (id e class) | 7 |
| Estilização de textos..... | 8 |
| Alinhamento | 8 |
| Efeitos em textos | 9 |
| Text-decoration..... | 9 |
| Text-Transform | 9 |
| Links | 10 |
| Boxes..... | 11 |
| Margin e Padding..... | 12 |
| Border | 14 |
| Border-radius | 14 |
| Box-shadow | 15 |
| Posicionamento absoluto e relativo | 15 |
| Posicionamento Absoluto..... | 15 |
| Posicionamento Relativo..... | 17 |
| Utilizando camadas (layers) | 18 |
| Layouts Baseados em CSSs – Tableless..... | 19 |

Antes de começar é importante salientar para você que os retângulos vermelhos são **exemplos de códigos em HTML** e os **azuis são códigos CSS**.

Introdução

O arquivo CSS assim como o HTML tem a sua extensão específica, toda vez que você salvar um arquivo CSS é necessário finalizar com a extensão **.css**.



Regras e Sintaxe

Para utilizarmos estilos CSS em um elemento, fazemos uso das regras CSS. Uma regra é escrita como o exemplo abaixo:

```
body {  
  background-color: white;  
}
```

No exemplo acima, definimos como cor de fundo da **tag body a cor branca**. Uma regra é feita dos seguintes componentes:

- **Seletor:** é o elemento HTML para o qual a regra será; (por exemplo: <p>, <h1>, <form>, .minhaclasse, #minhdiv etc...);
- **Propriedade:** é o atributo do elemento HTML ao qual será aplicada a regra (por exemplo: font, color, background, etc...).
- **Valor:** é a característica específica a ser assumida pela propriedade (por exemplo: letra tipo Arial, cor azul, fundo verde, etc...)

A Sintaxe é demonstrada abaixo:

```
seletor {propriedade: valor;}
```

Todas as declarações são terminadas por ponto-e-vírgula ";".

```
p {  
  font-size: 12px;  
}  
  
body {  
  color: #000000;  
  background: #white;  
  font-weight: bold; /*Comentário qualquer*/  
}
```

No primeiro exemplo, o **seletor** é o elemento HTML **<p>** parágrafo. No segundo exemplo o **seletor** é documento todo (**body**), e as propriedades declaradas com seus respectivos valores são:

- Cor da fonte: **preta** (*color: #000000;*);
- Cor do fundo: **branca** (*background: white;*);
- Peso da fonte: **negrito** (*font-weight: bold;*).

Dica 1: é de boa técnica usar-se sempre o ponto-e-vírgula após cada regra final;

Dica 2: escreva suas regras CSS digitando uma propriedade em cada linha. Isto facilita a manutenção e atualização das Folhas de Estilo;

Dica 3: Insira comentários nas suas Folhas de Estilo. Eles facilitaram a você se lembrar no futuro, de detalhes da programação, que certamente serão esquecidos com o tempo.

Método Incorporado, Inline e Externo

Quando falamos em desenvolvimento de sites, chamamos tipo de CSS a forma com que o documento HTML irá usar os estilos CSS. Existem três tipos de folha de estilo CSS possíveis para utilização em projetos WEB: inline, incorporado e externo.

Estilo inline

Estilo inline é quando os estilos CSS são aplicados diretamente dentro da *tag* de abertura de um elemento HTML. No entanto, este tipo de uso das CSS não é recomendado e encontra-se em desuso pelos desenvolvedores WEB. Veja a seguir um exemplo de utilização deste tipo de estilo:

HTML

```
<div style= "background-color: blue; color: red; width:300px; height:200px;">  
  <h1>...</h1>  
  <p>...</p>  
  <p>...</p>  
</div>
```

Veja que para o elemento *div* acima existe uma formatação de estilo *inline*, ou seja, dentro da própria *tag* de abertura do elemento *div*. A manutenção deste tipo de CSS é muito complicada e difícil, gerando altos custos para o projeto.

Estilo incorporado

O estilo incorporado também não é o mais indicado e recomendado, uma vez que estilos escritos desta forma ficam restritos a um único documento, não sendo possível a sua reutilização em outras páginas do projeto WEB.

Neste método, você declara a **folha de estilo** diretamente dentro da seção *head* do documento HTML, utilizando o elemento *style*. Veja um exemplo a seguir da utilização deste tipo de estilo:

```
HTML

<html>
  <head>
    <style type="text/css">
      h2 {color: #000000;}
    </style>
  </head>
  <body>
    <h2>Título Secundário da Página</h2>
  </body>
</html>
```

Veja que no exemplo acima o elemento **h2** está recebendo uma formatação de estilo. Perceba que esta formatação está embutida no próprio documento HTML, na seção *head*.

Estilo externo

Este tipo de uso de CSS em documentos HTML é o mais indicado e recomendado. Neste tipo, os códigos CSS ficam em um documento externo ao documento HTML, separando assim as camadas de estrutura e apresentação do projeto WEB.

Para o tipo de estilo CSS externo linkado, usamos o elemento *link* para realizar a ligação entre o documento CSS externo ao documento HTML. O elemento *link* deve ser usado **dentro da seção head do documento HTML**.

Veja a seguir um exemplo da utilização do elemento *link* para incorporar uma folha de estilo CSS ao documento HTML. Neste exemplo, imagine que o documento CSS esteja dentro de uma pasta chamada "css" e o seu arquivo HTML, "index.html" esteja na raiz, como pode ser visto na ilustração abaixo:

HTML

```
<head>  
  <link rel="stylesheet" type="text/css" href="css/estilos.css" />  
</head>
```

Partindo do princípio que o documento CSS esteja nomeado como “estilos.css” e que esteja dentro de uma pasta chamada “css”, o atributo *href* deve ficar exatamente com esta informação: “**css/estilos.css**”, que é a forma com que o navegador entender onde se encontra o arquivo CSS a ser linkado.

Especificidade e Precedência

As CSS também obedecem a algumas regras de precedência. Quando várias regras se aplicam a um mesmo elemento, o navegador precisa decidir qual delas tem precedência. Vamos criar uma div:

HTML

```
<div id="sidebar">  
  <p class="descricao">  
    Texto exemplo.  
  </p>  
</div>
```

Vamos supor que temos duas regras aplicadas à div:

CSS

```
#sidebar {  
  color: #F00;  
}  
div#sidebar p {  
  color: #F0F;  
}
```

[Visualizar Exemplo](#)

Qual delas será aplicada? Se você pensou a "mais específica", acertou. O seletor **div.sidebar p** tem precedência.

Seletores CSS no HTML (id e class)

No CSS existem basicamente três formas de definir estilos para os elementos do HTML.

- o próprio elemento;
- class;
- id;

O próprio elemento seria chamar a tag do elemento HTML no CSS, *class* é o seletor iniciado por um “.” e o *id* são os seletores iniciados por um “#” Exemplificando:

HTML

```
<div></div>
<div class="container"></div>
<div id="box"></div>
```

CSS

```
div {
  background-color: red;
}
.container {
  background-color: green;
}
#box {
  background-color: blue;
}
```

Esta é a primeira diferença, a forma que referenciamos cada um deles no HTML um por *class* e outro por *id*. E a segunda a chamada no CSS, class por “.” e id por “#”.

Nas diferenças mais técnicas: as classes são utilizadas quando pretende-se repetir o elemento várias vezes na página. Então utilizamos a mesma classe para cada uma destas repetições. Uma classe será utilizada em elementos que precisam repetir muitas vezes em uma página.

O *id* é o contrário, quando o elemento não vai repetir na página, elemento único, ele preferencialmente deve receber um *id*. Então não é permitido ter dois *ids* iguais na mesma página. Veja o exemplo:

```
<ul id="compras">
  <li class="item">Arroz</li>
  <li class="item">Feijão</li>
  <li class="item">Macarrão</li>
  <li class="item">Carne</li>
</ul>
```

Neste caso teremos apenas uma lista de compras na nossa página. Então esta lista receberá o *id* de **compras**. Compras contém vários itens nesta lista e é possível reaproveitar em outra, utilizamos a *class* **item** para padronizar o estilo deste elemento. Assim economizamos também linhas de CSS, declarando e estilizando ele uma só vez



- Um elemento pode ter só um *id*;
- Um *id* não deve ser utilizado mais que uma vez na página;
- Pode-se utilizar a mesma classe em vários elementos;
- Um elemento pode ter múltiplas classes;

Outro ponto importante é que o *id* é mais específico que a *class*. Então se você colocar dois estilos, por exemplo, de background-color em um elemento que tem *class* e *id*, o estilo do *id* vai sobrepor o da classe. Como demonstramos na precedência do CSS.

Obs.: Se você não precisar de *class* e *id* não os utilize, utilize o próprio seletor do elemento, o código fica até mais simples de entender, porque todos já sabem o que um elemento **<a>** representa mas não o que a classe ou *id* deste elemento representa.

Estilização de textos

As propriedades para textos, definem as características dos textos inseridos dentro dos elementos HTML. As propriedades para textos são as listadas abaixo:

- color: cor do texto;
- text-align: alinhamento do texto;
- text-decoration: decoração do texto;
- text-transform: forma das letras;
- direction: direção do texto;
- letter-spacing: espaçamento entre;
- word-spacing: espaçamento entre palavras;

Alinhamento

Em relação ao alinhamento, temos a propriedade *text-align* que controla o posicionamento horizontal do conteúdo de um elemento nível de bloco. Os valores possíveis são:

- left
- right
- center
- justify

Esta propriedade coloca os textos às esquerdas, direita, centralizado ou justificados em relação a TAG do HTML que você indicou.

No exemplo abaixo, foi criado quatro classes para alterar o alinhamento do texto através da propriedade `text-align`:

CSS

```
.centro{text-align: center;}  
.direita{text-align: right;}  
.esquerdo{text-align: left;}  
.justificado{text-align: justify;}
```

Efeitos em textos

Text-decoration

Além de controlar espaçamentos e alinhar textos, podemos utilizar as propriedades CSS para adicionar uma série de efeitos em textos. A primeira propriedade a ser estudada é a **propriedade** *text-decoration* que causa um efeito decorativo no texto.

Os valores possíveis desse efeito são:

- none (sem decoração)
- underline
- overline
- line-through

HTML

```
<p class linhacortando ">Linha cortando</p>  
<p class duaslinhas ">Duas linhas</p>  
<p class ="piscar">Piscar</p>
```

CSS

```
.linhacortando{text-decoration: line-through;}  
.duaslinhas {text-decoration: overline underline;}
```

Resultado

~~Linha cortando~~

Duas linhas

Text-Transform

A propriedade `text-transform` controla os efeitos de capitalização (tornar maiúscula ou minúscula) dos textos. Os valores possíveis para essa propriedade são:

- capitalize: Capitaliza a primeira letra de cada palavra. Por exemplo: "felipe pains" transforma-se para "Felipe Pains".

- **uppercase:** Transforma todas as letras em maiúsculas. Por exemplo: “felipe pains” para “FELIPE PAINS”.
- **Lowercase:** Transforma todas as letras para minúsculas. Por exemplo: “Felipe PAINS” para “felipe pains”.
- **none:** Sem transformações, o texto é apresentado como foi escrito no HTML.

Independentemente de como o texto foi escrito no código HTML, essas propriedades vão modificar como os textos serão apresentados.

Links

O elemento âncora (<a>) juntamente com o atributo href têm a função de criar links em um documento. A novidade aqui é que você pode definir as propriedades de maneira diferenciada de acordo com o estado do link ou seja visitado, não visitado, ativo ou com o ponteiro do mouse sobre o link.

Para estilizar estes efeitos você usará as chamadas **pseudoclasses**. Uma **pseudoclasse** permite estilizar levando em conta condições diferentes ou eventos ao definir uma propriedade de estilo para uma tag HTML. São quatro pseudoclasses para links:

- **a:link** – Como fica o link no estado inicial (sem ser visitado). No exemplo abaixo os links que não forem visitados ficarão com a cor rosa.

```
a:link {  
  color: pink;  
}
```

- **a:visited** – Como fica o link depois que foi clicado. No exemplo abaixo os links visitados ficarão na cor verde.

```
a:visited {  
  color: green;  
}
```

- **a:hover** – Como fica o link quando passamos o cursor do mouse sobre ele. No exemplo a seguir, quando o usuário passar o mouse sobre o link, ele ficara laranja com a fonte em itálico.

```
a:hover {  
  color: red;  
  font-style: italic;  
}
```

- **a:active** – Como fica o link quando clicamos sobre ele. No exemplo a seguir, os links ativos ficarão na cor amarela.

```
a:active {  
  color: violet;  
}
```

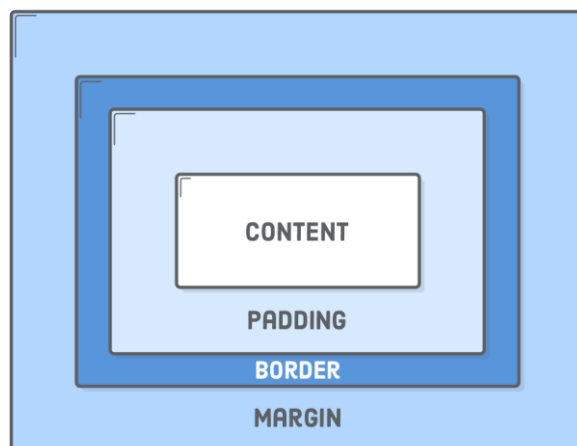
Boxes

Todo elemento HTML é uma "caixa" retangular a ser apresentada na tela com as estilizações determinadas pelas regras CSS. As caixas são empilhadas uma após a outra e constituídas de margens, bordas, espaçamentos e o conteúdo propriamente dito.

Desta forma, cabeçalhos (<h1>,<h2>,<h3>..), parágrafos (<p>), listas (,), 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 caixas CSS são constituídas por 04 (quatro) áreas retangulares, que se desenvolvem de dentro para fora, na ordem listada abaixo:

- conteúdo (content);
- espaçamentos (padding);
- bordas (border);
- margens (margin).



Para utilizar esses recursos é importante entender as unidades de medidas que podem ser utilizadas, que podem ser relativas ou absolutas. Unidades absolutas são unidades de medida de comprimento definidas nos sistemas de medidas pela física, a citar centímetros, milímetros, polegadas, etc. As unidades de medida relativas são aquelas que tomam como base um valor de referência e são mais apropriadas para ajustes de uso em diferentes tipos de mídia.

Exemplos de medidas absolutas são:

- in = polegada
- cm = centímetro
- mm = milímetro
- pt = ponto

As quatro unidades de medida relativa são:

- em = 1em é igual ao tamanho de fonte definido para o elemento em questão
- ex = 1ex é igual a altura da letra xis minúscula(x) da fonte definida
- px = ao dispositivo (mídia) de exibição;
- %: ... a uma medida previamente definida.

Voltando ao modelo de caixa(box-model), uma box é constituída por partes distintas.

A margin é invisível, não possui cor de fundo e não esconde elementos

A border permite visualizar os limites visíveis da box

O padding define o espaço entre o conteúdo e o border.

Veja os códigos abaixo e em seguida visualize o resultado.

```
<p class="box">CONTEÚDO</p>
```

```
.box {width: 100px;  
height: 50px;  
border: 10px solid #ccc;  
padding: 20px;  
margin: 20px;  
}
```

#ccc é uma cor em hexadecimal que representa a cor cinza.

Resultado:



Margin e Padding

A propriedade para margens (**margin**), define um valor para espessura das margens dos elementos HTML. As propriedades para margens são as listadas abaixo:

- margin-top: define a margem superior;

São válidos valores **negativos** para margem, com o objetivo de sobrepor um elemento.

- `margin-right`: define a margem direita;
- `margin-bottom`: define a margem inferior;
- `margin-left`: define a margem esquerda;
- `margin`: maneira abreviada para todas as margens.

Os valores possíveis para as propriedades das margens são:

- `auto`: valor default da margem
- `length`: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
- `%`: porcentagem da largura do elemento pai.

A propriedade `margin` admite a sintaxe abreviada, a qual consiste em declarar uma lista de valores separados por um espaço, conforme mostrado. Há quatro modos de se declarar abreviadamente as margens:

- `margin: valor1` - as 4 margens terão valor1;
- `margin: valor1, valor2` - margem superior e inferior terão valor1 - margem direita e esquerda terão valor2
- `margin: valor1, valor2, valor3` - margem superior terá valor1 - margem direita e esquerda terão valor2 - margem inferior terá valor3
- `margin: valor1, valor2, valor3, valor4` - margens superior, direita, inferior e esquerda nesta ordem.

```
.box {  
margin: 20px /* margem de 20px nos quatro lados */  
margin: 15px 10px /* margem superior e inferior de 15px e direita e esquerda de 10px */  
margin: 15px 5px 10px /* margem superior 5px, direita e esquerda 5px e inferior de 10px */  
margin: 15px 5px 10px 20px /* margem superior 5px, direita 5px, inferior 10px e esquerda de 20px */  
}
```

A propriedade para espaçamentos (**`padding`**), define um valor para os espaçamentos entre o conteúdo e as bordas dos elementos HTML. Assim como no **`margin`**, as propriedades são similares.:

- `padding-top`: define a espaçamento superior;
- `padding-right`: define a espaçamento direita;
- `padding-bottom`: define a espaçamento inferior;
- `padding-left`: define a espaçamento esquerda;
- `padding`: maneira abreviada para todas os espaçamentos.

Os valores também são similares aos do **`margin`**:

- `auto`: valor default do espaçamento
- `length`: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
- `%`: porcentagem da largura do elemento pai

```
.box {  
padding: 20px /* espaçamento de 20px nos quatro lados */  
padding: 15px 10px /* espaçamento superior e inferior de 15px e direita e esquerda de 10px */  
padding: 15px 5px 10px /* espaçamento superior 5px, direita e esquerda 5px e inferior de 10px */  
padding: 15px 5px 10px 20px /* espaçamento superior 5px, direita 5px, inferior 10px e esquerda de 20px */  
}
```

Border

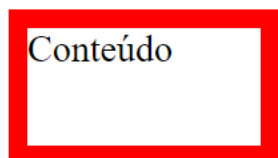
A propriedade **border** define a espessura, a cor e o estilo das bordas do box. A sintaxe da propriedade é dada abaixo:

```
border: border-width border-style border-color;
```

Exemplo:

```
.box {  
border: 1px solid red;  
}
```

Resultado:

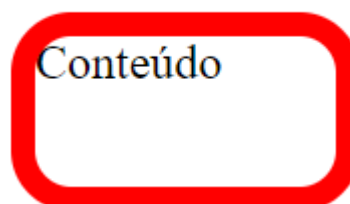


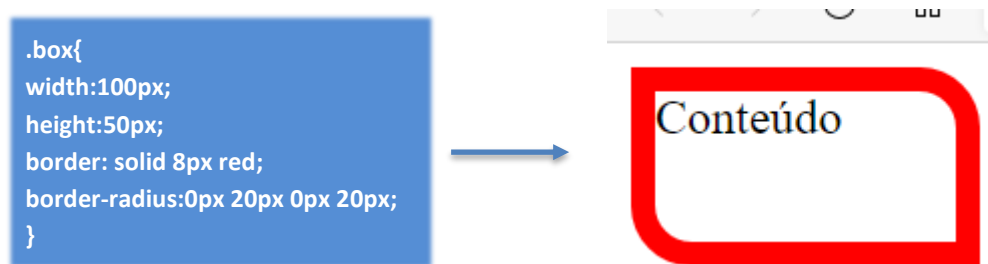
Border-radius

Esta propriedade define uma borda arredondada de acordo com o valor fornecido, que indica o raio de uma circunferência teoricamente posicionada nos vértices do elemento e a partir da qual se obtém o efeito desejado.

A propriedade **border-radius** pode ser definida de duas formas: definindo um valor único para todos os vértices (pontas) ou especificando individualmente o valor de cada vértice.

```
.box{  
width:100px;  
height:50px;  
border: solid 8px red;  
border-radius:20px;  
}
```





Na definição de valores individuais, a ordem dos vértices é a seguinte:

- Superior esquerdo
- Superior direito
- Inferior esquerdo
- Inferior direito

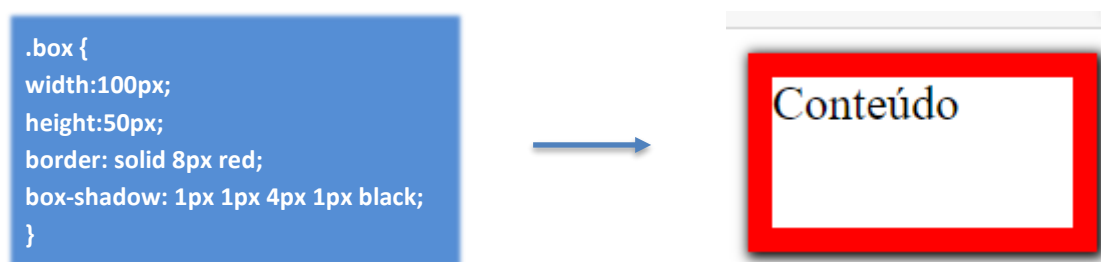
Box-shadow

Outro efeito que há algum tempo necessitava de algumas imagens para ser obtido é o de sombreamento. Esta propriedade é composta por alguns parâmetros e sua sintaxe de uso é a seguinte:

Sintaxe:

```
box-shadow: h-shadow v-shadow blur spread color;
```

- h-shadow: posição horizontal da sombra.
- v-shadow: posição vertical da sombra.
- blur: define a intensidade do efeito blur (borrão) da sombra (opcional).
- spread: o tamanho da sombra (opcional).
- color: a cor da sombra.



Posicionamento absoluto e relativo

Existem quatro tipos de posicionamento em CSS: estático, relativo, absoluto e fixo. A propriedade CSS que possibilita posicionar um elemento qualquer é a propriedade **position**. Esta propriedade deverá receber um dos valores referentes aos quatro tipos de posicionamento CSS. Neste capítulo vamos falar do **relativo** e **absoluto**.

Posicionamento Absoluto

O posicionamento absoluto é definido pelo valor **absolute** da propriedade position

Seletor {position: absolute}

Quando um elemento é posicionado absolutamente, sua posição é computada de acordo com a posição do elemento “posicionado” mais próximo, que o contém. Elemento “posicionado” é qualquer elemento que tenha seu posicionamento definido como relativo, absoluto ou fixo. Quando não há nenhum elemento “posicionado”, a posição é computada com relação ao elemento body.

Na regra CSS, ao definir a posição absolute utilizamos as propriedades (left) que define o quanto a imagem dista da margem esquerda e a propriedade (top) que define o quanto a imagem dista do topo.

```
  

```

```
#img1 {  
  position: absolute;  
  top: 100px;  
  left: 0px;  
}  
#img2 {  
  position: absolute;  
  top: 100px;  
  left: 0px;  
}
```

Resultado:

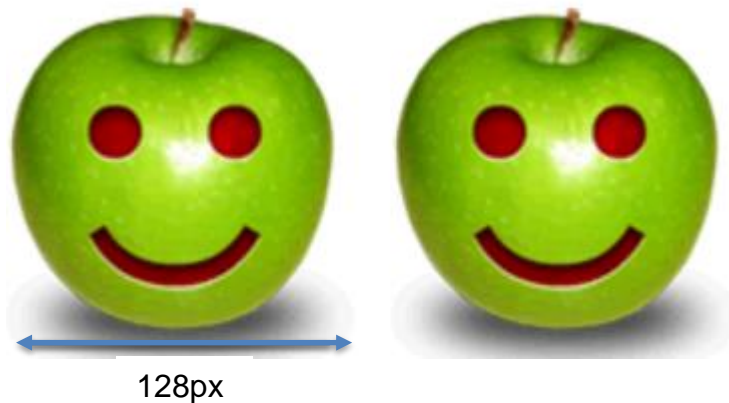


Uma imagem está na frente da outra, por isso visualizamos apenas umas.

Uma solução possível para isso seria altera no CSS o código para o elemento **img2**, como é demonstrado a seguir:


```
#img2{  
position: absolute;  
top:100px;  
left:128px;  
}
```

Resultado:



A segunda imagem à direita está com a distância da borda da página de 128px, que é o tamanho da largura da imagem.

Posicionamento Relativo

O posicionamento absoluto é definido pelo valor absolute da propriedade position

Seletor {position: relative}

A diferença entre os dois tipos de posicionamento é a maneira como o posicionamento é calculado. O posicionamento para **posição relativa** é calculado com base na posição original do elemento no documento. Isto significa uma movimentação do elemento para a esquerda, para a direita, para cima ou para baixo.

Utilizando o mesmo exemplo apresenta no posicionamento absoluto, mas com o relativo

```
  

```

```
#img1 {  
position: relative;  
top: 100px;  
left: 0px;  
}  
#img2 {  
position: relative;  
top: 100px;  
left: 0px;  
}
```

Resultado:



Com o posicionamento absoluto, as imagens apareceram sobrepostas, já que elas estavam posicionadas a 0px do elemento que as contém (**body**)

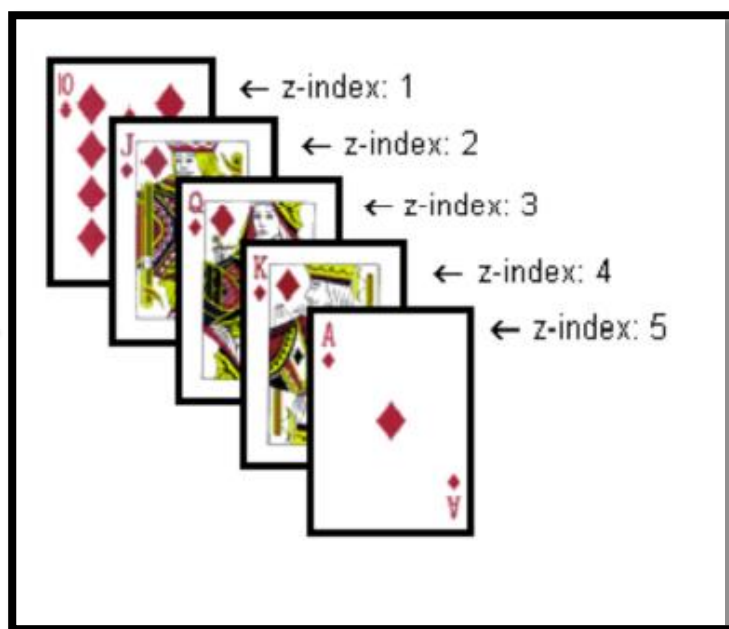
Com o posicionamento relativo, a primeira imagem aparecerá a 0(zero) px do elemento anterior(**body**), e a segunda imagem aparecerá a 0px do elemento anterior(**img1**).

Utilizando camadas (layers)

CSS usa o espaço tridimensional - altura, largura e profundidade. Camadas significam como os elementos se sobrepõem uns aos outros. Para fazer isto definimos para cada elemento um número índice (z-index). O comportamento é que elementos com número índice maior se sobrepõem àqueles com menor número.

Vamos supor um **Royal flush** no jogo de poker. As cartas podem ser apresentadas como se cada uma delas tivesse um z-index:

```
#dez_ouro {  
  position: absolute;  
  left: 100px;  
  bottom: 100px;  
  z-index: 1;  
}  
#valet_ouro {  
  position: absolute;  
  left: 115px;  
  bottom: 115px;  
  z-index: 2;  
}  
#dama_ouro {  
  position: absolute;  
  left: 130px;  
  bottom: 130px;  
  z-index: 3;  
}  
#rei_ouro {  
  position: absolute;  
  left: 145px;  
  bottom: 145px;  
  z-index: 4;  
}  
#as_ouro {  
  position: absolute;  
  left: 160px;  
  bottom: 160px;  
  z-index: 5;  
}
```



Layouts Baseados em CSSs – Tableless



TABLELESS

A internet precisou passar por algumas mudanças, uma delas é o Tableless. Com ela é possível chegar ao nível que estamos hoje, com sites responsivos, sites móveis e a enorme quantidade de novas tecnologias que surgem em tempo recorde.

Tableless: Um site tableless é aquele que não utiliza tabelas para a estruturação do Layout. É um site que segue os Padrões Web. O termo “tableless” é usado mais comumente aqui no Brasil. Em outros países, outros foram mais difundidos, por exemplo: CSS Layouts.

Site Responsivo: Um site responsivo é aquele que, independente do dispositivo que o usuário esteja acessando, seja ele móvel ou não, o site automaticamente se organizará para que a melhor experiência de uso seja proporcionada ao usuário.

Um dos principais responsáveis por essa padronização é o W3C (<http://www.w3.org/>), um órgão especializado e responsável por definir as padronizações na internet.