

Programação Web 01

Aula 07 - CSS: Posicionamento e Layout CSS

Roteiro da Aula

- Esquemas de posicionamento: absolute e relative
- Construção de layouts com divs e regras CSS



POSICIONAMENTO

Esquemas de posicionamento

- 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**
- A propriedade position deverá receber um dos valores referentes aos quatro tipos de posicionamento CSS.

Posicionamento Estático

- O posicionamento estático é definido pelo valor **static** da propriedade **position**.
- Assim: elemento { **position: static** }.
- Um elemento posicionado estaticamente segue o fluxo normal dos elementos da página, ou seja, se posiciona abaixo do elemento imediatamente anterior e acima do imediatamente posterior, quando nenhum destes está posicionado de outra forma que não a estática.
- Como o posicionamento estático é o padrão, não é necessário escrever regras para esse tipo de posicionamento.

Posicionamento Absoluto

- O posicionamento absoluto é definido pelo valor **absolute** da propriedade **position**.
- Assim: **elemento{ 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**.

Posicionamento Absoluto

- 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.

```
img
{
position:absolute;
top:100px;
left:100px;
}
```



Posicionamento Absoluto

- Podemos também alterar as referências do elemento posicionado, mudando as propriedades para **right**(direita) e **bottom** (inferior). Dessa forma, podemos fazer combinações para os quatro cantos da tela.
- Qualquer unidade de medida CSS pode ser usada para posicionar, inclusive medidas relativas como são aquelas expressas em porcentagens.

Posicionamento Absoluto

- Considere a marcação HTML a seguir. Para marcação HTML foi desenvolvida duas regras CSS.
- Marcação HTML

```
  

```

- Regra CSS

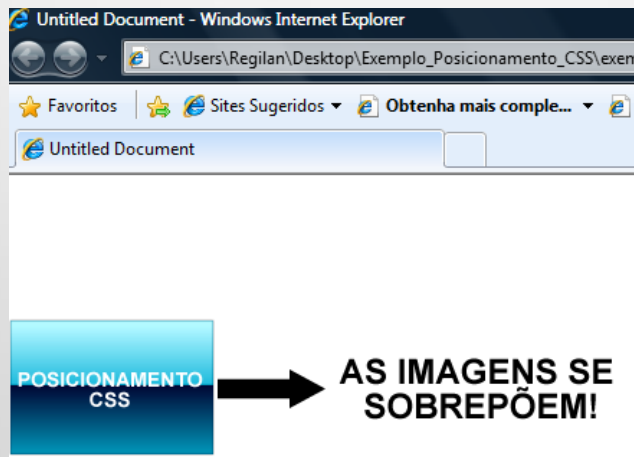
```
#pos1  
{position: absolute;  
top: 100px;  
left: 0px;}
```

```
#pos2 {  
position: absolute;  
top: 100px;  
left: 0px;}
```

- O que será exibido com as regras acima?

Posicionamento Absoluto

- Como as duas id criadas possuem posicionamento absoluto, e o posicionamento a esquerda é de 100px, as imagens se sobrepõem, ou seja são posicionadas uma em cima da outra.



Posicionamento Absoluto

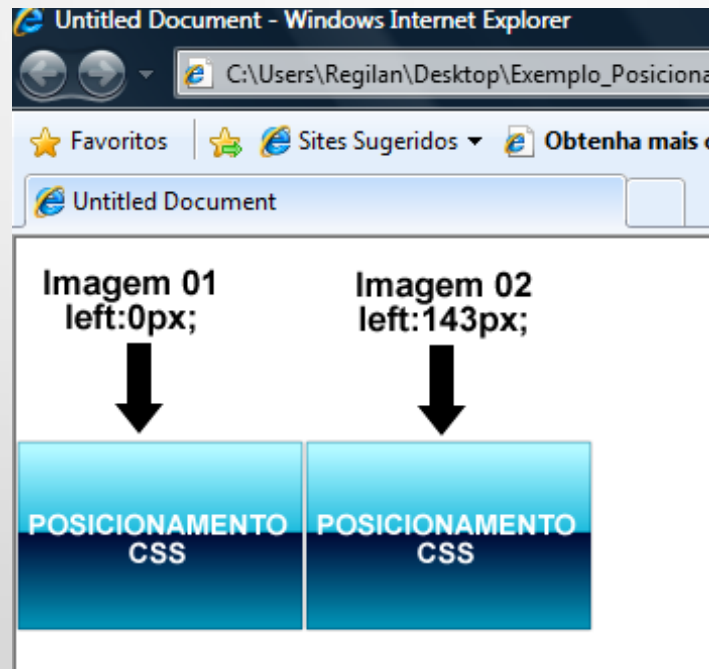
- Para que as imagens não sejam sobrepostas, podemos alterar o código CSS para a seguinte forma:

```
#pos2 {  
position: absolute;  
top: 100px;  
left: 143px; }
```

- Como a imagem apresenta largura de 143px, o posicionamento a esquerda da segunda imagem estará configurado com 143px, de forma que aparecerá logo em seguida a apresentação da primeira imagem. Veja o resultado no slide a seguir!

Posicionamento Absoluto

- Posicionamento Absoluto:



Posicionamento Relativo

- Especificando **position: relative** podemos utilizar **top** ou **bottom**, **right** ou **left** para posicionar os elementos na página em relação ao lugar que ele ocuparia no fluxo do documento. Veja a regra CSS para a posição **relative** para o exemplo apresentado anteriormente.

```
#pos1{  
position: relative;  
top: 100px;  
left: 0px; }
```

```
#pos2{  
position: relative;  
top: 100px;  
left: 0px; }
```

- O que será exibido com as regras acima?

Posicionamento Relativo

- Quando foi utilizado o posicionamento absoluto, as imagens apareceram sobrepostas, já que elas estavam posicionadas a `0px` do elemento que as contém(o **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(imagem1).
- Sendo assim, teremos o seguinte resultado apresentado no slide a seguir.

Posicionamento Relativo



Posicionamento Relativo + Absoluto

- Considere a seguinte marcação HTML:

```
<div id="div1">  
<div id="div2"></div>  
</div>
```

- Se definirmos uma regra CSS **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 relação ao **body** da página.
- Dessa forma ao configurarmos as propriedades: **left, top, bottom e right**, elas serão configuradas em relação a **div1**.

Posicionamento Fixo

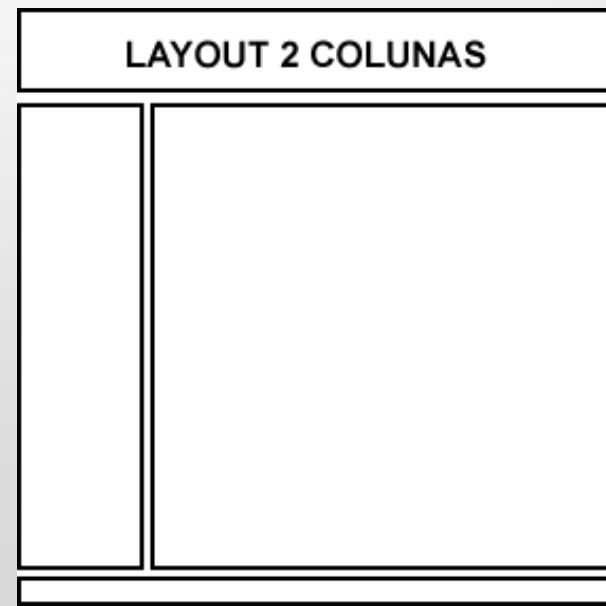
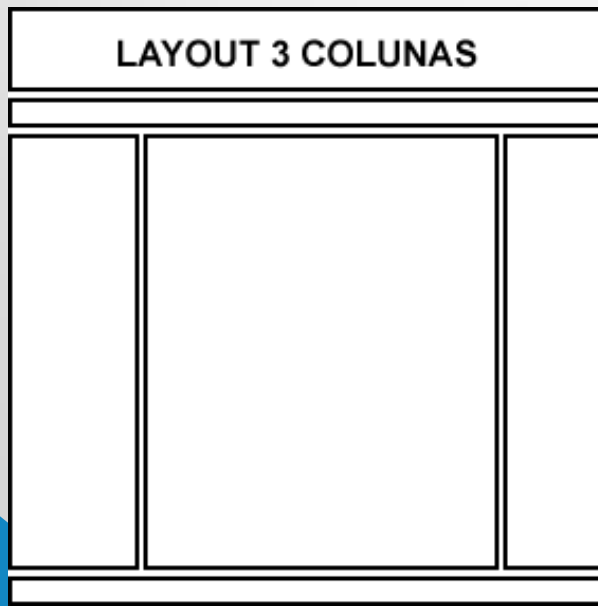
- O esquema de posicionamento FIXO, posiciona um elemento fixadamente na página em relação a janela, ou seja, independente de a página conter um scroll ou não, o elemento sempre ficará visível ao usuário em relação as medidas definidas.
- O posicionamento fixo é definido pelo valor **fixed** da propriedade **position**. Assim: elemento **{position:fixed }**.
- Da mesma forma que o posicionamento relativo e absoluto, deverá também ser configurado as propriedades **left, right, top e bottom**.



LAYOUT

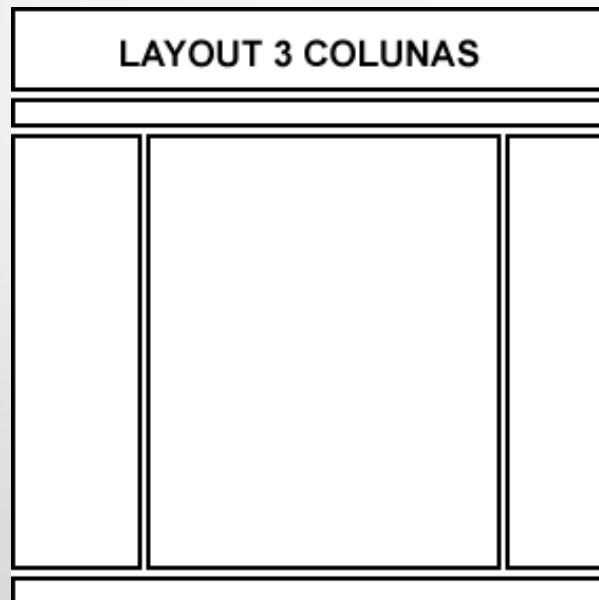
Layout CSS

- O primeiro passo para construir um layout css é definir como será o design da página. O design de uma página pode ser feito em ferramentas(Fireworks, CorelDrawn, Photoshop) apropriadas pra criação de design de páginas WEB ou mesmo em uma folha de papel.



Layout CSS

- Em nosso exemplo construiremos um layout 3 colunas.



Layout CSS

- A estrutura HTML para definir as seções do layout consistirá de divs e para cada div será definida regras CSS para posicioná-las.

```
<div id="containerGeral">  
  <div id="cabecalho"> </div>  
  <div id="menu"> </div>  
  <div id="le"> </div>  
  <div id="centro"> </div>  
  <div id="ld"> </div>  
  <div id="rodape"> </div>  
</div>
```

Layout CSS

- O código do containerGeral e do cabeçalho ficará assim:

```
#containerGeral
{
    position:relative;
    width:800px;
    height:auto;
    border:1px solid #000;
    margin:0 auto;
}

#cabecalho
{
    width:800px;
    height:130px;
    background-color:#06F;
}
```

Layout CSS

- O containerGeral deverá ter uma regra de posicionamento. Usaremos o posicionamento do fluxo do documento, posicionamento relative.
- A regra margin:0 auto; configurará a margem superior e inferior para o valor 0 e a esquerda e direita para automática, neste caso a página ficará centralizada.

Layout CSS

- O menu, rodapé e lateral esquerda ficará assim:

```
#menu, #rodape
{
    width: 800px;
    height: 30px;
    background-color: #0F6;
}

#le
{
    width: 200px;
    height: 500px;
    background-color: #F00;
}
```


Layout CSS

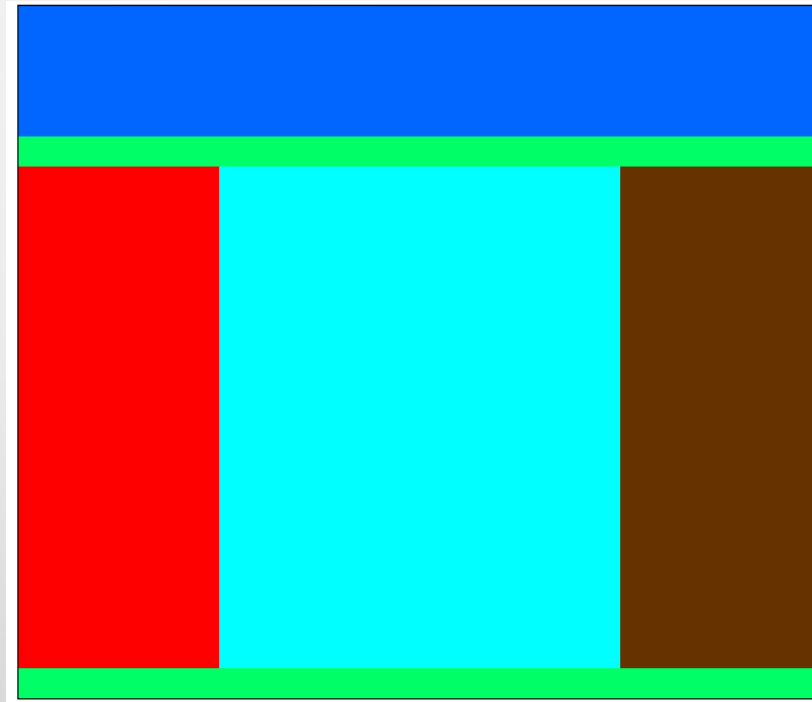
- O centro e a lateral direita possui a seguinte codificação CSS

```
#centro
{
    width:400px;
    height:500px;
    background-color:#0FF;
    position:absolute;
    top:160px;
    left:200px;
}
```

```
#ld
{
    width:200px;
    height:500px;
    background-color:#630;
    position:absolute;
    top:160px;
    left:600px;
}
```

Layout CSS

Resultado:



Tags semânticas – HTML 5

- No exemplo mostrado no slide anterior, utilizamos `<div>` e uma ID específica para cada parte do layout.
- O HTML 5 oferece uma série de tags semânticas para cada região do site.
- Exemplos: `<header>`, `<nav>`, `<section>`, `<footer>`, `<article>`, etc.
- Um elemento `<div id="topo">` pode ser substituído por `<header>`.
- Um elemento `<div id="rodape">` pode ser substituído por `<footer>`

Próxima Aula

- Atividade prática
- Estilização de textos e pseudo-elementos

