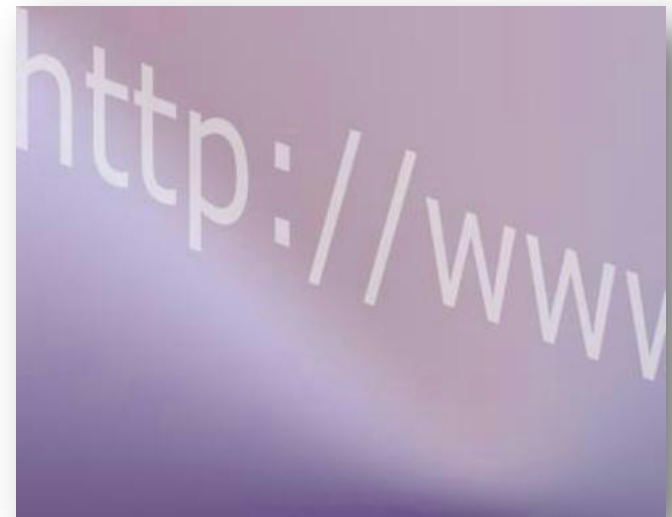


# Introdução ao CSS

# Estilos de Páginas

- CSS – Cascading Style Sheets:
  - Estilos de Página em Cascata (trad. livre);
  - É uma linguagem de estilos
  - Define a apresentação de documentos:
    - HTML;
    - XHTML;
    - XML;
    - Outros.



# Histórico

- Originalmente, as configurações referente a apresentação de elementos em uma página, podiam ser feitas através de atributos HTML 3.2;
  - Nesta abordagem, as páginas tinham a complexidade de código exponencial!
- Nove linguagens diferentes apareceram:
  - W3C escolheu definiu apenas duas:
    - CHSS (Cascated HTML Style Sheet);
    - SSP (Stream-based Style Sheet Proposal);

# Histórico

- Seus autores uniram esforços e, juntos, iniciaram o padrão CSS:
  - A letra “H” foi removida do CHSS devido a sua aplicação em outras tecnologias e não apenas ao HTML;
  - 1996: primeira especificação do CSS 1.0:
    - Empregado parcialmente no Internet Explorer 3.0;
    - Empregado plenamente no Internet Explorer 5.0 em 2000;

# Histórico

- Enquanto a Microsoft seguia o CSS, a Netscape apresentava uma alternativa chamada:
  - JSSS: JavaScript Style Sheets;
- CSS 2.0 em 1997;
- CSS 2.1 em 2007;
- CSS 3.0 sem Recomendação ainda;
- CSS 4.0 apenas discussão.

# Histórico

- A partir do HTML 4, e principalmente com o HTML 5, os atributos de apresentação passam a não serem mais exigidos ou suportados para dar lugar a configuração de apresentação mais dinâmicas: CSS.



# Vantagens

- Separação de Conteúdo e Apresentação;
- Consistência de Abrangência;
- Menor Largura de Banda;
- Reformatação mais Fácil;
- Acessibilidade.

# Limitações Básicas

- Limitações no controle vertical;
- Ausência de expressões aritméticas.



# Sintaxe

- Um regra CSS é dividida em duas partes:
  - *Propriedade/valor podem ser escritos em linhas separadas.*



Valor

Propriedade

# Criando um Arquivo

- Crie um novo arquivo:
  - Nome “*estilo.css*”;
- Vá até seu arquivo “curriculo.html”:
  - Insira dentro da seção <head>:  

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```
- Você terá um vínculo criado, entre sua página e um estilo.

# Primeiras Regras

- No arquivo “estilo.css”:  
body  
{  
background-color: LightBlue;  
}  
iframe  
{  
background-color: LightYellow;  
}

# Cores

- Existem várias formas de definir cores;
  - Algumas com nomes (em inglês);
  - Outra com base em RGB:
    - Vermelho, Verde e Azul;
    - A combinação de três cores primárias gera outras cores;
  - Representado por #RRGGBB:
    - RR = Vermelho de 00 à FF (0 – 255 em Hex);
    - GG = Verde de 00 à FF;
    - BB = Azul de 00 à FF;
- Ver Paint.NET

# Cores

- Existem várias formas de definir cores;
  - Exemplo para Vermelho:
    - `color:rgb(255,0,0);`
    - `color:#FF0000;`
    - `color:red;`

# Criando as Regras

- As regras podem ser criadas:
  - Em arquivo CSS (externas);
  - Como conteúdo da tag `<style>` (internas);
  - Como atributos (inline).

# Múltiplas Regras

- Quando tivermos regras definidas em diferentes lugares:
  - As propriedades que não forem repetidas, serão agregadas ao estilo (cascadeamento);
  - Se houverem propriedades repetidas, será mantida apenas uma regra, na ordem:
    1. *Inline*;
    2. Interna;
    3. Externa;
    4. Padrão do Browser.

# Seções HTML

- Dividindo seções de estilos diferentes no código HTML:

```
<div style="color:#0000FF">  
  <h3>Isto é um heading</h3>  
  <p>Isto é um parágrafo</p>  
</div>
```



Introdução ao CSS

# BACKGROUND



# Background

- Propriedades para background (fundo):
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

# background-color

- Valor padrão: “*transparent*”

**body**

**{**

**background-color:#00FF80;**

**}**

# Criando um Gradiente

- Abra o Paint.NET!
  - Escolha uma cor primária;
  - Escolha uma cor secundária;
  - Utilize a ferramenta Gradiente (tecla G);
  - Salve esta imagem na pasta do projeto:
    - fundo.png

# background-image

```
body
{
    background-image:url('fundo.png');
}
```

# background-repeat

```
body
{
    background-image: url('fundo.png');
    background-repeat: repeat-x;
}
```

# background-repeat

```
body
{
    background-image:url('fundo.png');
    background-repeat:repeat-y;
}
```

# background-repeat

```
body
{
    background-image:url('fundo.png');
    background-repeat:no-repeat;
}
```



# background-position

```
body
{
    background-image:url('fundo.png');
    background-repeat:no-repeat;
    background-position:right top;
}
```

# background-attachment

```
body
{
    background-image:url('fundo.png');
    background-repeat:no-repeat;
    background-position:right top;
    background-attachment:fixed;
}
```

Introdução ao CSS

# HERANÇA ENTRE TAGS

# Herança entre Tags

- Vimos que as Tags são inseridas umas dentro das outras:
  - A Tag interior será considerada de nível inferior em relação a outra Tag que a contém;
  - Todas as propriedades CSS que não forem definidas para uma determinada Tag, serão herdadas de suas superiores.

テキスト

Introdução ao CSS

**TEXTO CSS**

# Propriedades do Texto

<b>color</b>	Define a cor do texto
<b>direction</b>	Especifica a direção de escrita do texto
<b>letter-spacing</b>	Altera o espaço entre os caracteres no texto
<b>line-height</b>	Define a altura da linha
<b>text-align</b>	Específica o alinhamento horizontal do texto
<b>text-decoration</b>	Específica a decoração adicionada ao texto
<b>text-indent</b>	Específica a indentação da primeira linha em um bloco de texto
<b>text-shadow</b>	Específica o efeito de sombra adicionado ao texto
<b>text-transform</b>	Controla a “capitalização” do texto
<b>vertical-align</b>	Define o alinhamento vertical do elemento
<b>white-space</b>	Especifica como um “espaço branco” é tratado dentro do elemento
<b>word-spacing</b>	Altera o espaço entre as palavras de um texto

# CSS - color

- `color:red;`
- `color:#00FF00;`
- `color:rgb(0,0,255);`

# CSS - text-align

**left**

**right**

**center**

**justify**



# CSS – text-decoration

none

~~underline~~

overline

~~line-through~~

blink

# CSS – text-transform

Capitalize

UPPERCASE

lowercase



# CSS – vertical-align

- text-top
- text-bottom
- middle
- sub            -            Subscrito;
- super        -            Sobrescrito;
- % - valor qualquer (10pt, -20pt, ...)
- [W3c Demo.](#)

Introdução a CSS

# CLASSES E IDENTIFICADORES

# Classes Específicas

- É possível criar diferentes classes de estilos para cada chave:
  - `h1.minhaclasse{color:#0000FF;}`
- Para usar esta classe CSS, utilizamos o atributo global **class** na Tag H1 desejada:
  - `<h1 class="minhaclasse">Dados Pessoais</h1>`
- Esta classe serve apenas para tags H1.

# Classes Genéricas

- É possível criar classes de uso mais abrangente. Estas classes não serão restritas a uma única chave:
  - `.centro{text-align:center;}`
- Uso:
  - `<h1 class="centro">Dados Pessoais</h1>`
  - `<p class="centro">Este é um parágrafo..</p>`
  - ...

# Identificadores

- Também é possível criar um estilo que será aplicado a todos os elementos que possuam uma identificação em comum:
  - Atributo global ID do HTML:
    - `<h1 id="TituloPrincipal">Currículo</h1>`
- Regra CSS:
  - `#TituloPrincipal {color:#FF00FF;text-align:center;}`

Introdução ao CSS

# **FONTES**



# CSS - Fontes

- Configurações mais comuns:
  - font-family;
  - font-style;
  - font-weight;
  - font-size;
    - Absolute;
    - Relative;

# CSS - Fontes

- Nomes de Famílias:

- Georgia, Palatino Linotype, Book Antiqua, Times New Roman, Arial, Helvetica, Arial Black, Impact, Lucida Sans Unicode, Tahoma, Verdana, Courier New, Lucida Console, e várias outras...

- Famílias Genéricas:

- Serif - Com “*tracinhos*”
- Sans Serif - Sem “*tracinhos*”
- Monospace - Largura das letras iguais



Sans-serif



Serif



Serif  
(red serifs)

# CSS - Fontes

- Exemplos Serif:
  - Times New Roman – Georgia
- Exemplos Sans-Serif:
  - Arial – Verdana
- Exemplos Monospace:
  - Courier New
  - Lucida Console

# CSS - Fontes

- Comando:
  - font-family: 'nome1', 'nome2', ..., genérica;
  - Onde:
    - 'nome1', 'nome2', ... São nomes de fontes desejados, na ordem de preferência. Caso o browser não disponha da primeira fonte, irá para a segunda e assim por diante;
    - genérica é o nome da família genérica. Caso o Browser não disponha de nenhuma fonte solicitada, irá buscar uma fonte da família genérica solicitada.
- Ferramenta para escolher fontes do Google:
  - [Google Web Fonts!](#)

# CSS - Fontes

- Estilo de Fonte:
  - font-style:
    - normal;
    - *italic*;
    - *oblique*;

# CSS - Fontes

- Peso da Fonte:

- font-weight:

- normal;

- **bold;**

- **bolder;**

- lighter;

# CSS - Fontes

- Tamanhos das Fontes:
  - Absoluto:
    - font-size:
      - Tamanho número: px, cm, etc...

# CSS - Fontes

- Tamanhos das Fontes:
  - Relativo:
    - font-size:
      - xx-small, x-small, small, medium, large, x-large, xx-large,...;
      - % - tamanho percentual em relação ao texto ascendente;
      - Tamanho número: Em (ex: 1em = 16px, 0.5em = 8px)
  - O tamanho padrão para fontes de parágrafos nos browser é 16px.



Introdução ao CSS

**LINKS**

# Links

- É possível personalizar a cor dos hiperlinks:
  - `a:link {color:#FF0000;}`
    - Link ainda não visitado;
  - `a:visited {color:#00FF00;}`
    - Link já visitado;
  - `a:hover {color:#FF00FF;}`
    - Mouse encima do link;
  - `a:active {color:#0000FF;}`
    - Link selecionado.

Introdução ao CSS

# LISTAS

# Lista não Ordenada <ul>

- list-style-type:
  - circle
  - disc
  - none
  - Square
- list-style-type:
  - list-style-image:url('img.png');

# Lista Ordenada <ol>

- list-style-type:
  - armenian
  - cjk-ideographic
  - decimal
  - decimal-leading-zero
  - georgian
  - hebrew
  - hiragana
  - hiragana-iroha
  - katakana
  - katakana-iroha
  - lower-alpha
  - lower-greek
  - lower-latin
  - lower-roman
  - none
  - upper-alpha
  - upper-latin
  - upper-roman

Introdução ao CSS

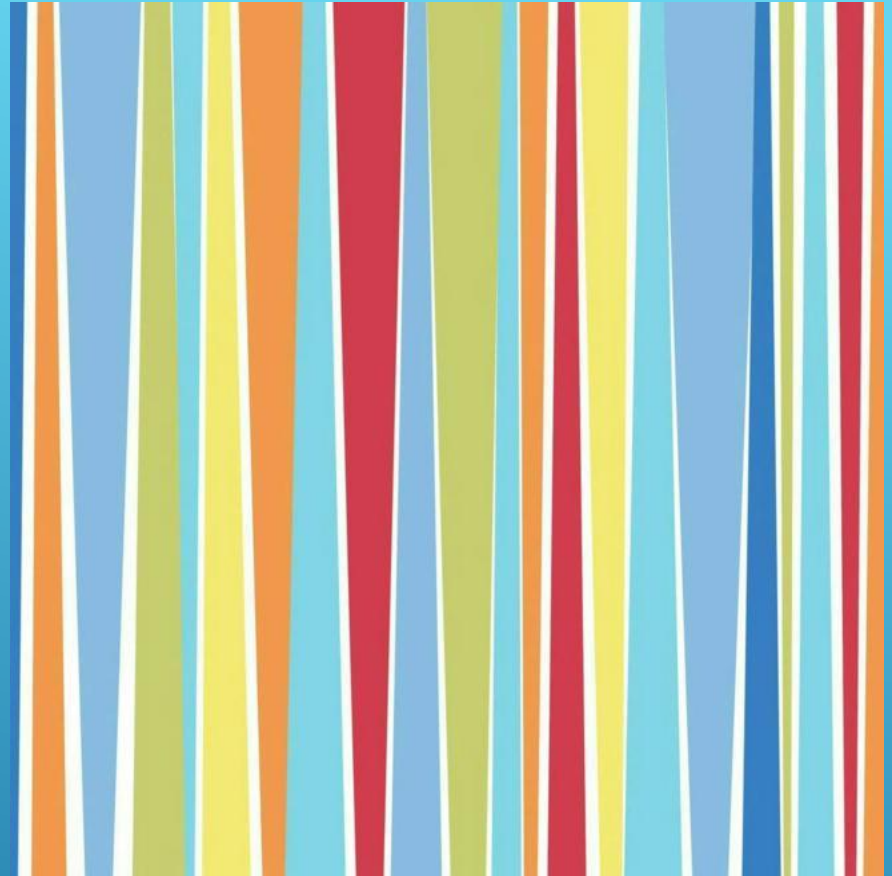
# TABELAS

# Tabelas

- Bordas: (table, td, th...)
  - border-width: 1px;
  - border-spacing: 2px;
  - border-style: solid;
    - outset, none, hidden, dotted, dashed, double, ridge, groove, inset;
  - border-color: #FF0000;
  - border-collapse: separate;
    - collapse;
- Ver ~~W3C~~ [sobre bordas individuais!](#)

Introdução ao CSS

# LAYOUTS





# Layouts

- O gerenciamento da disposição de conteúdo do HTML através do CSS é feito com o conceito de layouts.
  - Seções são separadas com a *tag* **<div>**, ou *tags* específicas:
    - **<section>**, **<aside>**, **<footer>**, **<article>**, **<nav>**...
  - Cada seção irá representar uma “caixa”;
  - Cada caixa terá um posicionamento, um estilo e um conteúdo;



# Exemplo



# layout.html

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <title>Teste com Layout</title>
```

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

```
  </head>
```

```
  <body>
```

```
    <div class="caixa1"> </div>
```

```
    <div class="caixa2"> </div>
```

```
    <div class="caixa3"> </div>
```

```
  </body>
```

```
</html>
```

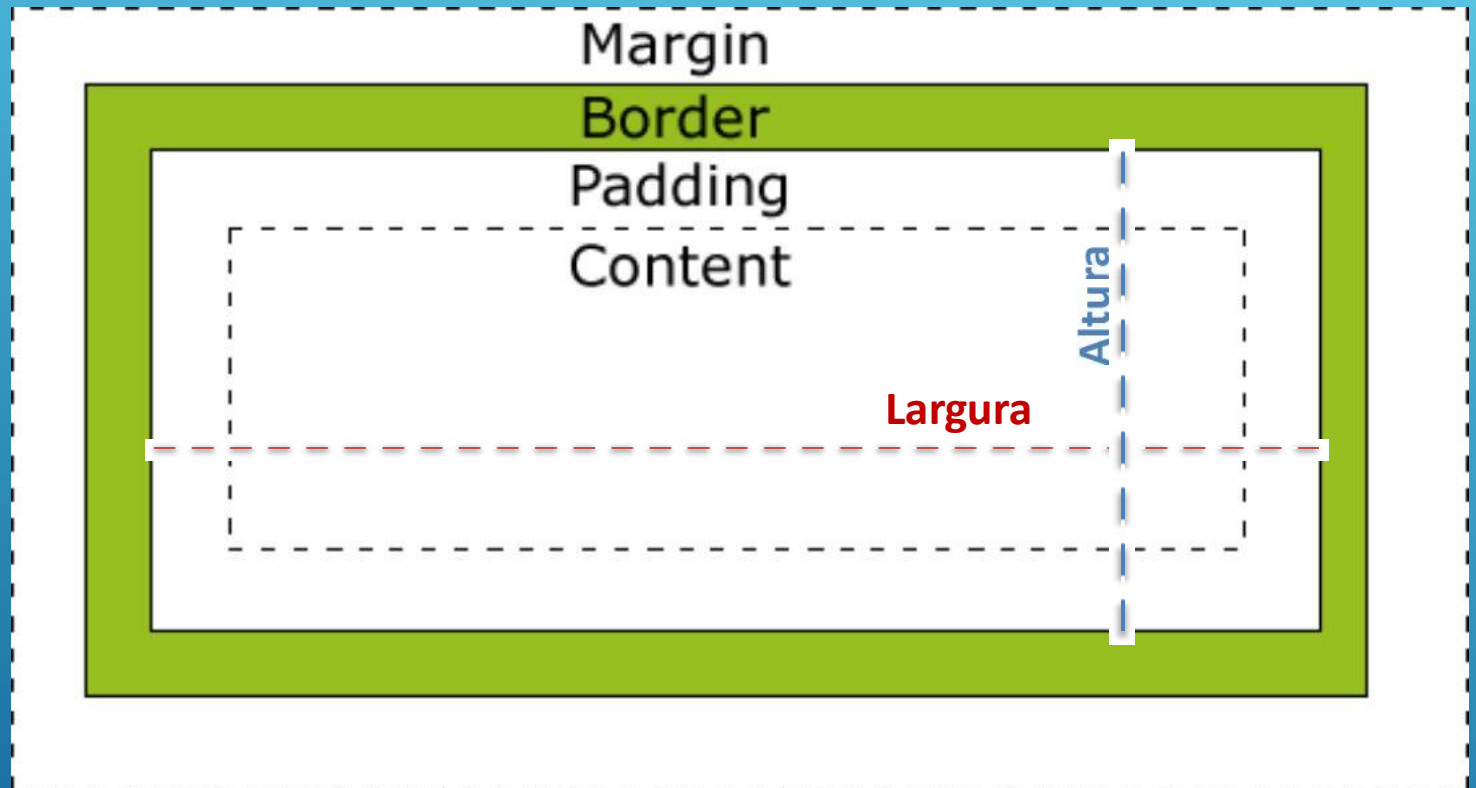
# estilo.css

```
div.caixa1 {  
    background-color: red;  
}
```

```
div.caixa2 {  
    background-color: green;  
}
```

```
div.caixa3 {  
    background-color: blue;  
}
```

# Box Model



# Box Model

- Tamanho da área de conteúdo:
  - Largura: (exemplos)
    - **width: 250px;**
    - **width: 50%;**
    - **max-width: 500px;**
  - Altura: (exemplos)
    - **height: 100px;**
    - **height: 40%;**
    - **max-height: 300px;**

# estilo.css

```
div.caixa1 {  
    background-color: red;  
    height: 100px;  
}
```

```
div.caixa2 {  
    background-color: green;  
    height: 50px;  
}
```

```
div.caixa3 {  
    background-color: blue;  
    height: 25px;  
}
```

# border-style

## border-style values:

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value



# border

- Largura da borda:
  - **border-width: 5px;**
- Cor da borda:
  - **border-color: red;**
- Bordas individuais:
  - **border-top-style: dotted;**
  - **border-right-style: solid;**
  - **border-left-style: solid;**
  - **border-bottom-style: dotted;**

# estilo.css

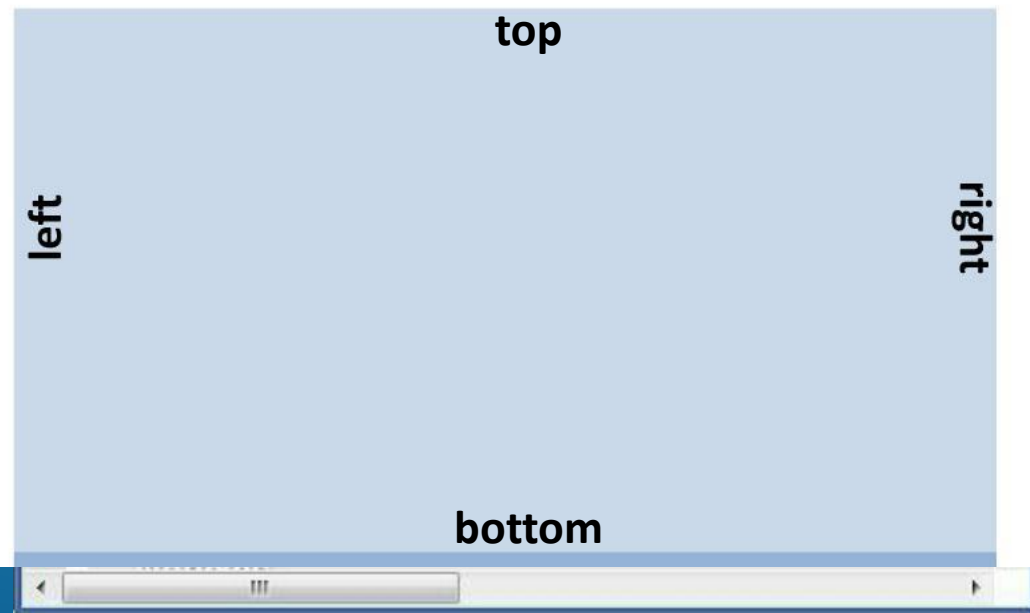
```
div.caixa1 {  
    background-color: red;  
    height: 100px;  
    border-style: dashed;  
    border-color: blue;  
    border-width: 1px;  
}
```

# estilo.css

```
div.caixa1 {  
    background-color: red;  
    height: 100px;  
    /*border-style: dashed;  
    border-color: blue;  
    border-width: 1px;*/  
    border: 1px dashed blue;  
}
```

# Posicionamento

- Podemos definir onde as “caixas” são colocadas na janela do navegador:



# position

- **position:static;**
  - Posição padrão. Fluxo normal da página.
- **position: fixed;**
  - Relativa a janela do navegador. Não move ao rolar a janela.
- **position: relative;**
  - Posição relativa às demais Tags inseridas.
- **position: absolute;**
  - Relativa ao primeiro elemento “pai” que tenha uma posição não static. Caso não haja, utiliza o <body>.

# position

- **left: 20px;**
  - Fixa a 20 pixels da borda esquerda.
- **right: 30px;**
  - Fixa a 30 pixels da borda direita.
- **top: 50px;**
  - Fixa a 50 pixels da borda superior.
- **bottom: 10px;**
  - Fixa a 10 pixels da borda inferior.

*Importante: lembre que tamanhos de janelas variam!*

# estilo.css

```
div.caixa1 {  
    background-color: red;  
    border: 1px dashed blue;  
    height: 100px;  
    width: 70%;  
}  
div.caixa2 {  
    background-color: green;  
    height: 50px;  
    width: 50%;  
}  
div.caixa3 {  
    background-color: blue;  
    height: 25px;  
    width: 90%;  
}
```

# estilo.css

```
div.caixa1 {  
    background-color: red;  
    border: 1px dashed blue;  
    height: 100px;  
    width: 70%;  
}  
div.caixa2 {  
    background-color: green;  
    height: 50px;  
    width: 50%;  
}  
div.caixa3 {  
    background-color: blue;  
    height: 25px;  
    width: 90%;  
    position: fixed;  
    bottom: 0px;  
}
```



# estilo.css

```
div.caixa1 {  
    background-color: red;  
    border: 1px dashed blue;  
    height: 700px;  
    width: 70%;  
}  
div.caixa2 {  
    background-color: green;  
    height: 500px;  
    width: 50%;  
}  
div.caixa3 {  
    background-color: rgba(0, 0, 255, 0.5);  
    height: 25px;  
    width: 90%;  
    position: fixed;  
    bottom: 0px;  
}
```

CSS 3

# Centralizando

- Para poder centralizar objetos do HTML, é necessário que estes objetos não possuam margens fixas e não tenha largura igual ao total da janela:
  - `margin-left: auto;`
  - `margin-right: auto;`
  - `width: 70%;`
  - `background-color: #FF8800;`

# estilo.css

```
div.caixa1 {  
    ...  
}  
div.caixa2 {  
    ...  
    margin-left: auto;  
    margin-right: auto;  
}  
div.caixa3 {  
    ...  
    right: 0;  
    left: 0;  
    margin-left: auto;  
    margin-right: auto;  
}
```

Elementos **fixed** ou **absolute** precisam ter as propriedades *right* e *left* definidos em 0 para poder centralizar

# Propriedades Notáveis

- z-index:
  - Número que define a ordem de exibição dos objetos. Quando objetos são desenhados em uma mesma área, define qual elemento será sobreposto ao outro.
- cursor:
  - auto, crosshair, default, e-resize, help, move, n-resize, ne-resize, nw-resize, pointer, progress, text, ....

# Layouts

- Além da documentação completa da W3C, este é um ótimo link com demonstrações de como gerenciar o layouts em CSS:
- <http://pt-br.learnlayout.com/>

# estilo.css

```
div.caixa1 {  
    background-color: red;  
    border: 1px dashed blue;  
    width: 70%;  
    height: 700px;  
    margin-left: 220px; /*Menu*/  
    margin-top: 25px; /*Barra*/  
}  
div.caixa2 {  
    background-color: green;  
    height: 500px;  
    width: 200px; /*Nova largura*/  
    margin-left: auto;  
    margin-right: auto;  
    position: absolute;  
    top: 0px; /*Encosta encima*/  
    margin-top: 25px; /*Barra*/  
}  
... /*Mantém estilo caixa3 como estava*/
```

# Responsividade

- Ver artigo:
  - “Criando websites para vários tipos de telas”.
- Estudar sobre consulta de mídia (*media queries*):
  - @media (max-wid

