

Desenvolvimento de Software para WEB

CSS

Prof. Regis Pires Magalhães

regismagalhaes@ufc.br - <http://bit.ly/ufcregis>

Referências

- Os slides a seguir mostram alguns exemplos da apostila da Caelum WD-43 Desenvolvimento Web com HTML, CSS e JavaScript.

CSS

- Cascading Style Sheets
- Folhas de Estilo em Cascata

CSS embutido no HTML (atributo style)

```
<p style="color: blue; background-color: yellow;">
```

O conteúdo desta tag será exibido em azul com fundo amarelo no navegador!

```
</p>
```

CSS tem uma sintaxe simples:

- declaração de propriedades
- valores separados por um sinal de dois pontos ":"
- cada propriedade é separada por um sinal de ponto e vírgula ";"

```
background-color: yellow;  
color: blue;
```

Regra CSS

- Possui três partes:
 - **Seletor**
 - elemento HTML identificado por sua tag, classe ou ID, e para o qual a regra será válida
 - **Propriedade**
 - atributo do elemento HTML ao qual será aplicada a regra
 - **Valor**
 - característica específica a ser assumida pela propriedade
- **Sintaxe:**
 - `seletor { propriedade: valor; }`

Tipos de CSS

- Tipos de vinculação de folhas de estilo
 1. Inline.
 2. Incorporadas.
 3. Importadas ou linkadas a partir de um documento externo.

CSS Interno (elemento style)

- Uma das grandes vantagens do CSS é manter o estilo fora do HTML.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Sobre a Mirror Fashion</title>
```

```
  <style>
```

```
    p {
```

```
      background-color: yellow;
```

```
      color: blue;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p>
```

```
    O conteúdo desta tag será exibido em azul com fundo amarelo!
```

```
  </p>
```

```
  <p>
```

```
    <strong>Também</strong> será exibido em azul com fundo amarelo!
```

```
  </p>
```

```
</body>
```

```
</html>
```

CSS Externo

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Sobre a Mirror Fashion</title>
```

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

```
</head>
```

```
<body>
```

```
<p>
```

O conteúdo desta tag será exibido em azul com fundo amarelo!

```
</p>
```

```
<p>
```

Também será exibido em azul com fundo amarelo!

```
</p>
```

```
</body>
```

```
</html>
```

css/estilos.css

```
p {  
  color: blue;  
  background-color: yellow;  
}
```


Seletor classe

- Exemplos

```
p.cor-um {  
  color:#000;  
}
```

```
p.cor-dois {  
  color:#00f;  
}
```

```
<p class="cor-um"> este parágrafo será na cor preta.</p>
```

```
<p class="cor-dois">este parágrafo será na cor azul.</p>
```

Classe aplicável a qualquer elemento HTML

```
.cor-tres {  
  color: #00f;  
}
```

Omite-se o nome do elemento antes da classe.

```
<h2 class="cor-tres">  
  Este cabeçalho é na cor azul.  
</h2>  
  
<p class="cor-tres">  
  Este parágrafo é na cor azul.  
</p>
```

Id aplicável a elemento HTML específico

```
#cor-tres {  
    color: #00f;  
}
```

```
<h2 id="cor-tres">  
    Este cabeçalho é na cor azul.  
</h2>
```

Só pode ser aplicado a UM elemento HTML dentro do documento.

```
#cabeçalho {  
    color: white;  
    text-align: center;  
}
```

Seletor hierárquico

```
#rodape img {  
  margin-right: 35px;  
  vertical-align: middle;  
  width: 94px;  
}
```

Aplicar apenas nos elementos **img** filhos do elemento com **id=rodape**.

Seletores identificam tags específicas

```
<div class="pageFrame" id="pageHead">  
  <h1>Welcome,  
    <span id="custName">Armando</span>  
      
  </h1>  
</div>
```

- Nome de tag: h1
- Nome de classe: .pageFrame
- ID do elemento: #pageHead
- Nome de tag e classe: div.pageFrame
- Nome de tag e id: img#welcome
- Relacionamento descendente: div #custName
- Atributos herdam dos padrões do navegador, a menos que sejam sobrescritos.

Exercício

- Que seletor CSS estiliza somente a palavra “bar”?

`<p class="a">foo,`

`bar</p>`

Agrupamento de seletores

- Vários seletores podem ser agrupados em uma regra CSS.

```
h1, h2, h3, h4, h5, h6 {  
    color: #0f0;  
}
```

Comentários

```
/* este é um comentário*/  
p {  
    font-size: 14px;           /* este é outro comentário*/  
    color: #000;  
    font-family: Arial, sans-serif;  
}
```


Famílias de Fontes

```
h1 {  
  font-family: serif;  
}
```

```
h2 {  
  font-family: sans-serif;  
}
```

```
p {  
  font-family: monospace;  
}
```

```
body {  
  font-family: "Arial", "Helvetica", sans-serif;  
}
```

Fontes

```
p {  
  font-family: monospace;  
  text-align: right;  
  line-height: 3px; /* tamanho da altura de cada linha */  
  letter-spacing: 3px; /* tamanho do espaço entre cada letra */  
  word-spacing: 5px; /* tamanho do espaço entre cada palavra */  
  text-indent: 30px; /* tam. da margem da primeira linha do texto */  
}
```

Imagem de fundo

```
h1 {  
  background-image: url(img/sobre-background.jpg);  
}
```

Bordas

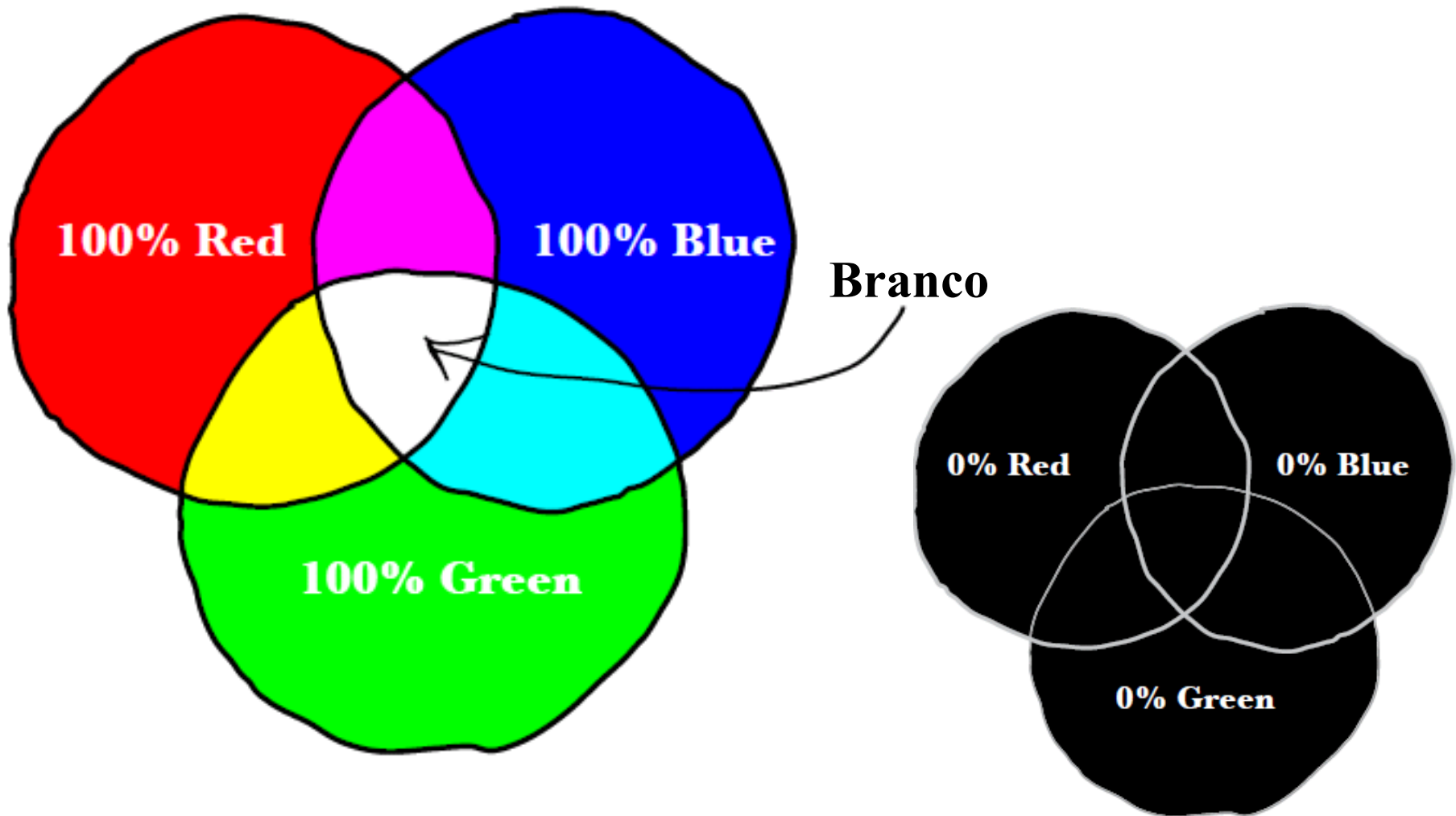
```
body {  
  border-color: red;  
  border-style: solid;  
  border-width: 1px;  
}
```

```
body {  
  border: 1px solid red;  
}
```

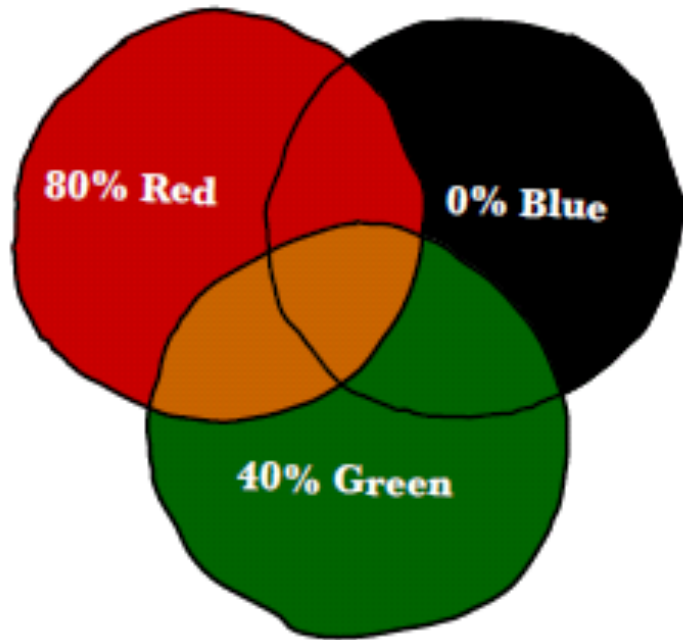
Cores

```
h1 {  
  color: red;  
}  
  
h2 {  
  background: yellow;  
}
```

Cores



Cores



`rgb (80%, 40%, 0%)`

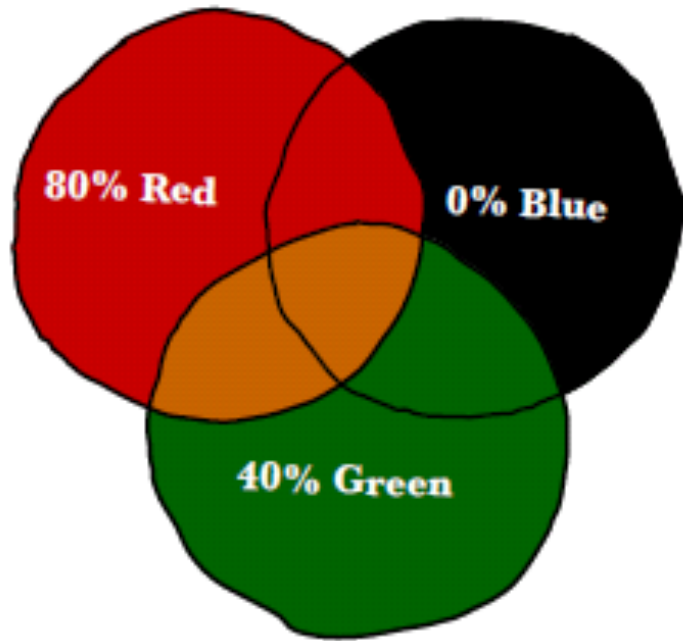
`rgb (204, 102, 0)`

`#CC6600`

Valores numéricos entre 0 e 255.

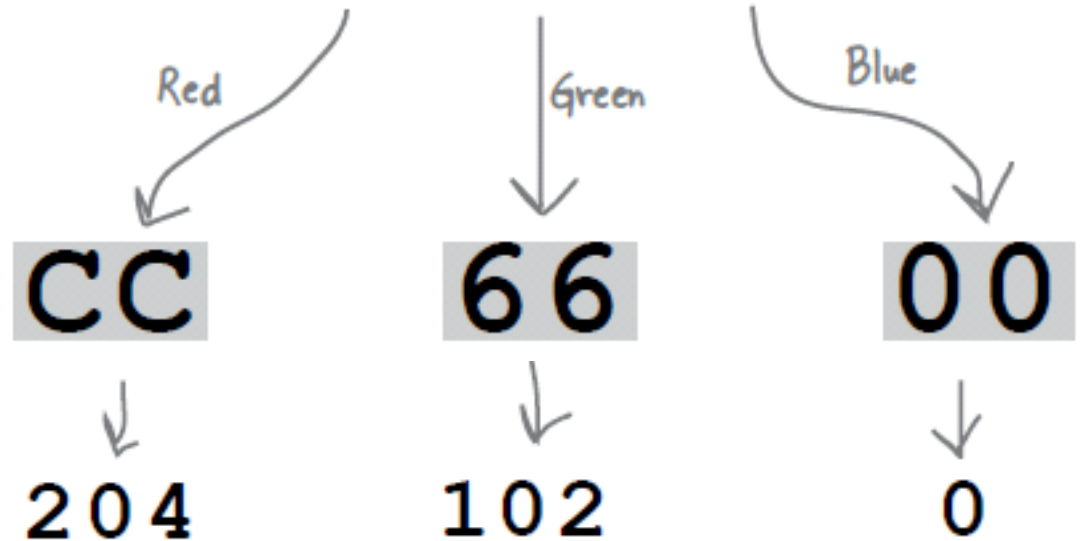
80% de 255 é 204,
40% de 255 é 102 e
0% de 255 é 0.

Cores



Cor em Hexadecimal

CC 66 00



$$\text{CC} = (12 \times 16) + 12 = 204$$

$$\text{66} = (6 \times 16) + 6 = 102$$

$$\text{00} = (0 \times 16) + 0 = 0$$

Cor pelo código hexadecimal

- #FFF = #FFFFFFF
- #CF9 = #CCFF99
- #cde = #ccddee
- #49c = #4499cc
- **Ferramenta online:** <http://www.colorschemer.com/online.html>

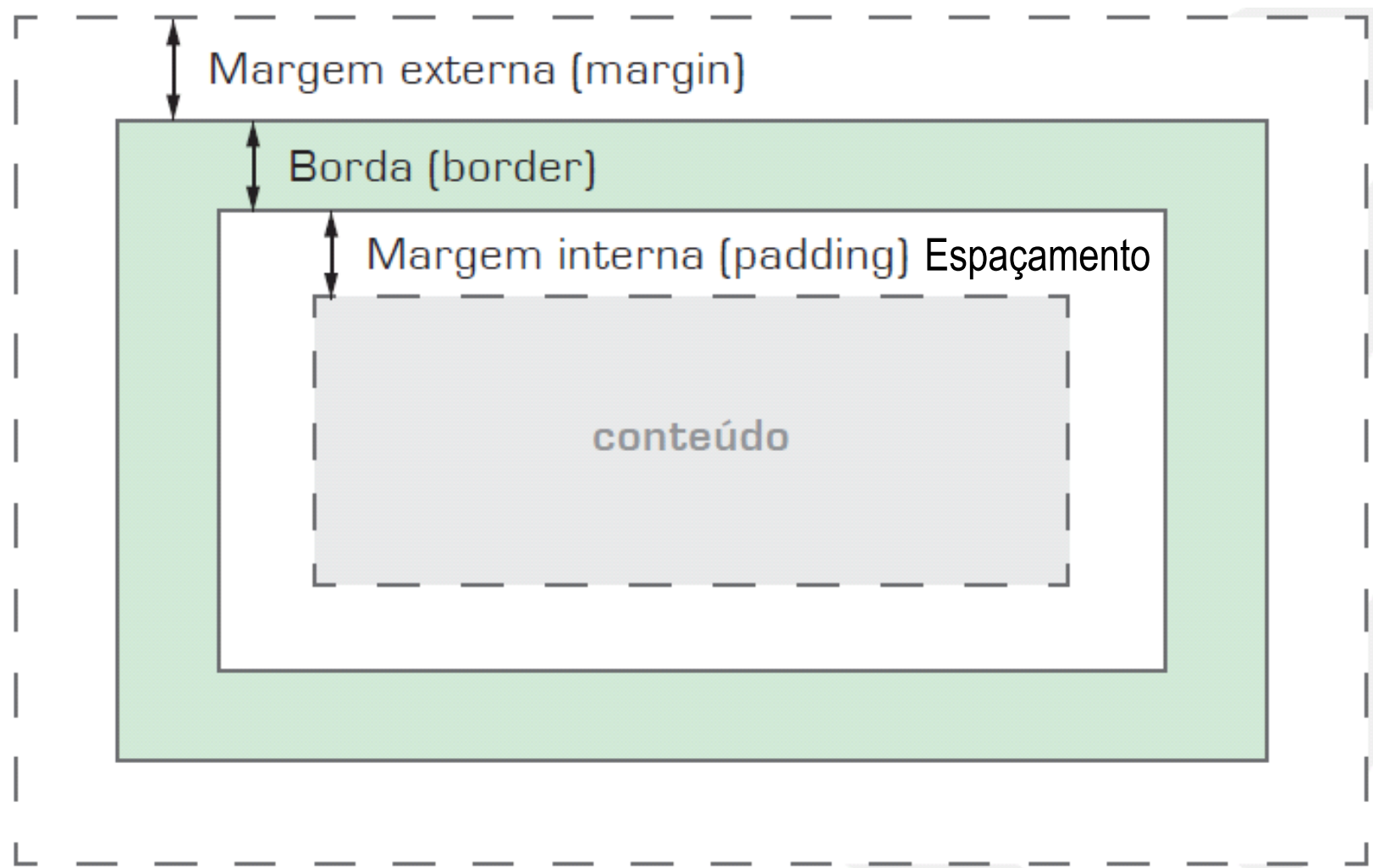
Cor por palavra-chave

- As Recomendações para CSS 2.1 listam as 17 cores a seguir:

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

Figura extraída das
Recomendações para
CSS2.1 no site do W3C
<http://www.w3.org>

Box model



Padding / Margin

```
p {  
  padding: 10px;  
  /* todas as direções */  
}  
  
p {  
  padding: 10px 15px;  
  /* 1º - padding-top e padding-bottom  
     2º - padding-right e padding-left */  
}  
  
p {  
  padding: 10px 20px 15px;  
  /* 1º - padding-top  
     2º - padding-right e padding-left  
     3º - padding-bottom */  
}  
  
p {  
  padding: 10px 20px 15px 5px;  
  /* sentido horário:  
     top, right, bottom, left */  
}
```

Unidades de medida

- Unidade relativa
 - tomada em relação à outra medida.
 - valor é tomado em relação:
 - **em**: ...ao tamanho da fonte ('font-size') herdada;
 - **ex**: ...a altura da letra x (xis) da fonte herdada;
 - **px - pixel**: ...ao dispositivo (midia) de exibição;
 - **%**: ...à percentagem de uma medida previamente definida.
- Unidade absoluta
 - não referenciada a qualquer outra unidade e nem é herdada.
 - usadas quando as mídias de exibição são perfeitamente conhecidas
 - cm, in, mm, etc.

Dimensões

```
p {  
  background-color: red;  
  height: 300px;  
  width: 300px;  
}
```

Seletores avançados

```
/* Seletor de irmãos */
h2 ~ p {
    font-style: italic;
}

/* Seletor de irmão adjacente */
h2 + p {
    font-variant: small-caps;
}

/* Seletor de filho direto */
article > h1 {
    color: blue;
}

/* Negação */
p:not(.especial) {
    color: gray;
}
```

```
<article>
  <h1>Título</h1>
  <p>Início</p>
  <h2>Subtítulo</h2>
  <p>Texto</p>
  <p>Mais texto</p>
  <p class="especial">
    Texto especial</p>
  <section>
    <h1>Título da seção</h1>
  </section>
</article>
```

Pseudo classes

São como classes CSS já pré-definidas.

É como se o navegador já colocasse certas classes por padrão em certos elementos.

```
<ul>
  <li>Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
  <li>Quarto item</li>
</ul>
```

```
li:first-child {
  color: red;
}
```

```
li:last-child {
  color: blue;
}
```

```
li:nth-child(2n) {
  color: green;
} /* elementos pares */
```

```
li:nth-child(2n+1) {
  color: blue;
} /* elementos impares */
```


Links

A hypertext link



Link não visitado

A hypertext link



Link ativo (clicado)

A hypertext link



Link visitado

unvisited link



hover state



active state



visited state



```
a:link {color: #696; text-decoration: none;
background-color: transparent}
a:visited {color: #699; text-decoration:
none; background-color: transparent}
a:hover {color: #c93; text-decoration:
underline; background-color: transparent}
a:active { color: #900; text-decoration:
underline; background-color: transparent }
```

Pseudo-classes

- Podemos usar hover em todo tipo de elemento, não apenas links.

Pseudo elementos

- Pseudo classes ajudam a selecionar elementos com certas características sem que seja preciso criar uma classe manualmente neles.
- Pseudo elementos permitem selecionar certo tipo de conteúdo que não possuem elemento específico para ele.

```
<p>Curso de HTML e CSS</p>
```

```
p:first-letter {  
    font-size: 200%;  
}  
  
p:first-line {  
    font-style: italic;  
}
```

Inserção de conteúdos em elementos

Exemplo 1: Inserir colchetes em todos os links.

```
a:before {  
  content: '[';  
}  
  
a:after {  
  content: ' ]';  
}
```

```
[ Link 1 ]  
[ Link 2 ]  
[ Link 3 ]
```

Exemplo 2: Inserir o texto ‘(externo)’ após todos os links externos.

```
a[href^=http://]:after {  
  content: ' (externo)';  
}
```

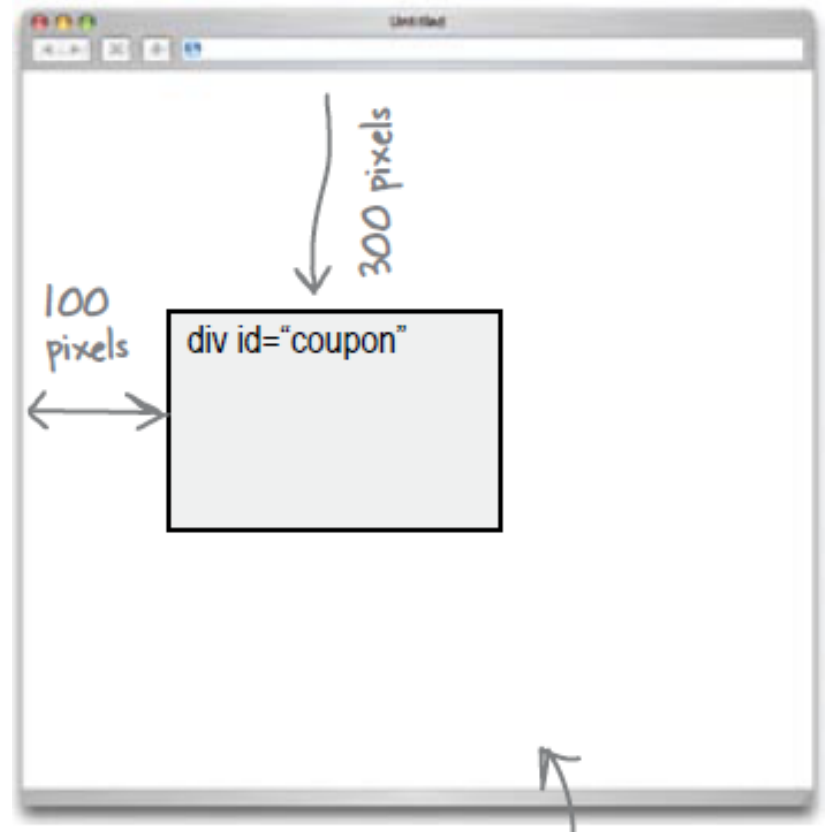
position - propriedades

- **static** (padrão)
 - Posiciona o elemento no fluxo normal da página e não permite ajustes externos.
- **absolute**
 - Posiciona o elemento em qualquer local da página.
- **fixed**
 - Posiciona o elemento relativo à janela do navegador e não move quando ocorre rolagem da página.
- **relative**
 - O elemento colocado em sua posição normal na página e, depois, é deslocado de acordo com os valores definidos, deixando vazio o espaço onde ele ficaria.
- **inherit**
 - Valor da posição herdado do elemento pai.

position

- fixed

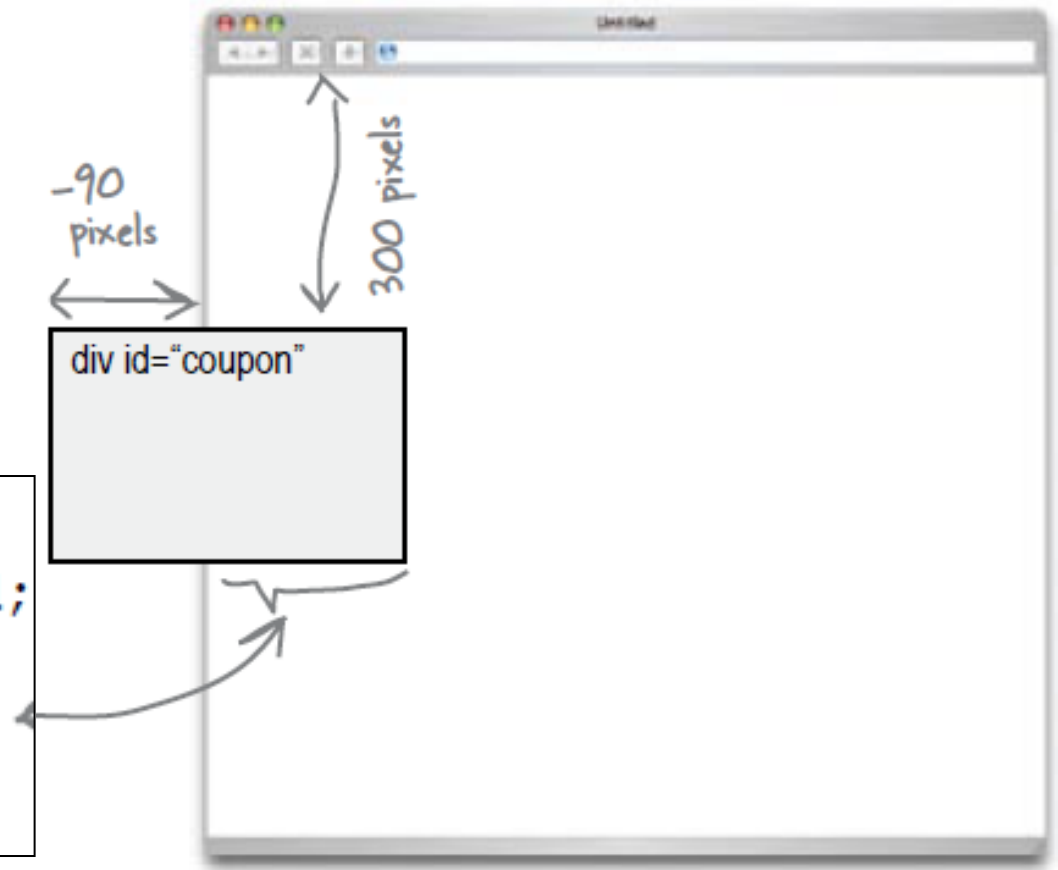
```
#coupon {  
    position: fixed;  
    top:      300px;  
    left:     100px;  
}
```



position

- fixed

```
#coupon {  
  position: fixed;  
  top: 300px;  
  left: -90px;  
}
```



z-index

- Elementos posicionados (**absolute**, **relative** ou **fixed**) podem ser colocados em camadas sobrepostas umas sobre as outras através da propriedade **z-index**.
- Permite definir a ordem de posicionamento dos elementos em uma pilha.

z-index

- Valores possíveis:
 - **auto** (padrão)
 - Valor igual ao dos elementos pais.
 - **valor**
 - Ordem do elemento na pilha.
 - Números negativos são permitidos.
 - **inherit**
 - z-index herdado do elemento pai.

z-index

- Exemplo:

```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

float

- Faz com que os elementos flutuem na página de acordo com determinada política
- Posicionamento float é uma solução flexível para layouts multi-colunas e permite que elementos “limpem” elementos ao seu lado.

float

- Um elemento float normalmente deve ter uma largura (**width**) específica declarada com um valor diferente de **'auto'**.
- Valores possíveis:
 - **left**
 - O elemento flutua para a esquerda.
 - **right**
 - O elemento flutua para a direita
 - **none** (padrão)
 - O elemento não flutua e será exibido de acordo com o fluxo normal do navegador.
 - **inherit**
 - O valor deve ser herdado do elemento pai.

float

Exemplo:

```
p {  
  float: left;  
}
```

Flutuar para a esquerda.

Deixa elementos um ao lado do outro.

clear

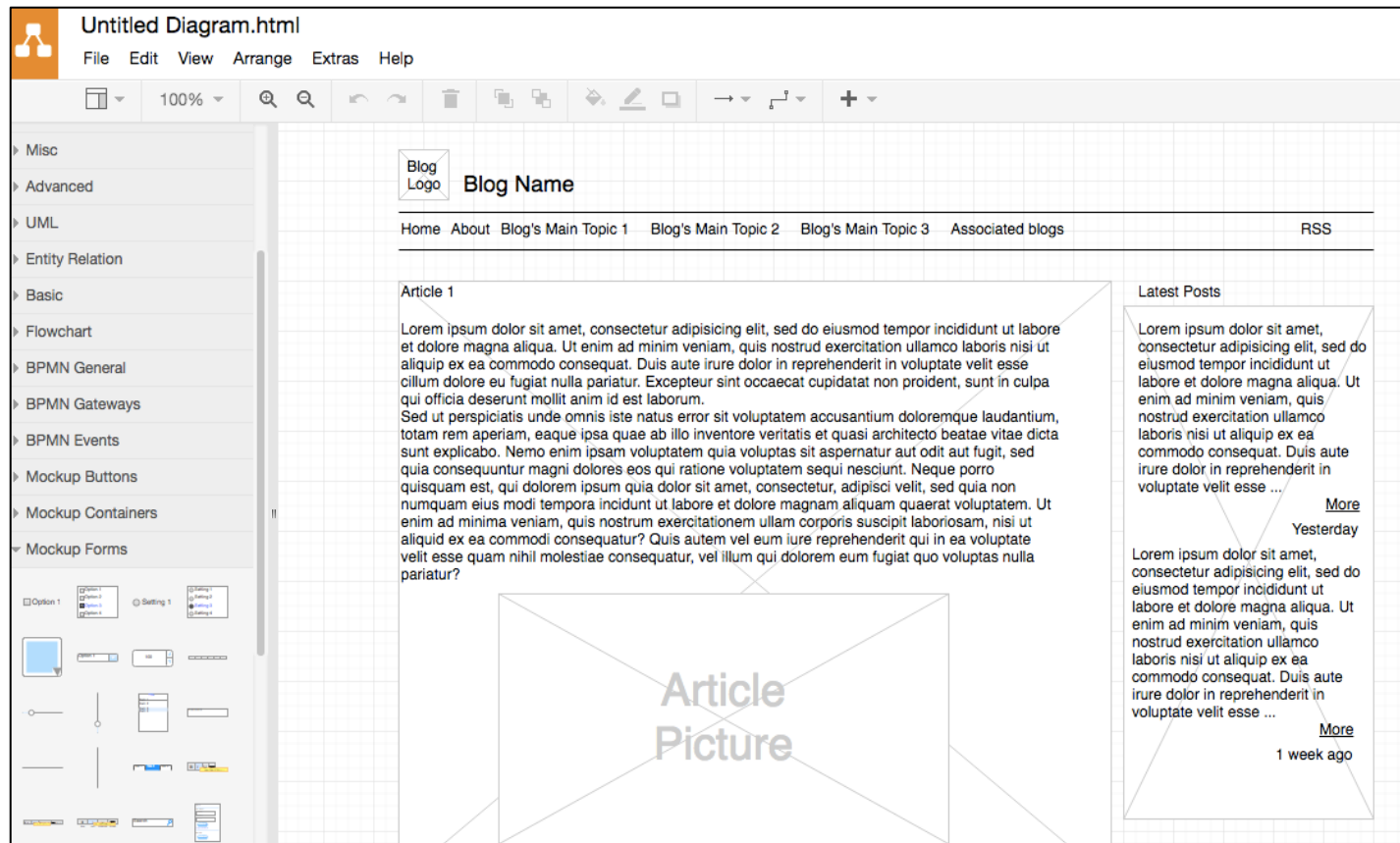
- Especifica que nenhum elemento float possa ficar à esquerda ou direita (ou ambos) de um elemento em bloco.
- O elemento com essa propriedade será movido para baixo até se livrar do elemento em bloco ao seu lado.

clear

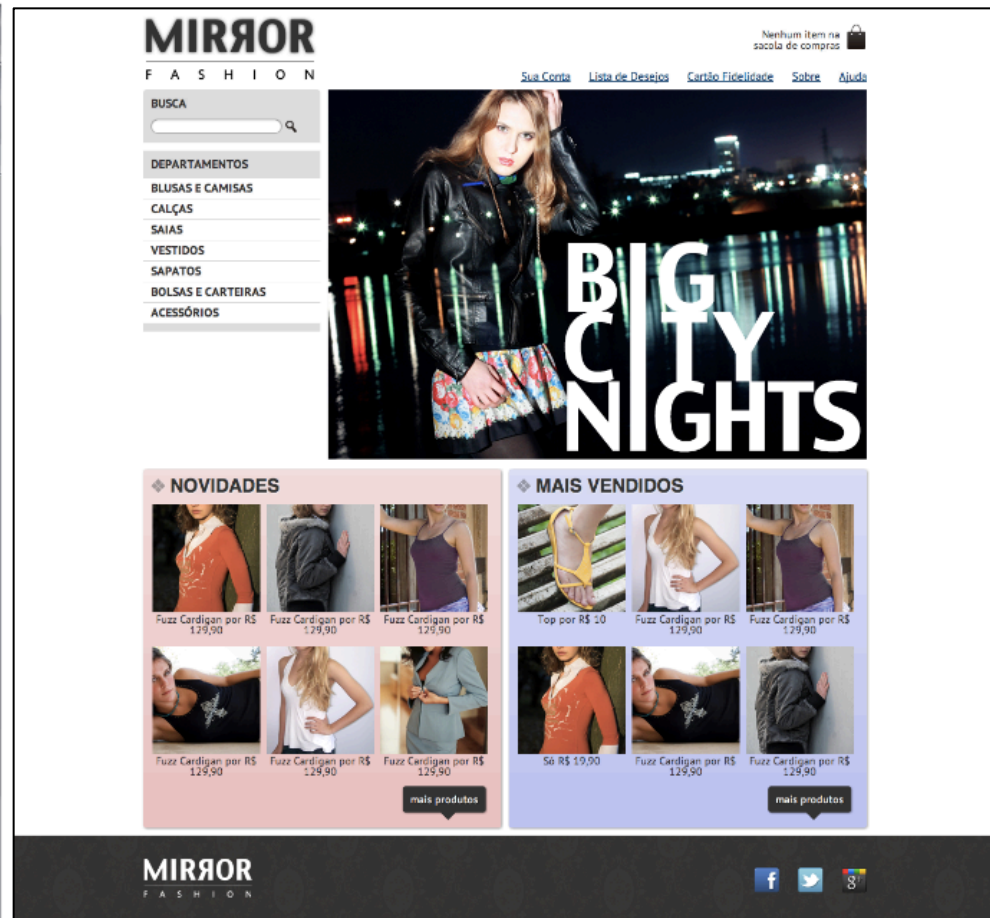
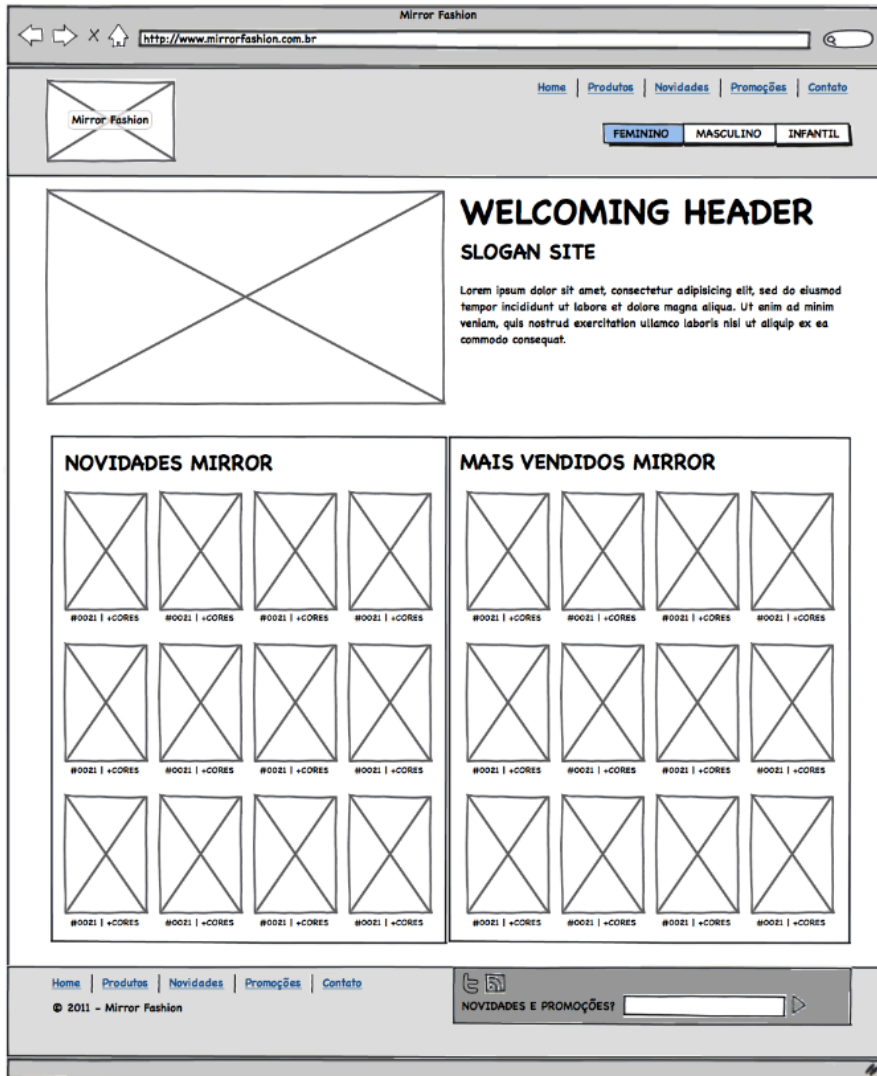
- Valores possíveis:
 - **left**
 - Não permite elementos float do lado esquerdo.
 - **right**
 - Não permite elementos float do lado direito.
 - **both**
 - Não permite elementos float em nenhum lado (esquerdo ou direito).
 - **none** (padrão)
 - permite elementos float em ambos os lados (esquerdo e direito)
 - **inherit**
 - Valor herdado do elemento pai.

Layout

- Wireframe
 - Ferramenta online
 - <http://www.draw.io>



Layout



Layout com tags semânticas

```
<body>

  <header>
    <!-- Conteúdo do cabeçalho -->
  </header>

  <section id="main">
    <!-- Conteúdo principal -->
  </section>

  <section id="destaques">
    <!-- Painéis com destaques -->
  </section>

  <footer>
    <!-- Conteúdo do rodapé -->
  </footer>

</body>
```

header, section e footer - são tags novas do HTML5.

Antes, teríamos criado apenas três div, uma para cada parte da página, e colocado ids diferentes pra cada uma.

A única diferença é o nome da tag e o significado que elas carregam.

Navegação

```
<nav class="menu-opcoes menu-cabecalho">
  <ul>
    <li><a href="#">Sua Conta</a></li>
    <li><a href="#">Lista de Desejos</a></li>
    <li><a href="#">Cartão Fidelidade</a></li>
    <li><a href="sobre.html">Sobre</a></li>
    <li><a href="#">Ajuda</a></li>
  </ul>
</nav>
```

```
.menu-opcoes {
  /* código de um menu de opcoes
     essas regras serão aplicadas ao nav */
}

.menu-cabecalho {
  /* código de um menu no cabeçalho
     essas regras TAMBÉM serão aplicadas ao nav */
}
```

Classe para diversos elementos

```
.container {  
  margin-right: auto;  
  margin-left: auto;  
}
```

```
<div class="info container">  
  <p>Conteúdo que deve ficar centralizado</p>  
</div>
```

CSS Reset

- Cada navegador tem estilos definidos por padrão, diferentes entre os navegadores.
- CSS Reset seta um valor básico para todas as características do CSS, sobrescrevendo totalmente os estilos padrão do navegador.
- Exemplos:
 - HTML5 Boilerplate
 - YUI3 CSS Reset
 - Eric Meyer CSS Reset
 - <http://meyerweb.com/eric/tools/css/reset/>

```
<link rel="stylesheet" href="css/reset.css">
```


Classificação dos layouts

- Número de colunas
- Largura da página
 - Fixa
 - A **largura é fixa** e não expande ou encolhe de acordo com a janela do navegador.
 - Líquida
 - O **conteúdo da página expande ou encolhe** para se ajustar à página de acordo com as dimensões da janela do navegador.
 - Elástica
 - A **largura do conteúdo é fixa**, mas as **margens expandem ou encolhem** de acordo com a janela do navegador.
 - **Conteúdo no centro da página.**

Layout

```
/* largura fixa */  
#container {  
    border: 1 px solid black;  
    width: 500px;  
}
```

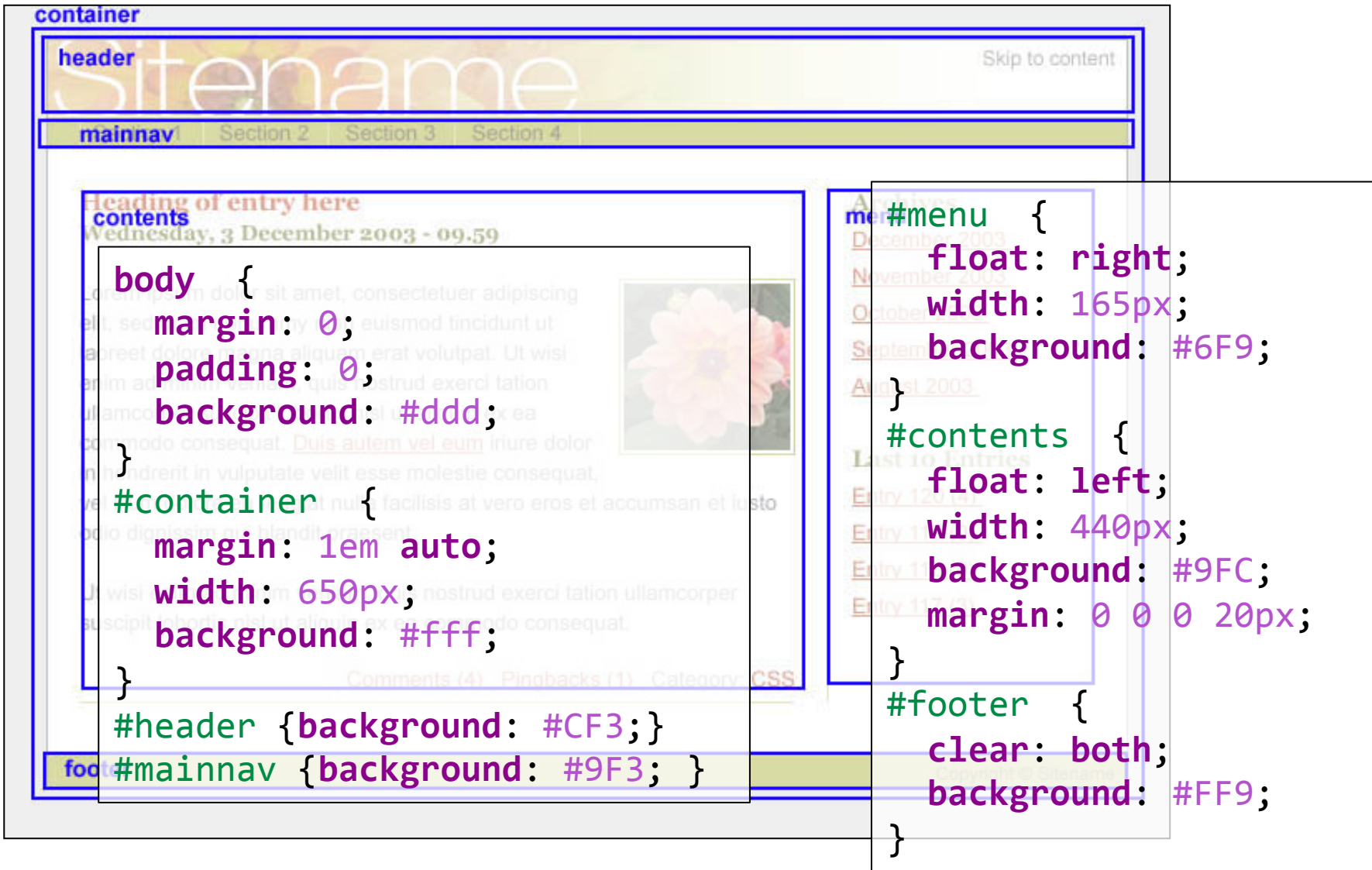
```
/* largura líquida */  
#container {  
    border: 1 px solid black;  
    width: 80%;  
    min-width: 300px;  
    min-height: 300px;  
}
```

```
#container {  
    border: 1 px solid black;  
    width: 40em;  
}
```

```
/* Centralização do layout */  
#container {  
    border: 1 px solid black;  
    width: 80%;  
    margin: 0 auto;  
}
```

```
#container {  
    border: 1 px solid black;  
    width: 500px;  
    margin: 0 auto;  
}
```


Layout – containers - css



The image shows a web page layout with several CSS rules overlaid, defining the styling for different containers. The layout includes a header, main navigation, a main content area, a sidebar menu, and a footer.

Containers and their CSS rules:

- body**
 - `margin: 0;`
 - `padding: 0;`
 - `background: #ddd;`
- #container**
 - `margin: 1em auto;`
 - `width: 650px;`
 - `background: #fff;`
- #header**
 - `background: #CF3;`
- #mainnav**
 - `background: #9F3;`
- #menu**
 - `float: right;`
 - `width: 165px;`
 - `background: #6F9;`
- #contents**
 - `float: left;`
 - `width: 440px;`
 - `background: #9FC;`
 - `margin: 0 0 0 20px;`
- #footer**
 - `clear: both;`
 - `background: #FF9;`

Page Content:

- Header:** Site name, Skip to content
- Mainnav:** Section 2, Section 3, Section 4
- Contents:** Heading of entry here, Wednesday, 3 December 2003 - 09.59
- Menu:** Last 10 Entries, Entry 120 (4), Entry 119 (3), Entry 118 (3), Entry 117 (3)
- Footer:** Comments (4), Pingbacks (1), Category: CSS

CSS para diferentes dispositivos

Modo 1: um arquivo css para cada css de dispositivo

```
<link rel="stylesheet" href="site.css" media="screen" />
<link rel="stylesheet" href="print.css" media="print" />
<link rel="stylesheet" href="handheld.css" media="handheld" />
```

Modo 2: um único arquivo com css de cada dispositivo

```
@media screen {
  body {
    background: blue;
    color: white;
  }
}

@media print {
  body {
    background: white;
    color: black;
  }
}
```

Media queries

```
<link rel="stylesheet" href="base.css" media="screen">  
<link rel="stylesheet" href="mobile.css" media="(max-width: 480px)">
```

```
@media screen {  
  body {  
    font-size: 16px;  
  }  
}  
  
@media (max-width: 480px) {  
  body {  
    font-size: 12px;  
  }  
}
```

Viewport

Ajuste para dispositivos com telas pequenas e grandes resoluções.
Celulares costumam ter uma resolução em dpi bem maior que Desktops.

```
<meta name="viewport" content="width=320">
```

```
<meta name="viewport" content="width=device-width">
```

Design responsivo

- A página deve se adaptar a diferentes condições, em especial a diferentes resoluções.
- Princípios básicos:
 - layout fluído usando medidas flexíveis, como porcentagens;
 - media queries para ajustes de design;
 - uso de imagens flexíveis.

Mobile First

- Abordagem simples e evolutiva.
- Evita desfazer trabalho já realizado.
- Começar o desenvolvimento pelo mobile e, depois, adicionar suporte a layouts desktop.
- Uso de media queries **min-width** ao invés de **max-width**, mais comum em códigos desktop-first.

Layout de 2 colunas

Cabeçalho da Página

Cursos

- XHTML
- CSS
- JavaScript
- jQuery
- JAVA
- SQL e Banco
- JSP e Servlet
- Scrum
- Portais Web

Serviços da Universidade XTI

Os vídeos são agrupados em cursos. Esses cursos são completos ou seja todo conteúdo sobre o assunto é apresentado. Para a elaboração de cada curso, a equipe da Universidade XTI adota em média 4 horas sobre o assunto para criar um curso de alta qualidade e verdadeiramente completo. Os cursos são formatados para atender a iniciantes e profissionais no assunto. No início de cada curso é apresentado uma contextualização do assunto para os iniciantes. Cada vídeo aula dura em média 15 minutos.

```
<div id="container">  
  <div id="header"></div>  
  <div id="left"></div>  
  <div id="content"></div>  
  <div id="footer"></div>  
</div>
```


Layout de 2 columnas - CSS

```
div {  
  border: 1px solid black;  
}  
  
#container {  
  width: 600px;  
  margin: 0 auto;  
}  
  
#header {  
  height: 80px;  
  background: #DDD;  
}  
  
#left {  
  float: left;  
  width: 130px;  
  background: #666;  
}
```

```
#content {  
  float: right;  
  width: 450px;  
}  
  
#footer {  
  clear: both;  
  background: #DDD;  
}
```


Layout de 3 columnas - CSS

```
div {  
    border: 1px solid black;  
}  
  
#container {  
    width: 600px;  
    margin: 0 auto;  
}  
  
#header {  
    height: 80px;  
    background: #DDD;  
}  
  
#left {  
    float: left;  
    width: 130px;  
    background: #666;  
}
```

```
#content {  
    float: right;  
    width: 450px;  
}  
  
#principal {  
    float: left;  
    width: 340px;  
}  
  
#right {  
    float: right;  
    width: 100px;  
}  
  
#footer {  
    clear: both;  
    background: #DDD;  
}
```

Bordas arredondadas e sombra

Bloco com Bordas

Arredondadas e Sombra

Utilizando Recursos do CSS 3

```
#container {  
    border: 2px solid;  
    width: 200px;  
    padding: 10px;  
  
    border-radius: 10px;  
    box-shadow: 10px 10px 5px #888;  
}
```

Opacidade e RGBA

```
p {  
  opacity: 0.3;  
}
```

```
p {  
  /* branco com 80% de opacidade */  
  color: rgba(255,255,255, 0.8);  
}
```

CSS3 Transforms

```
.header {  
  /* Move o elemento no eixo horizontal */  
  transform: translateX(50px);  
}
```



```
#main {  
  /* Move o elemento no eixo vertical */  
  transform: translateY(-20px);  
}
```

```
footer {  
  /* Move o elemento nos dois eixos (X, Y) */  
  transform: translate(40px, -20px);  
}
```

```
#menu-departamentos {  
  transform: rotate(-10deg);  
}
```



CSS3 Transforms

```
#novidades li {  
  /* Alterar a escala total do elemento */  
  transform: scale(1.2);  
}
```



```
#mais-vendidos li {  
  /* Alterar a escala vertical e horizontal do elemento */  
  transform: scale(1, 0.6);  
}
```

```
footer {  
  /* Distorcer o elemento no eixo horizontal */  
  transform: skewX(10deg);  
}
```



```
#social {  
  /* Distorcer o elemento no eixo vertical */  
  transform: skewY(10deg);  
}
```

```
html {  
  /* Combinar transformações no mesmo elemento */  
  transform: rotate(-30deg) scale(0.4);  
}
```

Aumentar o elemento em 20%, dando uma espécie de zoom

```
.painel li:hover {  
  transform: scale(1.2);  
}
```

```
.painel li:hover {  
  transform: scale(1.2) rotate(-5deg);  
}
```


Transições CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    transition: width 2s;
}
div:hover {
    width: 300px;
}
</style>
</head>
<body>
<p><b>Note:</b> This example does not work in Internet Explorer 9 and
earlier versions.</p>
<div></div>
<p>Hover over the div element above, to see the transition effect.</p>
</body>
</html>
```

Pré-processadores CSS



- Linguagens baseadas em CSS (mesma ideia, sintaxe semelhante) com recursos que fazem falta no CSS.
 - Variáveis
 - Suporte a operações aritméticas
 - Sintaxe mais compacta para representar hierarquias
 - Mixins.
- Chamado de pré-processador, pois é usado para gerar um arquivo CSS no final.
- Os mais usados são: **Sass**, LESS e Stylus.
- Ver: **Caelum WD-43 – Cap. 15 – Apêndice LESS**

Referências e Links importantes

- Caelum WD-43 - **Desenvolvimento Web com HTML, CSS e JavaScript.**
Disponível em:
<https://www.caelum.com.br/apostila-html-css-javascript/>
- Vídeos da Universidade XTI
- www.w3schools.com
- FREEMAN Eric & FREEMAN Elisabeth. **Use a Cabeça! HTML com CSS e XHTML.** Alta Books, 1ª Edição, 2006.
- Site sobre CSS e Padrões Web. Por Maurício “Maujor” Samy Silva. Disponível em: **http://www.maujor.com/**
- Apostila da K19 - **Desenvolvimento Web com HTML, CSS e Javascript.**

Referências e Links importantes

- SILVA, Maurício Samy. **Construindo sites com CSS e (X)HTML – Sites controlados por folhas de estilo em cascata**. São Paulo. Novatec Editora, 1ª Edição, 2008.
- SILVA, Maurício Samy. **CSS3 – Desenvolva aplicações web profissionais com o uso dos poderosos recursos de estilização das CSS3**. São Paulo : Novatec Editora, 1ª Edição, 2010.
- RODRIGUES, Andréa. **Desenvolvimento para Internet**. 1ª Edição, Editora do Livro Técnico, 2010.