

AULA 08 – Layout CSS

PARTE 1: Revisão Teórica

Requisitos Não Funcionais (RNF)

Além desses princípios que norteiam a organização de um layout, precisa ser observado os requisitos não funcionais (RNF) que potencializam a experiência positiva com o site, especialmente a usabilidade e a navegabilidade.

Também, não podemos nos distanciar da legibilidade, pois de nada adianta ser bonito se não pode ser compreendido.

E, por fim, destacamos a relevância da acessibilidade, permitindo um acesso universal e democrático às informações.

RNF	Descrição
Usabilidade	<ul style="list-style-type: none">✓ é um atributo de qualidade relacionado à facilidade de uso de algo.✓ Refere-se à rapidez com que os usuários podem apreender e utilizar algo e o quanto lhes agradam utilizá-la.✓ Se um recurso não pode ser utilizado ou não é utilizado por seus usuários, então, ele não precisa existir.
Navegabilidade	<ul style="list-style-type: none">✓ é uma propriedade da organização das informações que nos remete à capacidade de navegação, ou seja, nos permite percorrer intuitivamente os caminhos virtuais que nos são propostos de forma que saibamos onde estamos, de onde viemos e quais são nossas possibilidades futuras.
Legibilidade	<ul style="list-style-type: none">✓ é uma qualidade que determina a facilidade de leitura de alguma coisa;✓ mede o quão legível está se apresenta ao leitor.
Acessibilidade	<ul style="list-style-type: none">✓ no contexto da internet, é a propriedade que um site tem para permitir que portadores de necessidades especiais se utilizem dos recursos que oferece.✓ Dependendo como os elementos gráficos são utilizados no site, eles inviabilizam qualquer tentativa de um deficiente visual,✓ por exemplo: conseguir acessar as informações disponibilizadas pelo site

Anatomia de um site

Em função dos princípios discutidos até então, estudiosos da área de web design têm experimentado diferentes formas de organização de layouts.

Certamente, enquanto usuário da internet, você já deve ter tido boas e más experiências. Contudo, observamos que há uma espécie de padrão ou de roteiro com bastante aceitação e altamente difundido entre grandes e pequenos sites.

Tal modelo é composto por alguns elementos de conteúdo que podem ser visualizados na Figura 1, na forma de um esquema numerado.

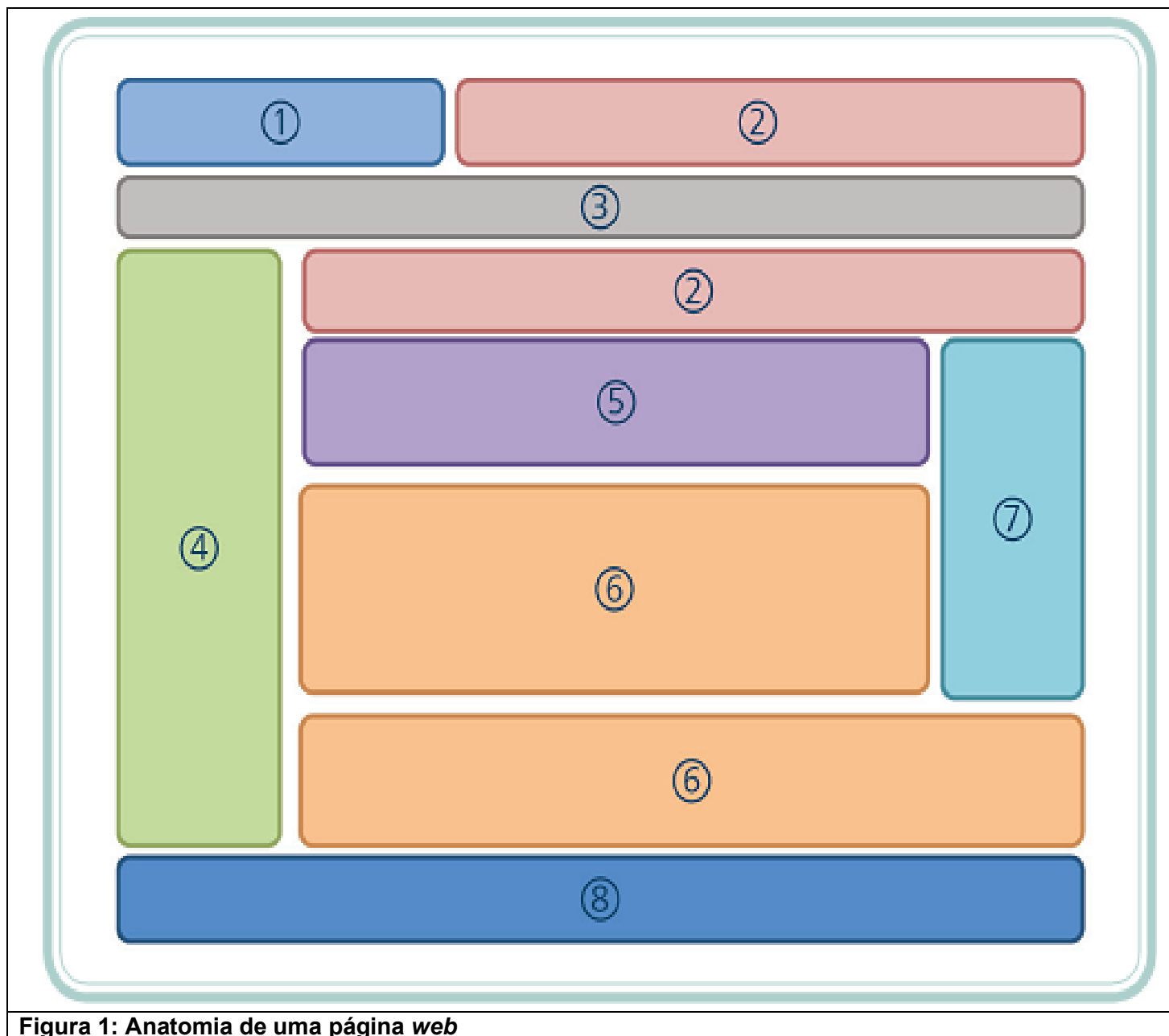


Figura 1: Anatomia de uma página web

Área	Descrição
1) Logotipo	<ul style="list-style-type: none"> ✓ o logotipo ou logomarca do site, normalmente, ocupa o espaço mais nobre, onde, comumente, a visualização ocorre primeiro (canto superior esquerdo).
2) Banner padrão	<ul style="list-style-type: none"> ✓ área de busca, anúncios externos – este é, também, um espaço nobre que, frequentemente, é explorado com um banner. ✓ Nessa área, recomenda-se a utilização de uma ferramenta de busca ou de um mapa do site (para simplificar e agilizar a localização de informações).
3) Menu administrativo	<ul style="list-style-type: none"> ✓ o menu administrativo é uma opção importante, no entanto deve ser discreto. ✓ É nele que colocamos informações sobre a empresa, identificação (login) ou cadastro de usuários, contato, ajuda, etc.
4) Menu de navegação	<ul style="list-style-type: none"> ✓ categorização das informações disponibilizadas pelo site. ✓ É comumente encontrado de forma horizontal, como o menu administrativo. ✓ É importante que as categorias ou opções do menu sejam curtas e claras, indicando ao usuário, rapidamente, a opção que ele deve escolher (excessivas opções também não são recomendadas).

5) Área de destaque	<ul style="list-style-type: none"> ✓ nesse espaço, recomenda-se focar algo de maior importância dentro do contexto do site. ✓ Um site de comércio eletrônico poderia divulgar uma promoção; um site de notícias poderia noticiar o fato mais marcante do dia; um site pessoal poderia remeter para a atividade mais recente ou última postagem.
6) Conteúdo	<ul style="list-style-type: none"> ✓ área de conteúdo é o lugar onde são exibidas as informações na medida em que navegamos pelo site
7) Anúncios	<ul style="list-style-type: none"> ✓ a barra lateral, da direita, pode ser utilizada para divulgação de atividades afins ao site, como anúncios ou links para outras páginas.
8) Rodapé	<ul style="list-style-type: none"> ✓ o rodapé é, normalmente, utilizado para informações sobre o portal. ✓ Não é, necessariamente, usada por visitantes comuns, mas sim por aqueles com algum interesse específico (como anunciar, trabalhe conosco, política de privacidade, termos de uso, etc.).

A Figura 2 sobrepõe a imagem de um portal (www.brasil.gov.br), utilizando o esquema numerado da Figura 1 de forma que, através de um exemplo real, possamos visualizar os elementos de uma página web e sua Anatomia.



Figura 2: Elementos de uma página web, do portal Brasil (www.brasil.gov.br)

Estrutura com html5

Antes do HTML5, DIVs eram usadas para estruturar um site. Agora, ele pode voltar a ser usado como um elemento de estilização, pois é possível substituí-lo pelos novos elementos que o HTML5 dispõe.

O HTML5 inclui vários elementos estruturais que podem ajudar a definir as partes de um documento.

Aqui estão os elementos suportados pelos navegadores mais populares.

Principais elementos estruturais do HTML5:

A seguir são listados alguns elementos servem mais para agrupar elementos do que para posicionar eles.

Ou seja, você pode ter várias **<section>** em uma página, cada uma tendo a sua própria **<header>** e **<footer>**.

tag	Descrição
<header>	<ul style="list-style-type: none">✓ Cabeçalho.✓ Aqui se pode colocar informações importantes como o nome ou logo do site.
<nav>	<ul style="list-style-type: none">✓ Navegação, ou seja,✓ o menu de representação de um bloco principal de links de navegação.✓ Mas cuidado, nem todo grupo de links deve ser tratado como um nav;
<section>	<ul style="list-style-type: none">✓ Seção.✓ utilizado para representar uma seção genérica, geralmente com um cabeçalho próprio e o seu conteúdo;✓ Aqui se pode ter uma seção para o artigo em destaque, outra para artigos de uma determinada categoria do site, etc.
<article>	<ul style="list-style-type: none">✓ Artigo.✓ identifica o conteúdo em si, como uma notícia de um portal, um post em blog ou um comentário em uma lista de comentários.
<aside>	<ul style="list-style-type: none">✓ Lateral.✓ Representa o conteúdo relacionado da página.✓ pode ser utilizada para representar uma seção de conteúdo secundário ou✓ auxiliar a outro pedaço de maior importância.✓ Por exemplo: citações, links de referência ou notas adicionais
<header>	<ul style="list-style-type: none">✓ referente ao cabeçalho de uma seção específica (ou da própria página), contendo títulos, introduções e outros elementos similares
<footer>	<ul style="list-style-type: none">✓ Rodapé,✓ Pode conter dados sobre o autor, copyright, ou até mesmo uma navegação de rodapé.

Cada elemento também pode ter uma **ID** ou **classe atribuída** a ele, como você faria tradicionalmente com as **DIVs**.

Vale notar que a mesma regra **de ID** se aplica no HTML5: só pode existir um **ID único** para **cada elemento**. Se quiser estilizar vários elementos com um mesmo estilo, utilize a **classe**.

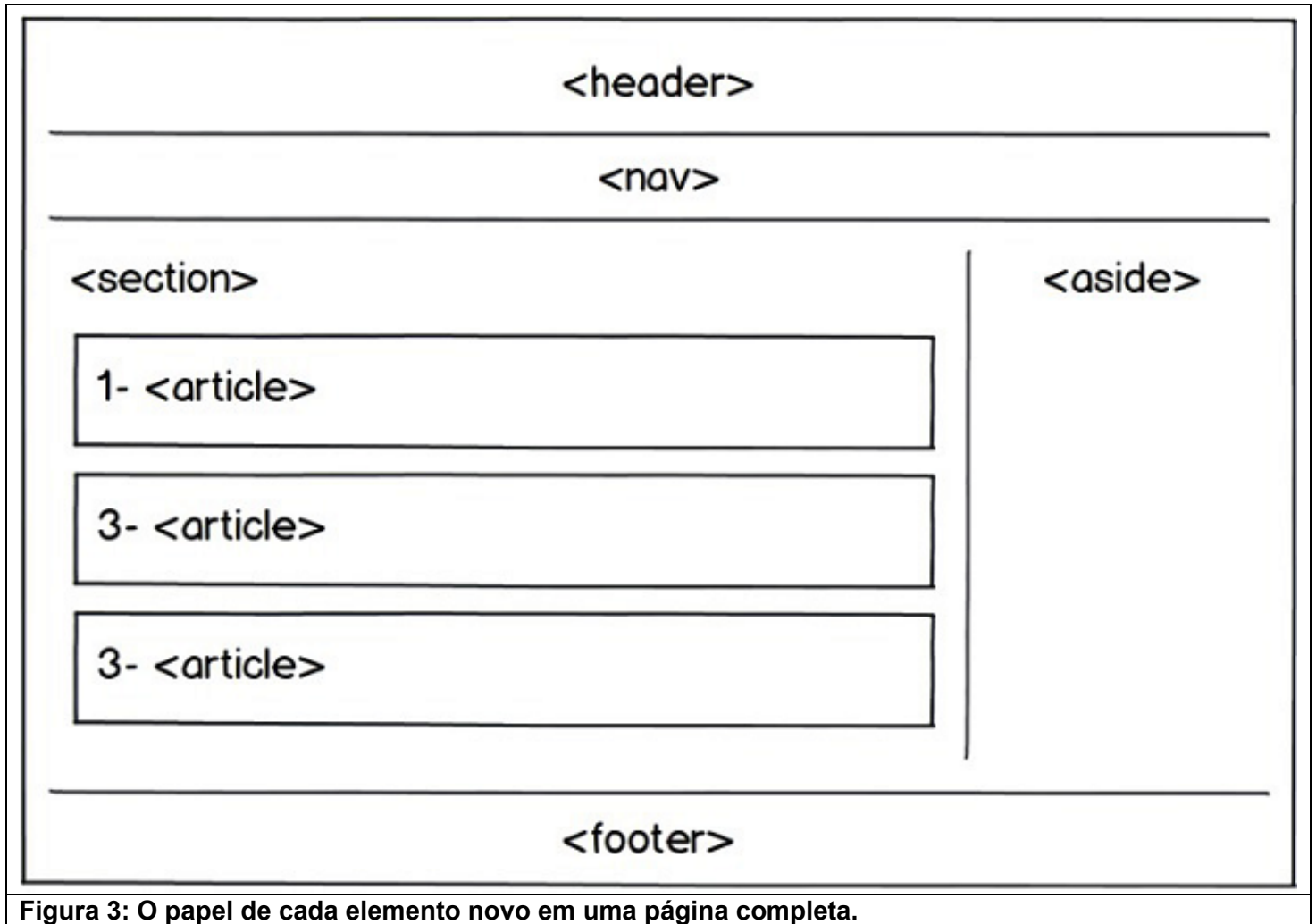


Figura 3: O papel de cada elemento novo em uma página completa.

Observações:

Qualquer elemento pode ter seu conteúdo separado por seções com o elemento **section**. Notar também que os elementos podem ter também um **header** e um **footer** independentes do resto do layout.

Comparando Xhtml com Html5, respectivamente

```
1. <body>
2. <div id="header">...</div>
3. <div id="menu">...</div>
4. <div class="post">...</div>
5. <div id="sidebar">...</div>
6. <div id="rodape">...</div>
7. </body>
```

Listagem 1: CF com Xhtml

```
1. <body>
2. <header>...</header>
3. <nav>...</nav>
4. <section>
5. <article>
6. ...
7. </article>
8. </section>
9. <aside>...</aside>
10. <footer>...</footer>
11. </body>
```

Listagem 2: CF com Html5

Um ponto crucial destas mudanças é que podemos definir diversas seções independentes entre si, cada uma com a sua hierarquia própria.

Por exemplo, podemos ter 2 elementos h1, cada um em sua seção, ou podemos criar um header para o cabeçalho de um blog, com o título e links de navegação, e cada post, devidamente criado em um article, ter um header com o título do post e a data em que ele foi publicado, por exemplo. Isso pode ser bastante útil para

se arquitetar páginas mais modulares, para melhorar a qualidade e a facilidade de manutenção do nosso código.

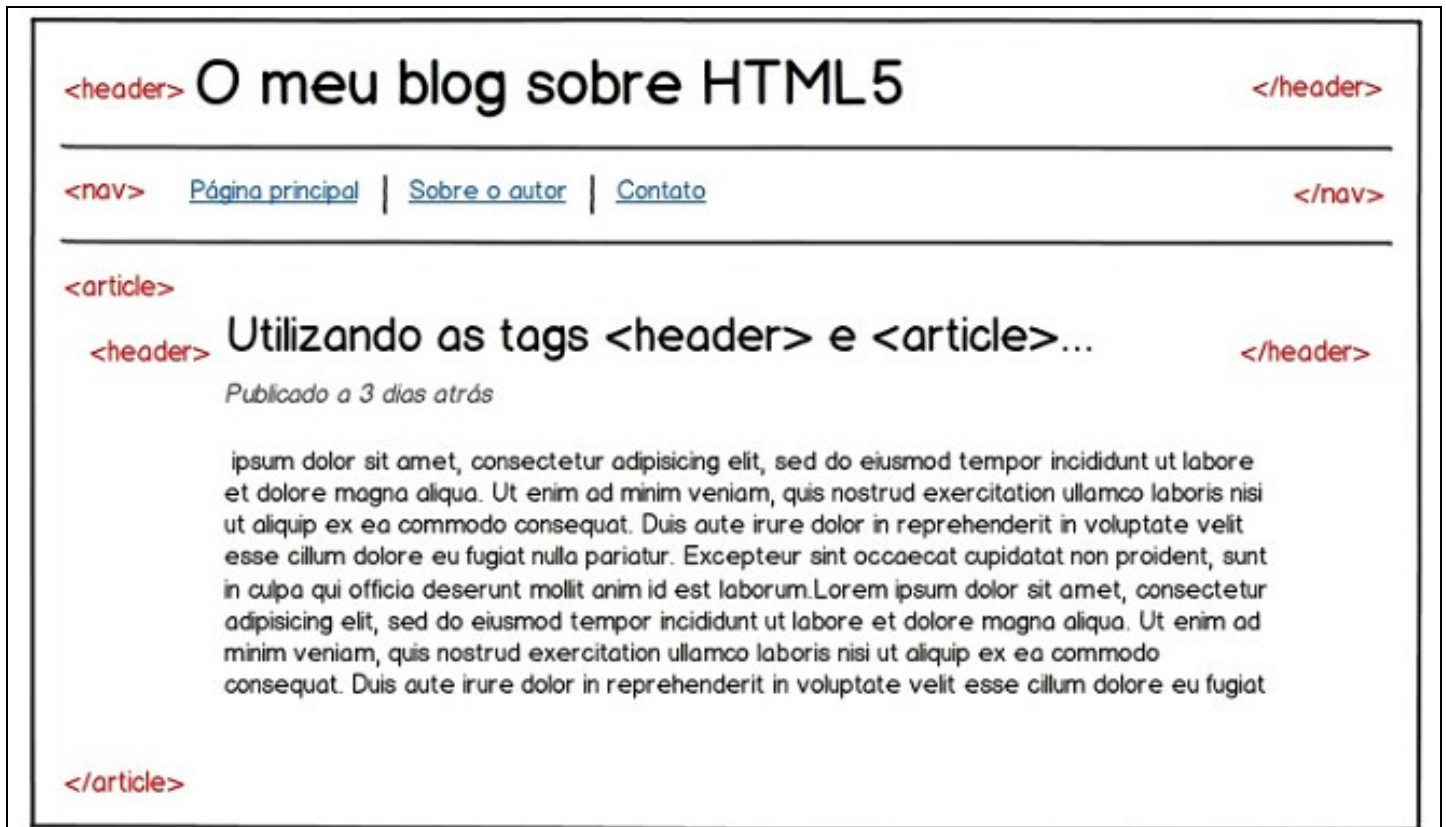


Figura 4: Um mockup do exemplo descrito - a tag '<header>' pode aparecer em 2 seções diferentes.

Exemplo CSS-6.1

```
body
{
  text-align: center; /* centraliza TUDO inclusive o container*/
}

#container
{
  width: 960px;
  margin: 0 auto;
  text-align: left; /*volta para a esquerda*/
}

#header
{
  background-color: Gray;
  height: 119px;
  text-align: center;
  font-size: 30px;
  color: darkgreen;
}

#mainnav
{
  background-color: #ffd800;
```

```
color: #272900;
padding: 2px 0;

}

#mainnav li
{
display: inline;
padding: 0 10px;
border-right: 1px solid #006;
}

#mainnav ul
{
margin: 0;
padding: 0;
}

#mainnav li a
{
text-decoration: none;
color: #272900;
}

#mainnav li a: hover
{
text-decoration: none;
color: #fff;
background-color: #272900;
}

#menuLateral
{
background-color: lightslategrey;
color: #e6e3e3;
height: 366px; /* 550 altura dividido por 3 */
width: 320px;
float: left;
padding-top: 20px;
}

#conteudo
{
background-color: #e6e3e3;
height: 366px; /* 550 altura dividido por 3 */
width: 640px; /* 960/3 = 320 * 2 = 640 */
/*padding-left: 20px; /* descontar do total width */
padding-top: 20px;
float: left;
}

#footer
{
background-color: gray;
height: 61px;
clear: both;
text-align: center;
}
```

Listagem 3: estilo81.css


```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, e CSS3" />
    <meta name="Keywords" content="CSS3, Div, Classes e Span" />
    <title>Aula 08 - Exemplo 81 (Curso de HTML 5 e CSS 3 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estilo81.css">
  </head>
<body>

<div id="container">

<div id="header">
<h1 style="margin-top:20px;">Primeiro exemplo de Layout</h1>
</div>

  <div id="mainnav">
    <ul>
      <li><a href="home.html">Home</a></li>
      <li><a href="quemsomos.html">Quem somos</a></li>
      <li><a href="mapa.html">Mapa do Site</a></li>
      <li><a href="localizacao.html">Localização</a></li>
      <li><a href="contato.html">Contato</a></li>
    </ul>
  </div>

  <div id="menuLateral">
<b>Navegação Lateral</b><br>
Opção1<br>
Opção2<br>
Opção3
</div>

  <div id="conteudo">
    Conteúdo Principal
  </div>

  <div id="footer">
    Copyright © layout com divs - Todos os direitos reservados
  </div>
</div>

</body>
</html>
```

Listagem 4: Exemplo81.html



Seja pragmático

Enquanto este mundo de novas possibilidades e tags a nossa disposição pode parecer a solução, é muito fácil se perder no mérito das discussões e definições do tipo de "isto aqui deveria ser um **article** ou um **section**?".

Continuar utilizar a **tag div** não causa mal nenhum, e você sempre pode voltar no seu código depois e trocar alguma tag por outra que você considere mais adequada.

Posicionando elementos

Com posicionamento CSS podemos colocar um elemento em uma posição exata na página. Combinado com **floats**, o posicionamento abre muitas possibilidades para criação de layouts precisos e avançados.

O princípio de posicionamento CSS

Considere a janela do navegador como um sistema de coordenadas:

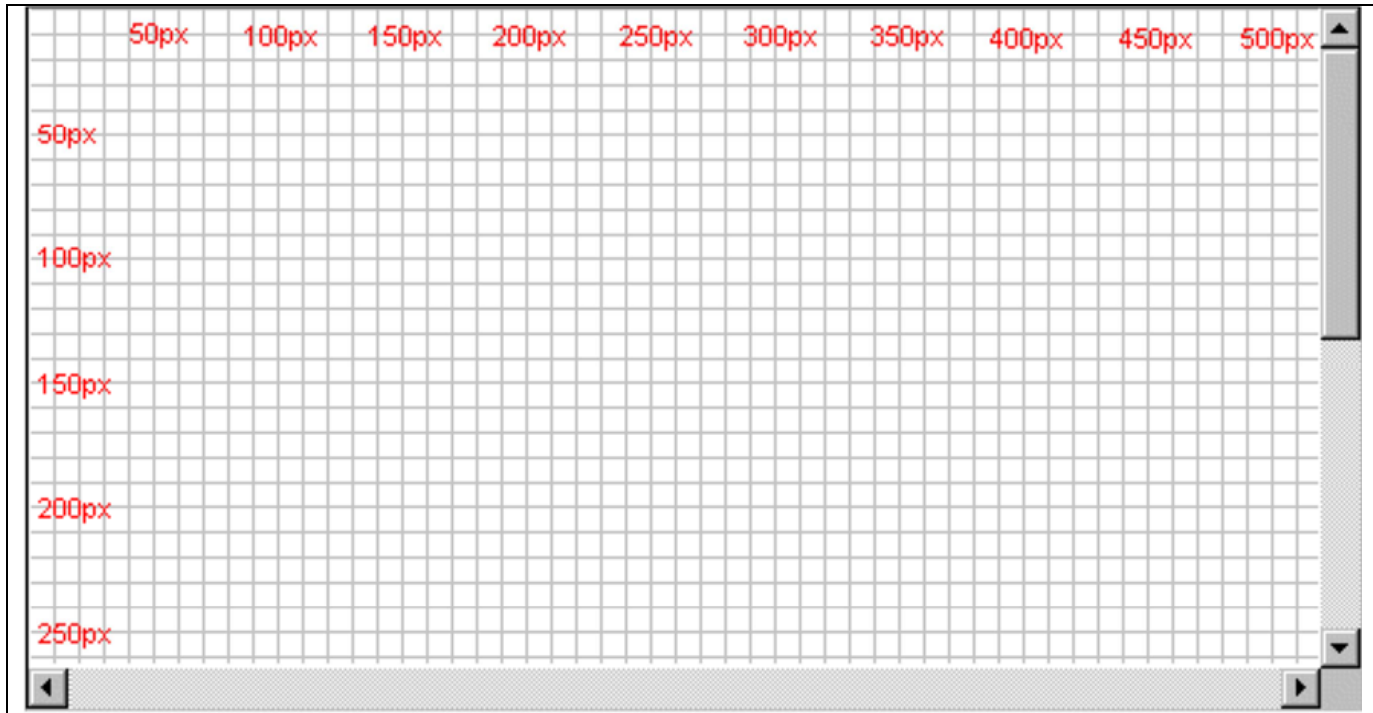


Figura 5:

O princípio de posicionamento CSS estabelece que é possível posicionar um elemento em qualquer lugar na tela usando um sistema de coordenadas. Por exemplo, vamos supor que queremos posicionar um cabeçalho.

Usando o **box model** o cabeçalho pode ser estilizado para ser apresentado como mostrado abaixo:

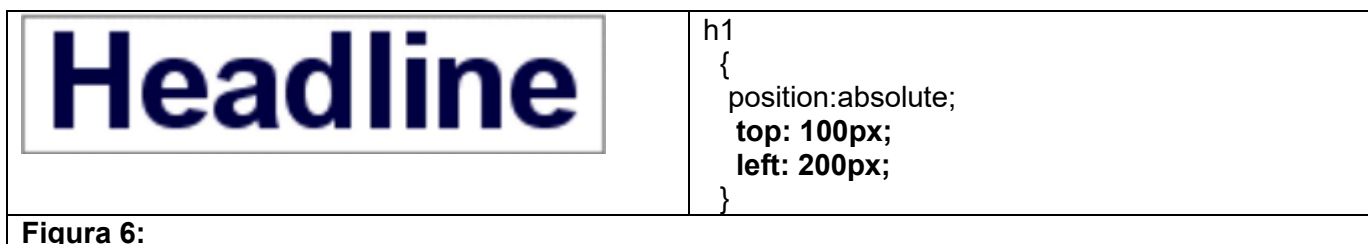


Figura 6:

O resultado é mostrado a seguir:



Figura 7:

Posicionamento absoluto

Um elemento posicionado absolutamente não cria nenhum espaço no documento. Isto significa que não deixa nenhum espaço vazio após ser posicionado. Para posicionar um elemento de forma absoluta a propriedade `position` deve ser definida para `absolute`.

Você pode então usar as propriedades **left**, **right**, **top**, e **bottom** para definir as coordenadas e posicionar o elemento.

Para exemplificar o posicionamento absoluto escolhemos colocar quatro boxes nos quatro cantos da página:

```
#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;
}
```

Listagem 5

Posicionamento relativo

Para posicionar um elemento de forma relativa a propriedade `position` deve ser definida para `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.

Assim fazendo o elemento ocupa um espaço após ser posicionado. Como exemplo de posicionamento relativo vamos tentar posicionar três imagens relativamente as suas posições originais na página. Notar como as imagens deixam um espaço vazio nas suas posições originais no documento:

```
#bola1
{
  position: relative;
  left: 350px;
  bottom: 150px;
}
```

```
#bola2
{
position: relative;
left: 150px;
bottom: 500px;
}

#bola3
{
position: relative;
left: 50px;
bottom: 700px;
}
```

Listagem 6:

Tutorial (passo a passo) de como criar um Layout.

Este exemplo é uma adaptação do <http://www.maujor.com/tutorial/layout-css-passo-a-passo.php>

1o. passo - Decidir a qual nível de navegadores o layout será servido

Antes de começar seu layout você deverá decidir a quais navegadores ele deverá ser servido. Verificar quais os usuários do **site (público-alvo)** e se possível examinar os relatórios e logs indicativos de navegadores mais usados.

Existem 3 níveis de suporte às CSS pelos navegadores, que devem ser considerados:

Navegadores modernos		Navegadores recentes	Navegadores antigos
Opção 1.	Suporte total às CSS	Suporte total às CSS	Suporte total às CSS
Opção 2.	Suporte total às CSS	Suporte total às CSS	Suporte parcial às CSS
Opção 3.	Suporte total às CSS	Suporte total às CSS	Conteúdos não estilizados
Opção 4.	Suporte total às CSS	Suporte parcial às CSS	Conteúdos não estilizados

Tabela 1: Opções para suporte às CSS

2o. passo - Escolher os containers do layout

Para posicionar os elementos na página é necessário criar os containers principais. Analise o design da página (que pode ser uma figura, um esboço em folha de papel ou um template existente) e definir os containers principais.

Imagem do layout com containers definidos.

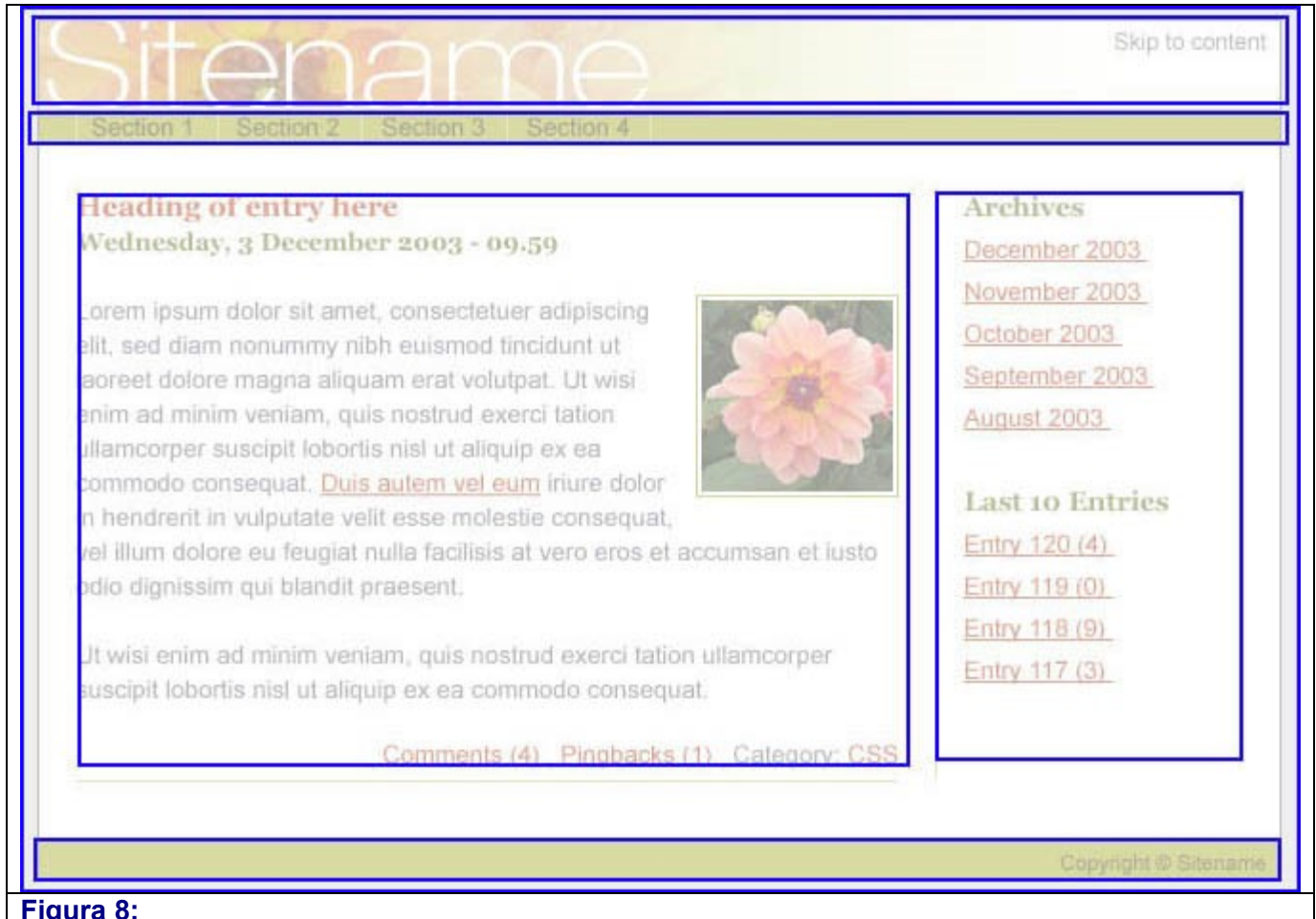


Figura 8:

3o. passo - Nomear os containers

Estes containers serão os elementos **DIV** principais nas quais serão inseridos os conteúdos e eles deverão ser designados com nomes ligados semanticamente às suas finalidades evitando-se nomeá-las de acordo com sua apresentação ou aparência.

No nosso exemplo adotaremos os seguintes nomes para as **DIVs** principais do layout:

- a) **container**
- b) **header**
- c) **mainnav**
- d) **menu**
- e) **contents**
- f) **footer**

Se o container for único na página use IDs no lugar de classes. Isto é importante para a aplicação de estilos na página. Caso ocorram regras CSS conflitantes, os seletores IDs terão prioridade sobre os seletores classes.

A figura 9 mostra a imagem do layout com containers nomeados

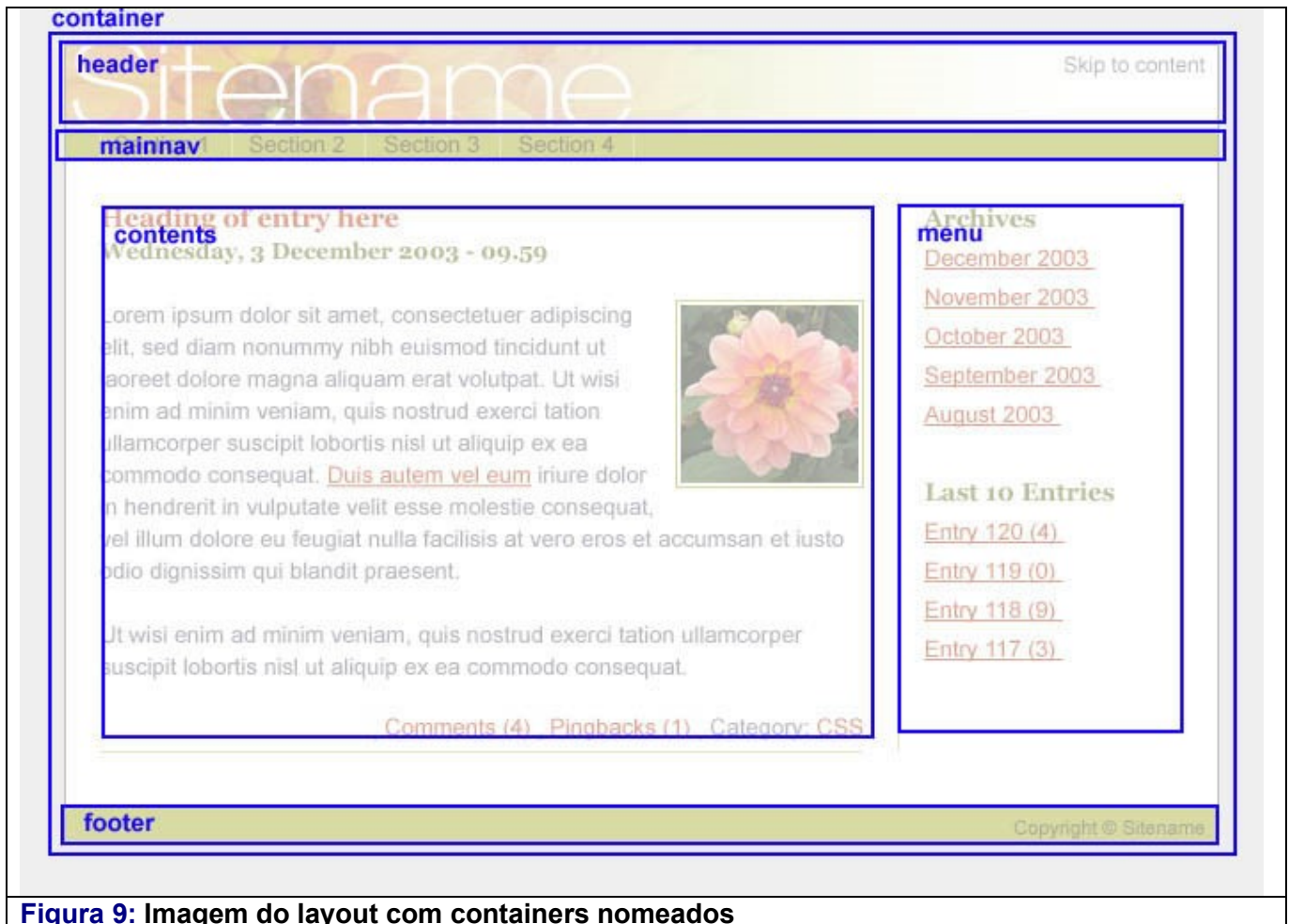


Figura 9: Imagem do layout com containers nomeados

4º. passo - Criar a marcação

Comece a declaração de DOCTYPE — no nosso caso vamos usar HTML 4.10 strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"> <html lang="en">
```

Prossiga com a seção **head** do documento, incluindo a codificação de caracteres, link para folhas de estilo que no nosso exemplo foi chamada de “estrutura.css”:

```
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <title>Page title</title>
  <link rel="stylesheet" href="estrutura.css" type="text/css" media="screen">
</head>
```

Listagem 7

Finalmente inclua as informações para a seção **body** — crie DIVs vazias para os containers do layout. Listagem 8, mostra o CF.

```
<body>
<div id="container">
  <div id="header"></div>
  <div id="mainnav"></div>
```

```

        <div id="menu"></div>
        <div id="contents"></div>
        <div id="footer"></div>
</div>
</body>
</html>

```

Listagem 8

5o. passo - Inserir conteúdo

```

<body>
<div id="container">
    <div id="header" title="sitename">
        <div id="skipmenu"><a href="#content">Skip to content</a></div>
        <h1>Sitename</h1>
    </div>
<div id="mainnav">
    <ul>
        <li><a href="#">Section 1</a></li>
        <li><a href="#">Section 2</a></li>
        <li><a href="#">Section 3</a></li>
        <li><a href="#">Section 4</a></li>
    </ul>
</div>
<div id="menu">
<h3>Archives</h3>
    <ul>
        <li><a href="#">December 2003</a></li>
        <li><a href="#">November 2003</a></li>
        <li><a href="#">October 2003</a></li>
        <li><a href="#">September 2003</a></li>
        <li><a href="#">August 2003</a></li>
    </ul>
<h3>Last 10 Entries</h3>
    <ul>
        <li><a href="#">Entry 120 (4)</a></li>
        <li><a href="#">Entry 119 (0)</a></li>
        <li><a href="#">Entry 118 (9)</a></li>
        <li><a href="#">Entry 117 (3)</a></li>
    </ul>
</div>
<div id="contents">
<div class="blogentry">
    <h2>
    <a href="#" title="Permanent link to this item">Heading here</a>
    </h2>
    <h3>Wednesday, 3 December 2003</h3>
    <p>

    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod
    tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci
    tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. <a href="#">Duis autem vel
    eum</a> iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat
    nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent.

    </p>
    <p>
    Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip
    ex ea commodo consequat.

```



```
</p>
<ul>
<li><a href="#">Comments (4)</a></li>
<li><a href="#">Pingbacks (1)</a></li>
<li>Category: <a href="#" title="Category">CSS</a></li>
</ul>
</div>
</div>
<div id="footer">
Copyright © Sitenome 2004
</div>
</div>
</body>
</html>
```

Listagem 9:

6o. passo - Posicionar os containers

Os containers do layout devem ser posicionados com uso dos métodos CSS de posicionamento. Para isso vamos definir regras de estilo no arquivo estrutura.css. Depois que os containers forem posicionados voltaremos ao arquivo da folha de estilo para acrescentar mais regras CSS.

```
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;
}
```

Listagem 10: estrutura.css

[Skip to content](#)

Sitename

- [Section 1](#)
- [Section 2](#)
- [Section 3](#)
- [Section 4](#)

Heading here

Wednesday, 3 December 2003



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. [Duis autem vel eum](#) iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

- [Comments \(4\)](#)
- [Pingbacks \(1\)](#)
- Category: [CSS](#)

Archives

- [December 2003](#)
- [November 2003](#)
- [October 2003](#)
- [September 2003](#)
- [August 2003](#)

Last 10 Entries

- [Entry 120 \(4\)](#)
- [Entry 119 \(0\)](#)
- [Entry 118 \(9\)](#)
- [Entry 117 \(3\)](#)

Copyright © Sitename 2004

Figura 10:**7o. passo - Algum problema?**

O layout construído funciona bem em todos os navegadores, mas apresenta problema no Internet Explorer 6. Observe como ele é renderizado naquele navegador.

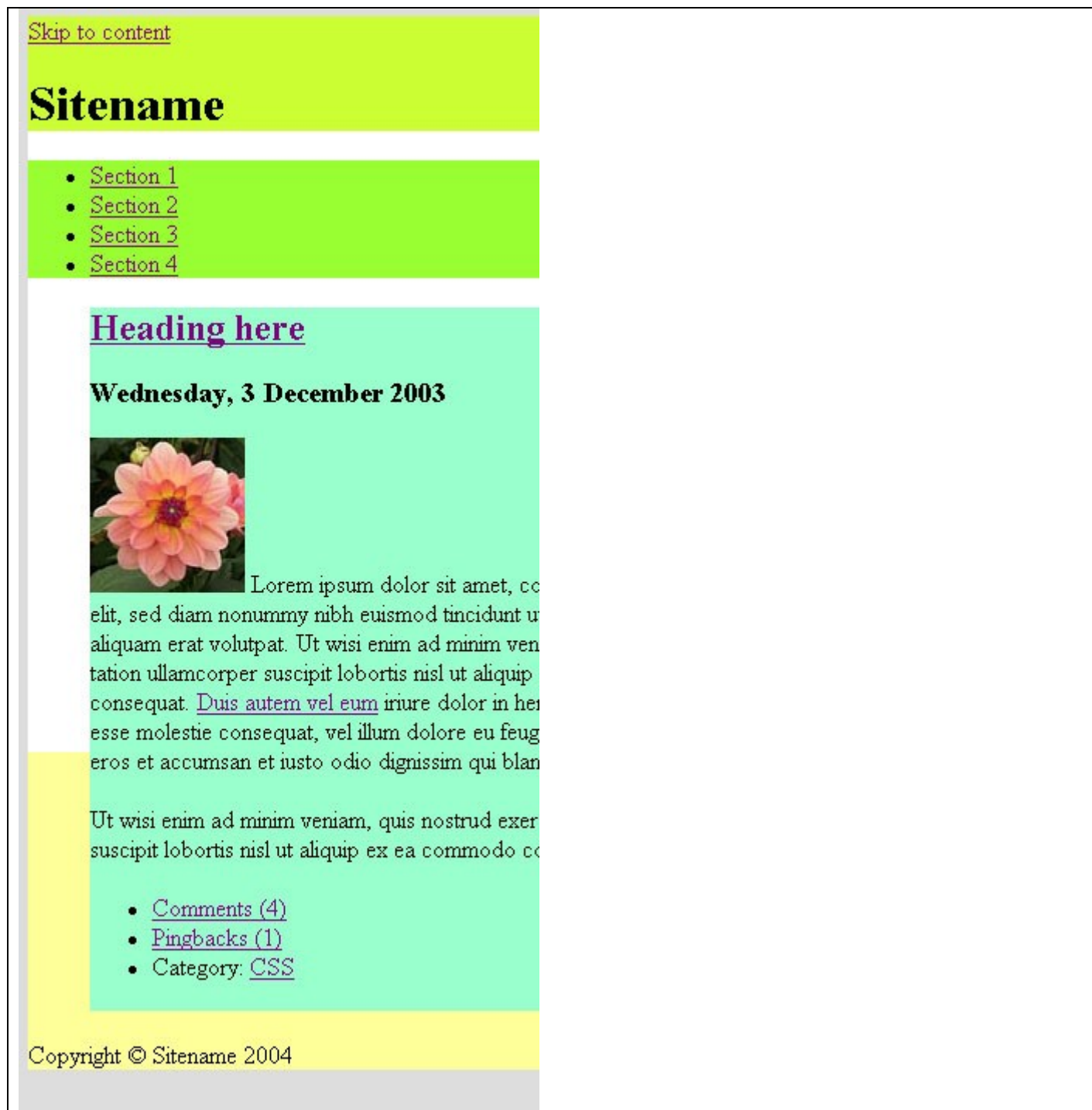


Figura 11:

A margem esquerda do container contents é muito grande. Isso acontece porque o IE6 apresenta o bug da margem dobrada para elementos flutuados. Esse bug pode ser resolvido acrescentando-se a declaração `display: inline` para o container flutuado, no caso `#contents`.

```
#contents {
  float: left;
  width: 440px;
  background: #9FC;
  margin: 0 0 0 20px;
  display: inline;
}
```

Listagem 11:

Como essa declaração é somente para o IE6 é melhor movê-la para um arquivo de folha de estilo separada a ser servida somente para o IE6. Essa folha de estilo será servida com uso de comentário condicional específico para o IE6, como mostrado a seguir.

```
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <title>Page title</title>
  <link rel="stylesheet" href="estrutura.css" type="text/css" media="screen">
  <!--[if IE 6]>
  <link rel="stylesheet" href="ie6.css" type="text/css" media="screen">
  <![endif]-->
</head>
```

Listagem 12:

8o. passo - Completar a folha de estilos

Uma vez posicionado os containers e sendo o layout consistente para todos os navegadores podemos completar nossa folha de estilos, como mostrado na listagem 13.

```
/* global styles */

body {
  margin: 0;
  padding: 0;
  font: 85% arial, hevetica, sans-serif;
  text-align: center;
  color: #333;
  background: #ddd url(img_39.gif) repeat 0 0;
}

a:link { color: #B52C07; }
a:visited { color: #b93411; }
a:focus { color: #000; }
a:hover { color: #7d8206; }
a:active { color: red; }
h1, h2, h3, h4, h5, h6 { margin: 0 0 .5em; }

h2 {
  color: #B52C07;
  font: 140% georgia, times, "times new roman", serif;
}

h2 a { text-decoration: none; }

h3 {
  color: #7d8206;
  font: 110% georgia, times, "times new roman", serif;
}

/* container styles */

#container {
  margin: 1em auto;
  width: 650px;
  text-align: left;
  background: #fff;
```

```
border: 1px solid #676767;
}

/* header styles */

#header {
  height: 45px;
  width: 100%;
  position: relative;
  background: url(header.jpg) no-repeat 0 0;
  border-bottom: 1px solid #fff;
}

#header h1 {
  position: absolute;
  left: -500em;
}

#skipmenu {
  position: absolute;
  left: 0;
  top: 5px;
  width: 645px;
  text-align: right;
}

#skipmenu a {
  color: #555;
  text-decoration: none;
}

/* mainnav styles */

#mainnav {
  background: #9FA41D;
  color: #272900;
  padding: 2px 0;
  margin-bottom: 22px;
}

#mainnav ul {
  margin: 0 0 0 20px;
  padding: 0;
  list-style-type: none;
  border-left: 1px solid #C4C769;
}

#mainnav li {
  display: inline;
  padding: 0 10px;
  border-right: 1px solid #C4C769;
}

#mainnav li a {
  text-decoration: none;
  color: #272900;
}

#mainnav li a:hover {
  text-decoration: none;
```

```
        color: #fff;
        background-color: #272900;
    }

/* menu styles */

#menu {
    float: right;
    width: 165px;
    border-left: 1px solid #C5C877;
    padding-left: 15px;
}

#menu ul {
    margin: 1em 0;
    padding: 0;
}

#menu ul li {
    margin: 0 0 1em;
    padding: 0;
    list-style-type: none;
}

/* contents styles */

#contents {
    float: left;
    width: 430px;
    margin: 0 0 0 20px;
}

#contents p { line-height: 165%; }
.blogentry { border-bottom: 1px solid #C5C877; }

.blogentry ul {
    text-align: right;
    margin: 1em 0;
    padding: 0;
    font-size: 95%;
}

.blogentry li {
    list-style-type: none;
    display: inline;
    margin: 0;
    padding: 0 0 0 7px;
}

.imagefloat {
    float: right;
    padding: 2px;
    border: 1px solid #9FA41D;
    margin: 0 0 10px 10px;
}

/* footer styles */

#footer {
    clear: both;
```

```
color: #272900;
text-align: right;
font-size: 90%;
background: #9FA41D;
padding: 5px;
}
```

Listagem 13: estrutura.css

9o. passo - Folha de estilo para impressão

Para terminar vamos preparar o layout para impressão. Se você pretende que o layout seja impresso exatamente como aparece na tela basta definir os valores all ou print para o atributo media do elemento que linka para a folha de estilo conforme mostrado a seguir.

```
<link rel="stylesheet" href="style.css" type="text/css" media="screen print">
```

Ou

```
<link rel="stylesheet" href="style.css" type="text/css" media="all">
```

Para criar uma folha de estilos destinada a impressão personalizada (sem os elementos da tela que não interessam, como por exemplo, logotipos, rodapés, figuras decorativas, etc) siga os seguintes passos. Crie uma cópia da folha de estilo principal (estrutura.css) e grave a cópia com o nome de backup.css.

Vamos trabalhar na folhastyle.css que estando linkada ao documento irá refletir no layout as alterações feitas, permitindo que se visualize imediatamente a alteração sem necessidade de dar um "visualizar impressão" cada vez que se faz uma alteração.

Quando terminarmos com as alterações basta renomear o arquivo para print.css, renomear o arquivo backup.css parastyle.css e linká-los ao documento.

Olhe para o layout e decida quais são os conteúdos que serão impressos.

No nosso exemplo nós descartaremos o gráfico do topo (header), a navegação principal (**mainnav**), o menu da direita bem como as cores e sublinhados dos links.

- remova todos os containers não necessários com uso da declaração display: none;
- Mude todas as cores para preto ou tons de cinza.
- Retire sublinhados de links com a declaração a { text-decoration: none;}
- Remova os tamanhos de fonte e deixe tudo com o tamanho padrão 100% (default).

```
body {
    margin: 0;
    padding: 0;
    font: 100% arial, hevetica, sans-serif;
    color: #000;
    background-color: #fff;
}

a {
    color: #000;
    text-decoration: none;
}

#header { border-bottom: 1px solid #999; }
#header h1 { color: #000; }
#mainnav, #menu, .imagefloat, #skipmenu { display: none; }
```



```
#menu ul {
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    line-height: 165%;
}

#contents p { line-height: 165%; }
.blogentry { border-bottom: 1px solid #999; }

.blogentry ul {
    list-style-type: none;
    text-align: right;
    margin: 1em 0;
    padding: 0;
    font-size: 95%;
}

.blogentry li {
    display: inline;
    padding: 0 0 0 7px;
}

#footer {
    clear: both;
    color: #000;
    text-align: right;
    padding: 5px;
    font-size: 90%;
    border-top: 1px solid #999;
    margin-top: 2em;
}
```

Listagem 14:

Sitename

Heading here

Wednesday, 3 December 2003

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Comments (4) Pingbacks (1) Category: CSS

Copyright © Sitename 2004

Figura 12

Acrescente mais um link para folha de estilo na seção head do documento como mostrado a seguir.

```
<link rel="stylesheet" href="print.css" media="print">
```

Fim do layout passo a passo

Terminamos com um documento HTML que:

- renderiza corretamente em navegadores modernos;
- renderiza razoavelmente em navegadores recentes (pequeno bug no IE5 para Windows);
- renderiza de maneira aceitável em navegadores antigos (a estilização é parcial e o conteúdo é acessível);
- renderiza diferenciadamente para impressão.

Resumo

É importante destacar que a web é um espaço muito democrático sempre aberto a inovações, no entanto existem alguns princípios e técnicas largamente discutidos e que, quando seguidos, oferecem ao usuário uma melhor experiência. Nesta aula, discutimos alguns princípios e técnicas que nos orientam a organizar conteúdos para publicação em ambientes on-line, bem como noções de como organizá-los em uma proposta de layout.

FORMULÁRIOS HTML

CRIANDO FORMULÁRIO COM HTML

Um formulário HTML obriga a algumas tags e regras que não podem falhar. Um formulário deve ser iniciado sempre pela tag form que tem de ser fechada no final. Dentro da tag form, deve ser declarada também a action (o arquivo que vai processar a informação) e o método de envio dos dados

```
<form method="post" action="trata.php">
  <p><label>Email: <input type="email" name="email" placeholder="Introduza o seu email"></label></p>
  <p><input type="submit" value="Enviar"><input type="reset"></p>
</form>
```

Listagem 1. Estrutura mínima de um formulario

Um dos novos tipos adicionados ao controlo input é o email.

Este permite de imediato transmitir ao browser que o que será inserido neste campo é um endereço de correio electrónico.

A grande vantagem disto prende-se com o facto de, por exemplo em dispositivos móveis, o browser apresentar um teclado já adaptado ao conteúdo a usar.

A tag form define a área onde estará inserido o formulário com os campos input.

```
<form>
  inserir inputs
</form>
```

O ELEMENTO INPUT

O elemento input é o elemento mais importante do formulário

O elemento input pode ser exibido de várias formas dependendo do tipo de atributo necessário, como veremos nos exemplos (ver menu Exemplos na barra de navegação).

Cada input contido no formulário deve conter o atributo name, haja vista que os valores são associados com a propriedade name de cada input.

Isto acontece com muitos outros tipos de inputs no que são novos no HTML5, tais como:

Valor	Descrição
button	Devine um botão clicável. Utilizado sobretudo para activar JavaScript
checkbox	Cria uma checkbox
color	Cria um color picker
date	Define um formulário com controlador de data, sem hora
datetime-local	Define um formulário com controlador de data e hora

email	Define um campo para inserir email.
file	Define um campo para escolha de ficheiros. Usado para upload de ficheiros.
hidden	Define um campo oculto
image	Permite usar uma imagem como botão
month	Cria um controlo para mês e ano
number	Cria um campo para inserir apenas números
password	Define um campo para inserir passwords
radio	Cria um radio button
range	Define um slider
reset	Cria um botão para limpar os dados já inseridos no formulário.
search	Define um campo para inserir texto de pesquisa
submit	Cria um botão para enviar os dados inseridos.
tel	Cria um campo para inserir o telemóvel/telefone.
text	Input por defeito. Permite inserir texto e números.
time	Define um campo para inserir hora.
url	Cria um campo para inserir o URL de uma página.
week	Cria um campo para enviar a semana.

Esta tabela foi adaptada diretamente do site www.w3schools.com. Visite-a para ver mais exemplos.

a) **Componentes HTML (Form)**

- ✓ É o componente principal onde se encontram todos os campos do formulário.
- ✓ Sintaxe:

```
<form method="post" action="inserir.php">
...outros campos...
</form>
```

Listagem 3: Componente Form

b) **Componentes HTML (TextField)**

- ✓ Name: chave do \$_GET ou \$_POST
- ✓ type: igual a text (tipo texto)
- ✓ value: valor inicial (opcional)
- ✓ Sintaxe:

Digite o seu nome <input name="_nome" type="text" value=" " />

Listagem 4: Componente TextField

c) Componentes HTML (Password)

- ✓ Name: chave do \$_GET ou \$_POST
- ✓ type: igual a password (tipo senha)
- ✓ Sintaxe:

<input name="_senha" type="password" />

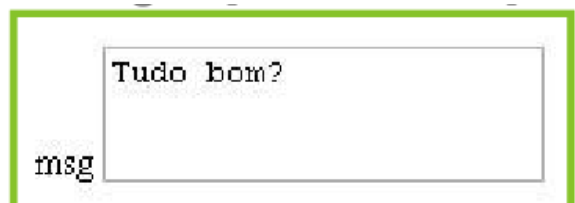


Listagem 5: Componente Password

d) Componentes HTML (TextArea)

- ✓ Sintaxe:

```
<textarea name="_msg" id="msg"> [valor inicial]
</textarea>
```



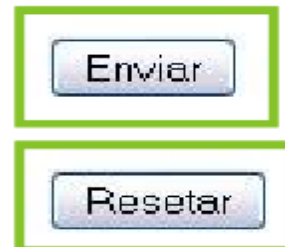
Listagem 6: Componente TextArea

e) Componentes HTML (Button)

- ✓ value: rótulo do botão
- ✓ type: reset ou submit
- ✓ Sintaxe:

```
<button type="submit"> Enviar </button>

<button type="reset"> Resetar </button>
```

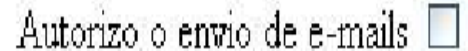


Listagem 6: Componente Button

f) Componentes HTML (CheckBox)

- ✓ Name: chave do \$_GET ou \$_POST
- ✓ value: valor quando for marcado
- ✓ Sintaxe:

```
Autorizo o envio de E-MAIL<input name="_opcao" type="checkbox" value="1" />
```

**Listagem 7: Componente CheckBox****g) Componentes HTML (RadioButton)**

- ✓ Name: chave do \$_GET ou \$_POST (devem possuir o mesmo nome)
- ✓ Value: valor do \$_GET ou \$_POST
- ✓ Sintaxe:

```
Masculino <input type="radio" name="sexo" value="masculino" />
```

```
Femenino <input type="radio" name="sexo" value="fememino" />
```

**Listagem 8: Componente RadioButton****h) Componentes HTML (Select)**

- ✓ Name: chave do \$_GET ou \$_POST
- ✓ value: valor de “cidade” quando a opção for selecionada
- ✓ Sintaxe:

```
<select name="cidade" id="cidade">
<option value="Bertioga">Bertioga</option>
<option value="PraiaGrande">Praia Grande</option>
<option value="Santos" selected="selected"> Santos</option>
<option value="SaoVicente">São Vicente</option>
</select>
```

Listagem 9.1: a6-formularioCADASTRO.html***CRIANDO UM FORMULÁRIO EM HTML***

A seguir temos um formulário básico de contato, neste é pedido 3 informações:

- nome
- e-mail
- mensagem

VALIDAR FORMULÁRIOS

A definição do tipo de input no HTML5 é muito importante porque permite que o browser valide os dados inseridos ao submeter sem a necessidade de código extra, ao contrário do que acontecia com o HTML4

Com o HTML5 e os browsers modernos, a validação com JavaScript no lado do cliente deixa de ser obrigatória.

No entanto ela também deve ser feita do lado do servidor, por razões de segurança.

Uma validação pode ser feita no lado do cliente e no lado do servidor ou apenas no lado do servidor. NUNCA deve ser feita apenas no lado do cliente.

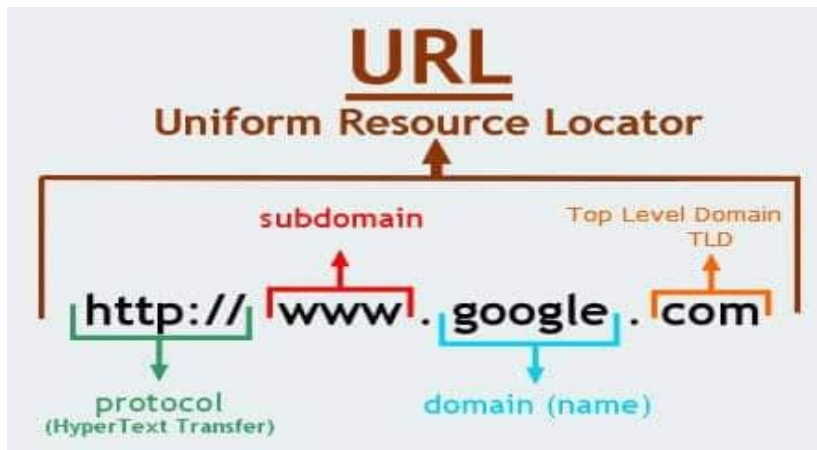
Referência bibliográfica

<http://www.maujor.com/tutorial/layout-css-passo-a-passo.php>

<https://www.devmedia.com.br/criando-form-de-contato-com-html5-css3-e-javascript/29415>

<http://www.sergiomartins.pt/web/index.php/blogser/14-tutoriais/141-formularios-html5>

Apêndice



DOMAIN NAMES

• Com	→	Commercial
• net	→	Network
• edu	→	Education
• org	→	organization
• mil	→	Military
• gov	→	Government
• info	→	Information
• arpa	→	Infrastructure
• int	→	International
• mobi	→	Mobile
• travel	→	Travel
• jobs	→	Jobs
• ca	→	Canada
• coop	→	Cooperative

TYPES OF DOMAIN NAME

.com → Commercial	.infra → Infrastructure
.net → Network	.int → International
.edu → Education	.mobi → Mobile
.org → Organization	.travel → Travel
.mil → Military	.jobs → Jobs
.gov → Government	.ca → Canada
.info → Information	.coop → Cooperative



Document Outline <ul style="list-style-type: none"> <!DOCTYPE> Version of (X)HTML <html> HTML document <head> Page information <body> Page contents 	Lists <ul style="list-style-type: none"> Ordered list Unordered list List item <dl> Definition list <dt> Definition term <dd> Term description 	Objects <ul style="list-style-type: none"> <object> Object <param /> Parameter
Comments <ul style="list-style-type: none"> <!-- Comment Text --> 	Forms <ul style="list-style-type: none"> <form> Form <fieldset> Collection of fields <legend> Form legend <label> Input label <input /> Form input <select> Drop-down box <optgroup> Group of options <option> Drop-down options <textarea> Large text input <button> Button 	Empty Elements <ul style="list-style-type: none"> <area /> <base /> <input />
 <link /> <col /> <meta /> <hr /> <param />
Page Information <ul style="list-style-type: none"> <base /> Base URL <meta /> Meta data <title> Title <link /> Relevant resource <style> Style resource <script> Script resource 	Tables <ul style="list-style-type: none"> <table> Table <caption> Caption <thead> Table header <tbody> Table body <tfoot> Table footer <colgroup> Column group <col /> Column <tr> Table row <th> Header cell <td> Table cell 	Core Attributes <ul style="list-style-type: none"> class style id title <p>Note: Core Attributes may not be used in base, head, html, meta, param, script, style or title elements.</p>
Document Structure <ul style="list-style-type: none"> <h[1-6]> Heading <div> Page section Inline section <p> Paragraph
 Line break <hr /> Horizontal rule 	Language Attributes <ul style="list-style-type: none"> dir lang <p>Note: Language Attributes may not be used in base, br, frame, frameset, hr, iframe, param or script elements.</p>	Keyboard Attributes <ul style="list-style-type: none"> accesskey tabindex
Links <ul style="list-style-type: none"> Page link Email link Anchor Link to anchor 	Images and Image Maps <ul style="list-style-type: none"> Image <map> Image Map <area /> Area of Image Map 	Window Events <ul style="list-style-type: none"> onLoad onUnload
Text Markup <ul style="list-style-type: none"> Strong emphasis Emphasis <blockquote> Long quotation <q> Short quotation <abbr> Abbreviation <acronym> Acronym <address> Address <pre> Pre-formatted text <dfn> Definition <code> Code <cite> Citation Deleted text <ins> Inserted text <sub> Subscript <sup> Superscript <bdo> Text direction 	Common Character Entities <ul style="list-style-type: none"> &#34; " Quotation mark &#38; & Ampersand &#60; < Less than &#62; > Greater than &#64; @ "At" symbol &#128; € Euro &#149; • Small bullet &#153; ™ Trademark &#163; £ Pound &#160; Non-breaking space &#169; © Copyright symbol 	Form Events <ul style="list-style-type: none"> onBlur onReset onChange onSelect onFocus onSubmit
		Keyboard Events <ul style="list-style-type: none"> onKeyDown onKeyUp onKeyPress
		Mouse Events <ul style="list-style-type: none"> onClick onMouseout onDbclick onMouseover onMouseDown onMouseup onMouseMove

PARTE 2: Laboratório (Mão na massa!!)

Para começar, vamos primeiramente preparar o ambiente dentro do VS Code.

- 1) Utilizando o editor VS Code no seu pendrive (recomendável) com o nome **Curso-HTML5-2023**.
- 2) Para isto crie uma pasta no seu pendrive com o nome **Curso-HTML5-2023**.
- 3) Dentro da pasta **Curso-HTML5-2023** crie outra pasta **Aula07-HTML e CSS-VS Code**
- 4) Para cada exercício criar um arquivo (Exemplo81.html, Exemplo82.html, Exemplo83.html,.....Exempl85.html) para melhor organização, como mostra a Figura b
- 5) **BOA PRÁTICA!!!**

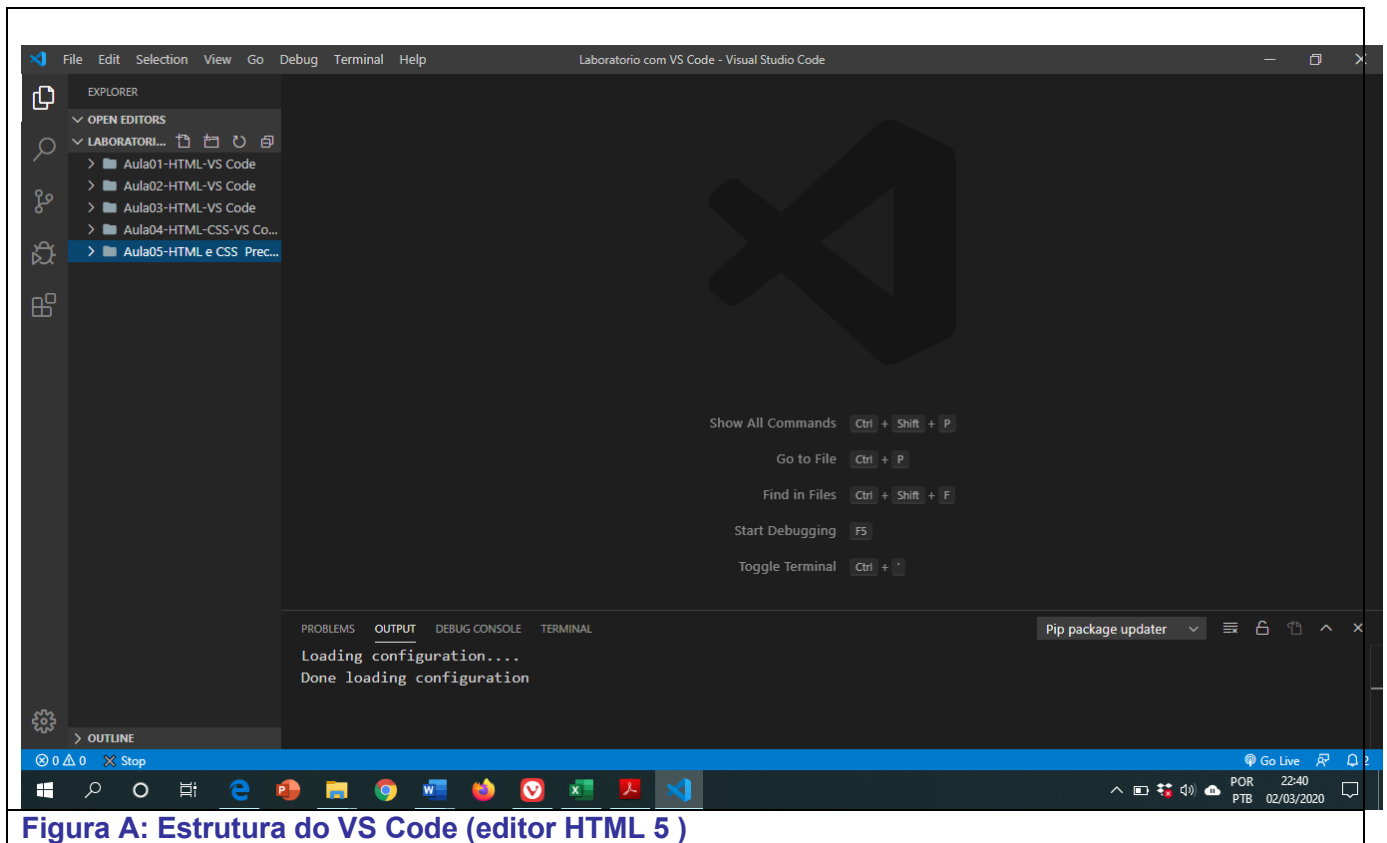


Figura A: Estrutura do VS Code (editor HTML 5)

PASTA: Curso-HTML5-2023.

Aula08-HTML e CSS-VS Code \ Exemplo81-layout

File(arquivo): Exemplo81.html

File(arquivo): estilo81.css

```
body
{
    text-align: center; /* centraliza TUDO inclusive o container*/
}

#container
{
    width: 960px;
```

```
        margin: 0 auto;
        text-align: left; /*volta para a esquerda*/
    }

#header
{
    background-color: Gray;
    height: 119px;
    text-align: center;
    font-size: 30px;
    color: darkgreen;
}

#mainnav
{
    background-color: #ffd800;
    color: #272900;
    padding: 2px 0;

}

#mainnav li
{
    display: inline;
    padding: 0 10px;
    border-right: 1px solid #006;
}

#mainnav ul
{
    margin: 0;
    padding: 0;
}

#mainnav li a
{
    text-decoration: none;
    color: #272900;
}

#mainnav li a:hover
{
    text-decoration: none;
    color: #fff;
    background-color: #272900;
}

#menuLateral
{
    background-color: lightslategrey;
    color: #e6e3e3;
    height: 366px; /* 550 altura dividido por 3 */
    width: 320px;
    float: left;
    padding-top: 20px;
}

#conteudo
{
    background-color: #e6e3e3;
```

```
height:366px;/* 550 altura dividido por 3*/
width:640px; /*960/3=320*2 = 640 */
/*padding-left: 20px; /* descontar do total width*/
padding-top: 20px;
float:left;
}
```

```
#footer
```

```
{
  background-color:gray;
  height:61px;
  clear:both;
  text-align: center;
}
```

estilo81.css

```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, e CSS3" />
    <meta name="Keywords" content="CSS3, Div, Classes e Span" />
    <title>Aula 08 - Exemplo 81 (Curso de HTML 5 e CSS 3 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estilo81.css">
  </head>
  <body>

    <div id="container">

      <div id="header">
        <h1 style="margin-top:20px;">Primeiro exemplo de Layout</h1>
      </div>

      <div id="mainnav">
        <ul>
          <li><a href="home.html">Home</a></li>
          <li><a href="quemsomos.html">Quem somos</a></li>
          <li><a href="mapa.html">Mapa do Site</a></li>
          <li><a href="localizacao.html">Localização</a></li>
          <li><a href="contato.html">Contato</a></li>
        </ul>
      </div>

      <div id="menuLateral">
        <b>Navegação Lateral</b><br>
        Opção1<br>
        Opção2<br>
        Opção3
      </div>

      <div id="conteudo">
        Conteúdo Principal
      </div>

      <div id="footer">
        Copyright © layout com divs - Todos os direitos reservados
      </div>
    </div>
  </body>
</html>
```

```
</div>
</div>

</body>
</html>
```

Exemplo81.html

PASTA: Curso-HTML5-2023.

Aula08-HTMLeCSS-VS Code \ Exemplo82-layout

File(arquivo): Exemplo82.html

File(arquivo): estilo82.css

```
h1
{
  font-family: "Open Sans", "sans-serif";
  text-align: center;
}

h2, h3
{
  font-family: "Open Sans", "sans-serif";
  text-align: left;
}

#cabecalho
{
  background-color:lightgrey;
  clear: both;
  color:#039;
  font-family: "Open Sans", "sans-serif";
}

div ul li
{
  list-style: none;
}

a {
  color: #0066ff;
}

a:hover
{
  background-color:#fff;
  color: #191970;
}

#logo
{
  background-color:#9cf;
}

#mainnav
{
  background-color:#ffd800;
  color: #272900;
  padding: 2px 0;
```

```
}

#mainnav li
{
  display: inline;
  padding: 0 10px;
  border-right: 1px solid #006;
}

#mainnav ul
{
  margin: 0;
  padding: 0;
}

#mainnav li a
{
  text-decoration: none;
  color: #272900;
}

#mainnav li a:hover
{
  text-decoration: none;
  color: #fff;
  background-color: #272900;
}

#menu
{
  background-color: lightyellow;
}

#conteudo
{
  color: #000;
  background-color: #fff;
  font-family: "Open Sans", "sans-serif";
}

.item_menu
{
  background-color: #039;
  color: #0066ff;
  font-family: "Open Sans", "sans-serif";
}

#rodape
{
  background-color: lightgrey;
  clear: both;
  color: #039;
  font-family: "Open Sans", "sans-serif";
  text-align: center;
}
```

Listagem A1: estilo82.css

```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, e CSS3" />
    <meta name="Keywords" content="CSS3, Div, Classes e Span" />
    <title>Aula 08 - Exemplo 82 (Curso de HTML 5 e CSS 3 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estilo82.css">
  </head>
<body>
  <div id="logo">
    
  </div>

  <div id="cabecalho">
    <h1> AULA 08 – Layout com CSS3 </h1>
    <h2> Curso de HTML 5 e CSS 3 com VS Code </h2>
  </div>

  <div id="mainnav">
    <ul>
      <li class="item_menu"><a href="#" title="">Atendimento</a></li>
      <li class="item_menu"><a href="#" title="">Meus Pedidos</a></li>
      <li class="item_menu"><a href="#" title="">Carrinho de Compras</a></li>
    </ul>
  </div>

  <h1> Minha Loja Virtual </h1>

  <div id="menu">
    <ul>
      <li class="item_menu"><a href="#" title="">Cinema e Fotografia</a></li>
      <li class="item_menu"><a href="categoriaVideo.html" title="">Vídeo </a></li>
      <li class="item_menu"><a href="#" title="">Áudio</a></li>
      <li class="item_menu"><a href="#" title="">Informática</a></li>
    </ul>
  </div>

  <div id="conteudo">
    <h2> Principais Ofertas </h2>
    <h3> TV </h3>
    <h3> Notebook </h3>
  </div>

  <div id="rodape">
    <p> Loja Virtual / CNPJ: 00000000/0001-00 </p>
    <p> Endereço: Rua sem nome, 100 - Estado, UF - CEP </p>
  </div>

</body>
</html>
```

Exemplo82.html

PASTA: Curso-HTML5-2023.**Aula08-HTML5-CSS-VS Code \ Exemplo83-layout****File(arquivo): Exemplo83.html****File(arquivo): estilo83.css**

```
/* Utilizando regra dos Terços*/
body
{
    text-align: center; /* centraliza TUDO inclusive o container*/
}
#container
{
    width: 960px;
    margin: 0 auto;
    text-align: left; /*volta para a esquerda*/
}

#header
{
    background-color: Gray;
    height: 119px;
}

#mainnav
{
    background-color: #ffd800;
    color: #272900;
    padding: 2px 0;
}

#mainnav li
{
    display: inline;
    padding: 0 10px;
    border-right: 1px solid #006;
}

#mainnav ul
{
    margin: 0;
    padding: 0;
}

#mainnav li a
{
    text-decoration: none;
    color: #272900;
}

#mainnav li a: hover
{
    text-decoration: none;
    color: #fff;
    background-color: #272900;
}

#menu
```

```
{
  background-color:#000;
  color:#e6e3e3;
  height:366px;/* 550 altura dividido por 3*/
  width:320px;
  float:left;
  padding-top: 20px;
}

#content
{
  background-color:#e6e3e3;
  height:366px;/* 550 altura dividido por 3*/
  width:640px; /*960/3=320*2 = 640 */
  /*padding-left: 20px; /* descontar do total width*/
  padding-top: 20px;
  float:left;
}

#footer
{
  background-color:gray;
  height:61px;
  clear:both;
}
```

Estilo83.css

```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, e CSS3" />
    <meta name="Keywords" content="CSS3, Div, Classes e Span" />
    <title>Aula 08 - Exemplo 82 (Curso de HTML 5 e CSS 3 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estilo83.css">
  </head>
  <body>

    <div id="container">

      <div id="header">
        <h1 style="margin-top:20px;">Título da página</h1>
      </div>

      <div id="mainnav">
        <ul>
          <li><a href="home.html">Home</a></li>
          <li><a href="quemsomos.html">Quem somos</a></li>
          <li><a href="mapa.html">Mapa do Site</a></li>
          <li><a href="localizacao.html">Localização</a></li>
          <li><a href="contato.html">Contato</a></li>
        </ul>
      </div>
      <div id="menu">
        <b>Navegação Lateral</b><br>
```

```
Opção1<br>
Opção2<br>
Opção3
</div>

<div id="content">
Conteúdo Principal
</div>

<div id="footer">
Copyright © layout com divs
</div>

</div>

</body>
</html>
```

Exemplo83.html

PASTA: Curso-HTML5-2023.

Aula08-HTMLeCSS-VS Code \ Exemplo83-layout

File(arquivo): Exemplo84.html

File(arquivo): estilo84.css

```
/* Utilizando regra dos Terços*/
body
{
    text-align: center; /* centraliza TUDO inclusive o container*/
}
#container
{
    width: 960px;
    margin: 0 auto;
    text-align: left; /*volta para a esquerda*/
}

#header
{
    background-color: Gray;
    height: 119px;
}

#mainnav
{
    background-color: #ffd800;
    color: #272900;
    padding: 2px 0;
}

#mainnav li
{
    display: inline;
    padding: 0 10px;
    border-right: 1px solid #006;
}

#mainnav ul
```

```
{
  margin: 0;
  padding: 0;
}

#mainnav li a
{
  text-decoration: none;
  color: #272900;
}

#mainnav li a:hover
{
  text-decoration: none;
  color: #fff;
  background-color: #272900;
}

#menu
{
  background-color:#000;
  color:#e6e3e3;
  height:366px;/* 550 altura dividido por 3*/
  width:320px;
  float:left;
  padding-top: 20px;
}

#content
{
  background-color:#e6e3e3;
  height:366px;/* 550 altura dividido por 3*/
  width:640px; /*960/3=320*2 = 640 */
  /*padding-left: 20px; /* descontar do total width*/
  padding-top: 20px;
  float:left;
}

#footer
{
  background-color:gray;
  height:61px;
  clear:both;
}
```

Estilo83.css

```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina"
  />
    <meta name="description" content="HTML, e CSS3" />
    <meta name="Keywords" content="CSS3, Div, Classes e Span" />
```

```
<title>Aula 08 - Exemplo 82 (Curso de HTML 5 e CSS 3 com
Brackets)</title>

<link rel="stylesheet" type="text/css" href="estilo83.css">
</head>
<body>

<div id="container">

<div id="header">
<h1 style="margin-top:20px;">Título da página</h1>
</div>

<div id="mainnav">
<ul>
<li><a href="home.html">Home</a></li>
<li><a href="quemsomos.html">Quem somos</a></li>
<li><a href="mapa.html">Mapa do Site</a></li>
<li><a href="localizacao.html">Localização</a></li>
<li><a href="contato.html">Contato</a></li>
</ul>
</div>

<div id="menu">
<b>Navegação Lateral</b><br>
Opção1<br>
Opção2<br>
Opção3
</div>

<div id="content">
<form action="/pagina-processa-dados-do-form" method="post">
<div>
<label for="nome">Nome:</label>
<input type="text" id="nome" />
Masculino <input type="radio" name="sexo" value="masculino" />
Femenino <input type="radio" name="sexo" value="fememino" />
</div>
<div>
<br>
<label for="Senha">Senha:</label>
<input name="_senha" type="password" />
</div>
<div>
<br>
<label for="email">E-mail:</label>
```

```
        <input type="email" id="email" />
    </div>
    <div>
        <label for="SelecaoCidade">Selecione uma cidade:</label>
        <select name="cidade" id="cidade">
            <option value="Campinas selected="selected">Campinas</option>
            <option value="Sumare">Sumare</option>
            <option value="Indaiatuba" > Indaiatuba</option>
            <option value="Jundiai">Jundiai</option>
        </select>
    </div>

    <div>
        <br>
        <label for="msg">Mensagem:</label>
        <textarea id="msg"></textarea>
    </div>

    <div>
        <button type="submit"> Enviar </button>
        <button type="reset"> Resetar </button>
    </div>

</form>

</div>

<div id="footer">
Copyright © layout com divs
</div>

</div>

</body>
</html>
```

Exemplo84.html