

Aula 08: Posicionamento com CSS

E outras Técnicas Avançadas

Prof. Daniel Caetano

Objetivo: Apresentar recursos adicionais do CSS e como usá-lo para a construção de layout de página.

Bibliografia: W3, 2009; CASCADE, 2006; RAMALHO, 1999; NIELSEN, 2000.

INTRODUÇÃO

Conceitos Chave:

- Estilos "Avançados"
 - * Transparência, links visitados etc...
- CSS => não serve apenas para mudar estilos!
 - * Posicionamento de Elementos!
 - * Mudar o alinhamento... só?
- Auxílio do HTML
 - * É preciso indicar o "início" e o "fim" de cada elemento.
 - * É preciso ordenar estes elementos
 - + Tags de Divisão de documento

Anteriormente, foi apresentado como usar as CSSs para alterar a aparência de nosso documento. Muitas propriedades do documento podem ser alteradas, como cor de fundo, cor de texto, dentre outras.

Entretanto, o mecanismo visto para modificação de propriedades é um tanto quanto limitado com relação ao posicionamento dos diversos elementos de uma página: podemos mudar alinhamentos e, talvez, a ordem de alguns elementos. Entretanto, se quisermos posicionar o conteúdo de alguma forma menos tradicional, não será possível apenas com o que já vimos até agora.

O objetivo desta aula é, então, apresentar mais alguns recursos do CSS e como definir elementos que permitam posicionar diferentes "blocos" de uma página Web no lugar desejado. Como será visto, tudo isso será feito primordialmente no CSS, com poucas modificações no documento HTML.

1. PSEUDO-CLASSES E PSEUDO-ELEMENTOS

Conceitos Chave:

- Definindo estilos específicos de ações
 - * tag:estado { ... }
- Exemplos
 - a:hover
 - a:visited
 - p:first-child

Alguns elementos do HTML, com a tag <a>..., que define um link, possuem diversos "estados": um link pode ser um link não visitado, um link já visitado, um link com o ponteiro do mouse sobre ele, e assim por diante. Assim, para definir características específicas em cada um destes estados, é necessário acrescentar uma informação no nome da tag indicada no CSS.

```
elemento:estado {  
    propriedade: valor;  
}
```

Estes "estados" são chamados de "pseudo-classes" e as mais comuns são:

:active	Especifica estilo para um elemento ativo (ex.: a:active)
:focus	Especifica estilo para elemento em foco (ex.: input:focus)
:hover	Especifica estilo para elemento com mouse sobre ele (ex.: a:hover)
:link	Especifica estilo para link não visitado (a:link)
:visited	Especifica estilo para link já visitado (a:visited)
:first-child	Especifica estilos diferentes internos a uma primeira ocorrência de um elemento em uma região da página.

Por exemplo: para fazer com que um link não visitado seja verde e sublinhado, mas se torne vermelho e sublinhado quando o mouse passar por cima, usa-se o seguinte CSS:

```
a:link {  
    color: green;  
    text-decoration: none;  
}  
  
a:hover {  
    color: red;  
    text-decoration: underline;  
}
```

1.1. Pseudo-Elementos

Os pseudo-elementos são muito similares às pseudo-classes, mas definem o comportamento de elementos que não estão claramente definidos no HTML. Os pseudo-elementos mais comuns são:

:first-letter	Especifica estilo para a primeira letra de uma tag (ex.: p:first-letter)
:first-line	Especifica estilo para a primeira linha de uma tag (ex.: p:first-line)
:before	Insere algum conteúdo (áudio? vídeo?) antes de um elemento (url:)
:after	Insere algum conteúdo (áudio? vídeo?) depois de um elemento (url:)

2. TRANSPARÊNCIA

Conceitos Chave:

- Transparência/Opacidade de imagens
 - * CSS3 ($0.0 \leq x \leq 1.0$)
 - opacity: x

Algumas propriedades ainda não padronizadas pelo CSS2 já estão disponíveis nos navegadores. Estas propriedades estão padronizadas no CSS3, mas como o CSS3 não é, ainda, exatamente oficial, cada navegador implementa estas propriedades de um jeito, obrigando o programador a especificar o mesmo efeito em mais de um formato dentro do arquivo CSS.

Uma destas propriedades que é muito importante é a transparência, sendo este um dos mais desejados pelos web masters. Ele é conseguido através do seguinte atributo: opacity: x

Considere a página a seguir, que contém um pano de fundo, um texto em um H1 e um texto dentro de um parágrafo. Observe como o texto tem baixa legibilidade quando se encontra sobre a figura do planeta Terra, devido ao baixo contraste entre as cores da imagem e a cor usada no texto.

exemplo.html

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <title>Minha página pessoal.</title>
  <link rel="stylesheet" href="exemplo.css" />
</head>
<body>
  <h1>Minha Página Pessoal HTML com CSS Posicional</h1>
  <p>  Meu nome é Daniel Caetano e sou sócio-fundador do grupo
    Amusement Factory Software, sendo também professor das seguintes
    disciplinas:
  </p>
</body>
</html>
```

exemplo.css

```
body {
  background-color: rgb(0,0,0);
  background-image: url(earth2.jpg);
  color: rgb(255,255,255);
}

p {
  background-color: rgb(0,0,0);
}
```



É possível que o WebMaster não queira um fundo "bloco preto" para seu parágrafo, e considere interessante poder defini-lo com um bloco preto com 50% de transparência,

permitindo que o fundo seja visível através do bloco. Para isso, basta a seguinte modificação no arquivo .CSS:

exemplo.css

```
body {  
  background-color: rgb(0,0,0);  
  background-image: url(earth2.jpg);  
  color: rgb(255,255,255);  
}  
  
p {  
  background-color: rgb(0,0,0);  
  opacity: 0.5;  
}
```

O resultado é apresentado abaixo.



Observe, porém, que em todos os casos a transparência está sendo "herdada" pelo texto, o que a torna quase inútil... mas ela é a única que se aplica tanto a imagens quanto a cor de fundo.

Quando quisermos deixar apenas a cor de fundo de um elemento transparente, podemos definir essa cor com o elemento **rgba(vermelho, verde, azul, transparência)**. Exemplo:

exemplo.css

```
body {  
  background-color: rgb(0,0,0);  
  background-image: url(earth2.jpg);  
  color: rgb(255,255,255);  
}  
  
p {  
  background-color: rgba(0,0,0,0.5);  
}
```

O resultado será esse:



3. DIVISÃO DE DOCUMENTO

Conceitos Chave:

- DIV => <DIV ID="nome">...</DIV>
 - * Regiões a Reposicionar x Redefinir Estilos
 - * NAV, SECTION, ARTICLE, HEADER, FOOTER
- Exemplo

Até este momento, trabalhamos com corpo do documento HTML vimos como dividir um documento em partes ("menu", "cabeçalho do conteúdo", "área de notícias", "área de conteúdo"), mas não houve mais detalhes sobre a utilidade disso...

Ora, se queremos posicionar cada uma destas partes de maneira independente, isto é, queremos indicar exatamente onde cada uma delas deve estar, é para isso que precisamos dividir o documento HTML em todas estas partes.

As tags <div>, <section>, <article>, <nav>, <header> e <footer> são responsáveis por uma divisão lógica do documento, ou seja: um único arquivo, mas as **tags** indicam o que cada trecho representa. O corpo da página, ao invés de um elemento único, pode ser interpretado como um conjunto de blocos.

Pensando assim, passa a ser natural que possamos dar características próprias a cada um dos elementos, não só de posição, mas também de cores, fontes etc. Para tanto, a única modificação que será feita no HTML é justamente a adição de algumas tags que indicarão o que cada trecho da página representa. Por exemplo, na página abaixo:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <title>Título da Página</title>
  <link rel="stylesheet" href="exemplo.css" />
</head>
<body>

  <h1>Título da Página</h1>
  <hr />
  <h3>Menu</h3>
  <p>  <li><a href="...">Item 1</a></li>
      <li><a href="...">Item 2</a></li>
      <li><a href="...">Item 3</a></li>
  </p>
  <h2>Conteúdo</h2>
  <p>Esta é uma página pessoal!</p>
</body>
</html>
```

Claramente existe uma seção de "cabecalho do conteúdo", que é composto pelo <h1>...</h1> e o <hr>. Em seguida, temos um menu, composto pelo <h3> e o primeiro <p>...</p> e, finalmente, temos o conteúdo da página, composto pelo <h2> e o segundo <p>...</p>. Embora seja possível perceber isso analisando o documento, isso não está claramente identificado.

A identificação clara é precisa é o que será feita agora, usando a **tag** de *Divisão de documento*, a tag <DIV>...</DIV> ou, como recomendado pelo W3C para páginas HTML5, usar as tags mais específicas sempre que adequado: <section>, <article>, <nav>, <header> e <footer>. Todas essas tags definem uma região, elas possuem um início e um final. O documento marcado com as tags fica como indicado abaixo:

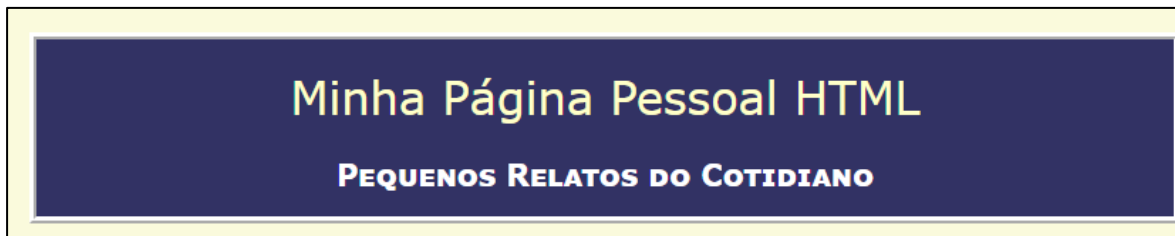
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <title>Título da Página</title>
  <link rel="stylesheet" href="exemplo.css" />
</head>
<body>
  <header>
    <h1>Título da Página</h1>
    <hr />
  </header>
  <nav>
    <h3>Menu</h3>
    <p>
      <li><a href="...">Item 1</a></li>
      <li><a href="...">Item 2</a></li>
      <li><a href="...">Item 3</a></li>
    </p>
  </nav>
  <section>
    <h2>Conteúdo</h2>
    <p>Esta é uma página pessoal!</p>
  </section>
</body>
</html>
```

Caso essa identificação não seja suficiente, podemos dar um nome, uma identificação, para cada seção. Isso pode ser feito com o parâmetro **ID="nome_da_seção"**. Assim, o documento anterior poderia ficar da seguinte forma:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <title>Título da Página</title>
  <link rel="stylesheet" href="exemplo.css" />
</head>
<body>
  <header>
    <h1>Título da Página</h1>
    <hr />
  </header>
  <nav>
    <h3>Menu</h3>
    <p>
      <li><a href="...">Item 1</a></li>
      <li><a href="...">Item 2</a></li>
      <li><a href="...">Item 3</a></li>
    </p>
  </nav>
  <section id="conteudo">
    <h2>Conteúdo</h2>
    <p>Esta é uma página pessoal!</p>
  </section>
</body>
</html>
```

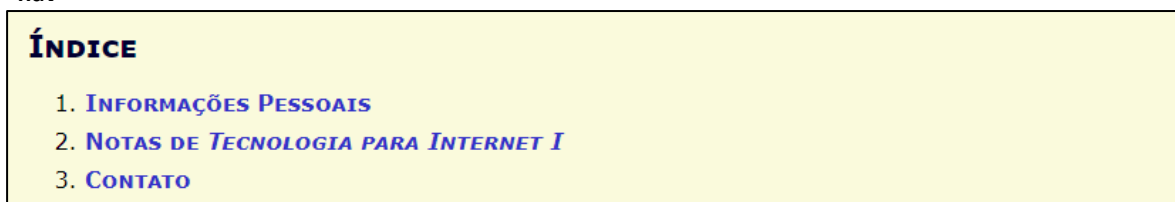

Um comentário importante é que as tags citadas são “estruturantes”, ou seja, só terão efeito visual no navegador se realizarmos mudanças efetivas na mesma, por meio do arquivo CSS. Caso contrário, será como se o navegador simplesmente tivesse ignorado a tag. Na página definida na aula anterior, podemos definir várias seções. Por exemplo:

```
<header>
```



```
</header>
```

```
<nav>
```



```
</nav>
```

```
<section id="pessoal">
```

1. INFORMAÇÕES PESSOAIS



Meu nome é Daniel Caetano e sou sócio-fundador da equipe Amusement Factory Software, sendo também professor das seguintes disciplinas:

- Informática para Engenharia
- Mecânica dos Sólidos
- Mobilidade e Sistemas de Transportes
- Pesquisa Operacional I
- Tecnologia para Internet I

```
</section>
```

```
<section id="notas">
```

2. NOTAS DOS ALUNOS DE TECNOLOGIA PARA INTERNET I

Tabela 1: Notas de 2019/02

NOME DO ALUNO ⁽¹⁾	NOTA FINAL
Sem Dados	Sem Dados
Sem Dados	Sem Dados

```
</section>
```

```
<section id="contato">
```

3. CONTATO

Caso precise entrar em contato comigo, [ENVIE-ME UM E-MAIL](#).

```
</section>
```

```
<footer>
```

Esta página é de autoria de Daniel Caetano, 2019. Todos os direitos reservados.

```
</footer>
```

Ao carregar esta página no navegador, ela será exibida exatamente como antes. Entretanto, isso só ocorre porque não realizamos qualquer mudança no arquivo CSS. Mas o que deveríamos modificar no arquivo CSS? Resumidamente, devemos indicar **o quê** estas seções devem possuir de diferente!

4. MODIFICANDO SEÇÕES COM CSS

Conceitos Chave:

- Definindo estilos de seções
 - * #secao { ... }
- Exemplos
 - * Titulo
 - * Indice
 - * Secao1, secao2, secao3

Como agora não iremos mais modificar apenas a apresentação de algumas tags do HTML e sim como serão apresentados pedaços inteiros do documento. Podemos modificar a tag ou, como já vimos com os IDs:

```
#secao {  
  ...  
}
```

Como uma regra geral, também aqui o CSS é muito chato com a sintaxe. Um pequeno erro de digitação, um ponto-e-vírgula faltando ou algo do gênero pode ser responsável por sua página não aparecer corretamente. Por esta razão, muita atenção ao digitar o arquivo .CSS!

Nas próximas páginas serão apresentados alguns exemplos de modificação de propriedades e posicionamento através de CSS.

4.1. Exemplos Aplicando CSS para Posicionamento

Veja o exemplo da aula, com os três diferentes CSSs, como é possível modificar o visual radicalmente.

5. BIBLIOGRAFIA

CASCADE Style Sheets, level 2 revision 1: CSS 2.1 Specification - W3C Working Draft 06 November 2006.
Disponível em < <http://www.w3.org/TR/CSS21/> >. Visitado em 21 de Dezembro de 2006.

W3 schools - CSS Tutorial. Disponível em < <http://www.w3schools.com/> >. Visitado em 10 de Março de 2009.

RAMALHO, J.A. *HTML 4 Prático e Rápido*. Editora Berkeley, 1999.

BOENTE, A. *Programação Web Sem Mistérios*. Editora Brasport, 2006.

NIELSEN, J. *Projetando Websites*. Ed. Campus, 2000.c