

TECNOLOGIAS WEB

**FOLHAS DE ESTILOS EM
CASCATA**

Olá!

Nesta aula, iremos aprender como construir e utilizar os conceitos e tags XHTML usando Folhas de Estilos em Cascata para a construção de páginas Web. Veremos, ainda, como aperfeiçoar nosso código para garantir maior flexibilidade empregando estes recursos.

Ao final desta aula, você será capaz de:

- Apresentar as principais características das Aula 09: Folhas de Estilos em Cascata
- Compreender a sintaxe utilizada em Aula 09: Folhas de Estilos em Cascata, seu uso e limitações.
- Abordar os diversos tipos de folhas de estilo
- Compreender as principais de soluções para transformar seu site
- Aplicar Aula 09: Folhas de Estilos em Cascata na construção de sites

1 O que são folhas de estilos em cascata?

Folhas de Estilo em Cascata ou CSS (Cascade Style Sheet) são estilos aplicados em HTML. A linguagem HTML, na sua concepção, não foi projetada para ser utilizada na elaboração de páginas com layouts sofisticados. Como a HTML é limitada, foi necessário desenvolver uma alternativa capaz de manipular os elementos para garantir que uma página não fosse mais textos e imagens corridas ou, no máximo, contidas em tabelas.

As Folhas de Estilos em Cascatas têm como funções:

- Definir a aparência das páginas em um único lugar, em vez de repetir-se uma e outra vez no site.
- Facilmente alterar a aparência das páginas, mesmo depois que eles são criados.
- Definir o tamanho da fonte e os atributos semelhantes com a mesma precisão que você tem com um processador de texto.
- Definir a posição do conteúdo das páginas com precisão de pixel.
- Definir estilos personalizados.
- Definir as camadas sem a necessidade de pop-ups.

2 Aplicando Folhas de Estilo em Cascata

Existem 3 formas de se aplicar estilo às páginas em HTML.

- In-line
- Interna
- Externa

Veremos as três, a seguir, mas, antes, vejamos como é composta a sintaxe em CSS.

A sintaxe do CSS segue o formato Seletor com uma ou mais declarações:

Seletor	Declaração		Declaração	
H1	{color:	red;	font-size:	12px;}
	Propriedade	Valor	Propriedade	Valor

Tudo deve ser descrito em minúsculas.

Desta forma, pode-se utilizar mais de uma declaração para o mesmo seletor. Isto é muito mais poderoso do que se podia fazer em HTML puro.

Para se declarar um estilo, tem-se que determinar como ele será implementado no código: junto ao elemento, interno ao código ou externo ao código.

3 Formas de se aplicar estilo as páginas em HTML

3.1 In-line

Esta forma descreve o estilo diretamente na tag HTML correspondente. Isto, por um lado, tende a facilitar o processo, mas, por outro, dificulta a manutenção do código.

Exemplo:
`<p style="color: red">texto</p>`

Neste caso, aplica-se, em todo o parágrafo, o estilo com a cor vermelha. Veja o resultado disso na figura:

```

Código html
<html>
<head>
</head>
<body>
< p style="color: red">texto</p>
Repare que o resto do texto na página não foi afetado
</body>
</html>

```

Figura 1 - Código html

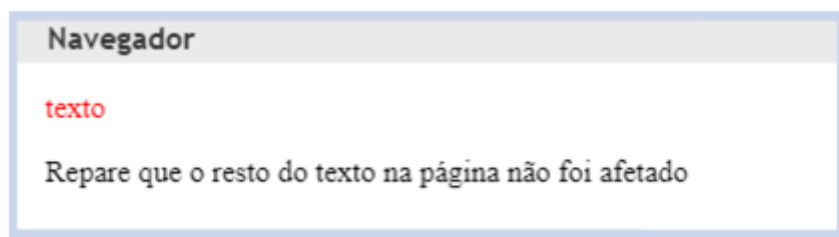


Figura 2 - Navegador

3.2 Interno

Também pode-se implementar os estilos dentro do código da página. A diferença em relação ao anterior é que, neste caso, o estilo não é definido direto no elemento (tag), mas em uma parte do código cuja aplicação é indicada através de recursos de identificação. Assim, a manutenção da página fica mais fácil, pois, ao se alterar o estilo, todos os elementos associados a ele passam a sofrer essa alteração.

Para que se defina os estilos internos da página, deve-se criá-los dentro de tags `<style..> </style>`, dentro do bloco `<head></head>`, como no exemplo a seguir.

```
Código html
<html>
<head>
<style type="text/css">
p
{
color:red;
}
</style>
</head>

<body>
<p>Primeiro Parágrafo</p>
<p>Segundo Parágrafo. Este também foi alterado, pois o
estilo aplicado a parágrafo é quem manda.</p>
</body>
</html>
```

Figura 3 - Código html

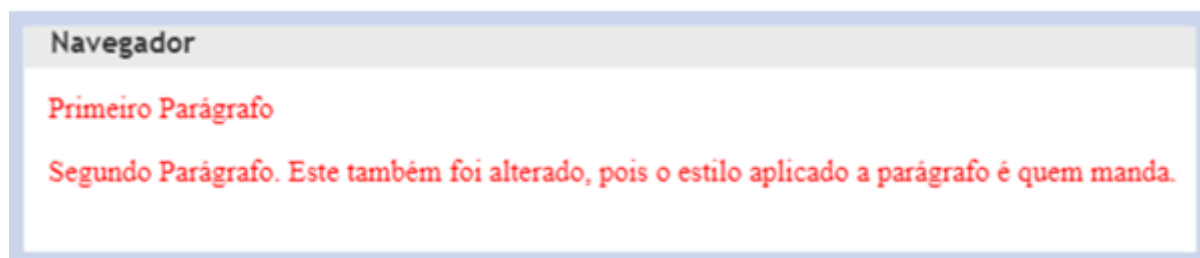


Figura 4 - Navegador

Repare que o type é text/css. Além do type, pode-se definir para que tipo de mídia (media) se quer aplicar estes estilos. Isso é particularmente importante quando queremos diferenciar os estilos por dispositivos, no caso da acessibilidade.

Podemos ter as seguintes mídias:

all	todos
aural	sintetizadores de voz
braille	para dispositivos adaptados ao Braille
handheld	para <i>palms</i> e outros tipos de dispositivos portáteis
print	para impressão ou visualização para impressão
projection	para projetores do tipo data show
screen	para monitores
tty	para teletipos
tv	para televisores

A omissão da declaração media determina o valor padrão all.

Exemplo:

No exemplo abaixo, são indicados estilos diferentes para dois tipos de mídias. Neste caso, se a mídia for screen (monitor), a fonte tem tamanho 25px (25 pixels) em negrito e, se for para impressão ou visualização para impressão, passa para 10px, também em negrito.

```
Código html

<html>
<head>
<style>
@media screen
{
p.exemplo_media{font-family:verdana, sans-serif;font-size:25px}
@media print
{
p.exemplo_media{font-family:times, serif;font-size:10px}
@media screen, print
{
p.exemplo_media{font-weight:bold}
}
}
</style>
</head>

<body>
<p class="exemplo_media">Exemplo para mídias diferentes</p>
Sem alteração nos textos restantes
</body>
</html>
```

Figura 5 - Código html



Figura 6 - Navegador

Na saída para visualização da impressão, conforme ilustrado abaixo, repare como a fonte de 25px caiu para 10px. Isto significa que este estilo, adaptado para diferentes mídias, funcionou.



Figura 7 - Código html

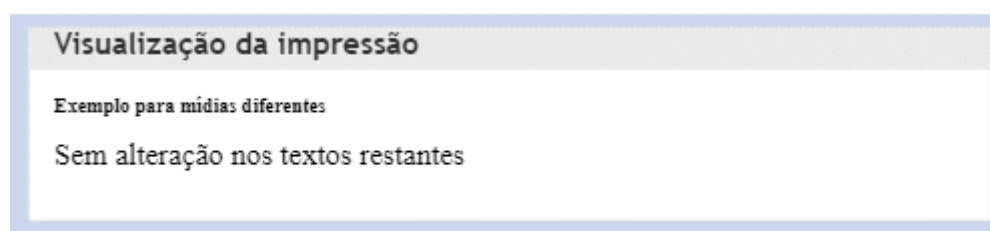


Figura 8 - Visualização da impressão

3.3 Externo

Esta, com toda a certeza, é a maneira mais prática de se manter um portal. Através de arquivos externos, economiza-se na manutenção de dezenas ou centenas de páginas, pois basta que se altere um arquivo para que tudo aconteça no portal como um todo.

O arquivo css deve ser um arquivo com extensão css e estar dentro do contexto da sua aplicação, isto é, dentro de um diretório onde o servidor Web possa alcançá-lo para o seu navegador.

Veja no exemplo que segue. Será criado um arquivo chamado `estilo_portal.css` com o conteúdo abaixo.

estilo_portal.css

```
body {background-color: beige}
```

```
h1 {color:blue; font-size:20pt}
```

```
hr {color:navy}
```

```
p {font-size:11pt; margin-left: 15px}
```

```
a:link {color:green}
```

```
a:visited {color:black}
```

```
a:hover {color:red}
```

```
a:active {color:yellow}
```

Olhe como ficou a página.

Ao passar o mouse sobre o link Google ele muda da cor verde para a cor vermelha por conta do hover.

O link que já foi visitado (visited) passa para a cor preta.

No caso dos links ativos (active), o fundo passa para a cor amarela.

Estilos - Externos

Aplicação de estilos externos

Google

A página nomeando o arquivo externo com os estilos

exemplo.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Exemplo CSS</title>
```

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

```
</head>
```

```
<body>
```

```
<h1>Estilos - Externos</h1>
```

```
<p>Aplicação de estilos externos</p>
```

```
<p><a href="http://www.universidade.br" target="_blank">Universidade</a>
```

```
</p>
```

```
</body>
```


</html>

4 Não se esqueça do doctype

Existem vários tipos de HTML: **HTML 4.01 Strict**, **HTML 4.01 Transitional**, **XHTML 1.0 Strict** entre outros.

Para cada situação, é preciso que se defina o tipo de documento a ser interpretado através da declaração de DTD (Document Type Definition). Isto não é complicado, mas precisa ser feito de forma consciente, pois alguns programas editores de código (X)HTML os definem com base no padrão do fabricante do software e não de acordo com padrão pessoal.

Até aqui, teve-se a liberdade de não defini-los, mas, quando se lida com CSS e, por conseguinte, com XHTML, isto se torna imperativo.

Vejamos, agora, a diferença básica de alguns dos tipos de HTML:

- **Strict** – Este padrão impõe regras muito rígidas na descrição dos elementos em (X)HTML.
- **Transitional** – Este é um padrão mais flexível, pois possibilita a escrita de códigos HTML compatíveis com outras versões e, ainda, permite que sejam adaptados aos navegadores Web.
- **Frameset** - Utilizado quando desejamos trabalhar com frames.

Exemplos

- **XHTML 1.0 (Strict, Transitional e Frameset)**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

- **HTML 4.01 (Strict, Transitional e Frameset)**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

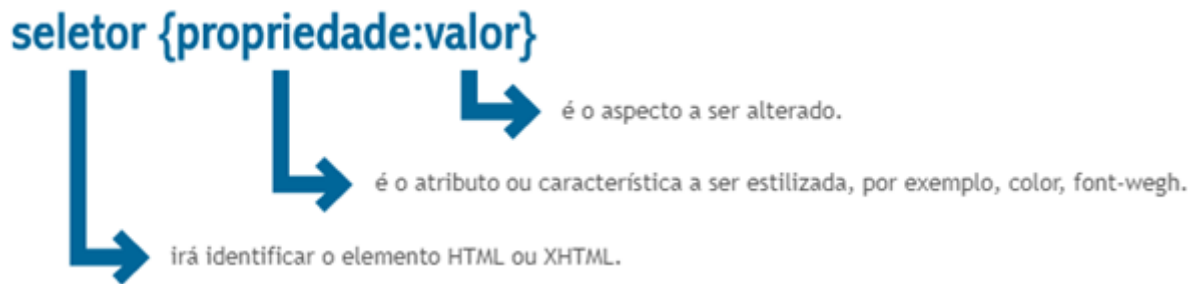
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
```

```
"http://www.w3.org/TR/html4/frameset.dtd">
```

5 Seletores

Seletores estão para CSS como tags estão para HTML. O seletor é uma strings que identifica o elemento HTML ao qual a regra será aplicada.

Trabalhar com seletores é simples. Veja a sintaxe:



Exemplo:

```
body {  
font-size: 0.8em;  
color: navy;  
}
```

Neste caso, aplica-se ao corpo da página uma formatação de texto com letras de tamanho 0.8em na cor navy.

Saiba mais



A respeito das Unidades utilizadas em (X)HTML e CSS: www.w3.org/TR/CSS2/syndata.html#values

6 Tipos de seletores

Existem diversos tipos de seletores, mas, nesta aula, apenas serão abordados alguns dos mais utilizados.

--	--

Simples	<p>Onde a regra se aplica ao tipo e/ou aos atributos do elemento e não à posição do mesmo.</p> <p>Em resumo, é aquele seletor constituído de um só elemento.</p> <p>Exemplo: h1, p.</p>
Contextual	<p>Onde o seletor é associado aos elementos de acordo com as posições que ocupam na estrutura do documento. Na verdade, o seletor contextual é um conjunto de seletores simples.</p> <p>Exemplo: h1.inicio b.</p>
Universais	<p>Aplicável a todos os elementos.</p> <p>Exemplo: * {color : red;}</p>
Classes	<p>Este seletor pode ser aplicado a qualquer elemento (X)HTML, tendo em vista que se possui a prerrogativa de se criar um nome para ele.</p>

Exemplo: .estilo_topo { color: navy;} – É importante perceber que, neste caso, criou-se uma classe chamada estilo_topo. Dessa forma, ao ser aplicado, tem-se a cor navy no elemento associado.

Exemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Exemplo</title>
```

```
<style type="text/css">
```

```
/* Exemplo de Comentário: Estilo Aplicado somente ao
```

```
Cabeçalho da Página
```

```
*/
```

```
.estilo_cabecalho
```

```
{
```

```
color:navy;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1 class="estilo_cabecalho">Cabeçalho Página</h1>

</body>

</html>
```

7 Pseudo-classes

São complementações feitas aos seletores que são criadas para acessar as informações que não estão disponíveis na árvore do documento.

Sintaxe:

seletor:pseudo-classe {propriedade:valor} ou, ainda, seletor.classe:pseudo-classe {propriedade:valor}

Âncora

```
a:active {color:#0000FF} /* link selecionado */
a:link {color:#FF0000} /* link ainda não visitado*/
a:visited {color:#00FF00} /* link visitado*/
a:hover {color:#FF00FF} /* mouse sobre o link */
```

O a:hover deve proceder o a:link e o a:visited para funcionar.

O a:active deve vir sempre após o a:hover.

- **:first-child**

Define um elemento específico que é o primeiro filho de um elemento.

Exemplo:

<http://www.w3.org/TR/html4/loose.dtd>>

```
<html>
<head>
<style type="text/css">
p:first-child
{
color:red
}
</style>
</head>
<body>
```

Exemplo de Pseudo-Class

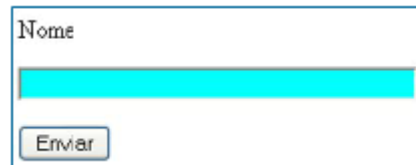
```
<p>Primeiro Parágrafo - Exemplo Pseudo-classe</p>
```

<p>Primeiro Parágrafo - Exemplo Pseudo-classe</p>

</body>

</html>

Veja como ficou:

Um formulário web simples com um campo de texto e um botão. O campo de texto tem o rótulo "Nome" e está preenchido com uma barra laranja. Abaixo dele, há um botão com o texto "Enviar".

<http://www.universidade.br/formulario> method="post">

Exemplo de Pseudo-Classe

Primeiro Parágrafo - Exemplo Pseudo-classe

Primeiro Parágrafo - Exemplo Pseudo-classe

Os pseudo-elementos são:

- **:after** - Adiciona o conteúdo após o elemento.
- **:before** - Adiciona o conteúdo antes do elemento.
- **:first-letter** - Aplica o estilo na primeira letra do texto.
- **:first-line** - Aplica o estilo na primeira linha do texto.

8 ID

Este é um seletor que se identifica com um único elemento (X)HTML, isto é, ele não pode ser aplicado a outro elemento dentro do documento.

Sua identificação é através do sinal # junto ao nome.

<html>

<head>

<title>Exemplo</title>

<style type="text/css">

```

/* Estilo Aplicado somente ao
Cabeçalho da Página
*/
#estilo_cabecalho
{
color:#000080;
background:#ffcc33;
}
</style>
</head>
<body>
<h1 id="estilo_cabecalho">Cabeçalho Página</h1>
<h1> Outro Cabeçalho Página</h1>
</body>
</html>

```

Veja como ficou o resultado:



9 Descendentes ou encadeados

São todos os elementos que descendem de outro.

Exemplo:

```
ul li a {color:#80ff00;}
```

Neste caso, todos os links de todos os itens de uma lista não ordenada serão verde-limão.

Veja como ficou na figura abaixo:

Saiba mais



Saiba mais sobre os diversos seletores em www.w3c.br/divulgacao/guiasreferencia/css2/#mod-seletores

10 Texto

Através das folhas de estilo em cascata, será possível alterar as fontes e as propriedades das mesmas na página.

Para isso, faz-se necessário que se aprenda bem as seguintes propriedades:

- **font-family**

É a fonte propriamente dita.

Ex: Arial, Verdana, Courier New.

Porém, saiba que a fonte escolhida deve estar instalada no microcomputador do usuário. Caso contrário, aparecerão outras fontes em substituição a ela e não se terá o mesmo apelo visual.

Ex: font-family: "Times New Roman"

- **font-size**

Define-se, utilizando esta propriedade: o tamanho da fonte.

Ex: font-size: 12px.

Veja outros exemplos:

1em é igual ao tamanho da fonte corrente. Ex: font-size: 2em

px é a unidade de referência para pixel. Ex: font-size: 12px

pt é a unidade para pontos. Ex: font-size: 12pt%

% é a unidade para porcentual. Ex: font-size: 80%

Dica Importante:

Algumas vezes, é interessante 'zerar' algumas definições. Para isso busque, no Google, zerar css ou, em inglês, reset css.

- **font-weight**

É o peso da fonte.

Ex: font-weight: bold ou font-weight: normal

- **font-style**

É a inclinação do texto. Se ele está, ou não, em itálico.

Ex: font-style: italic ou font-style: normal.

- **text-decoration**

O texto pode ter alguns elementos de marcação.

- text-decoration: overline, apresenta uma linha sobre o texto.
- text-decoration: line-through, apresenta uma linha atravessando o texto.
- text-decoration: underline, sublinha o texto. Cuidado, pois essa opção confunde o usuário com link.

- **text-transformation**

São recursos interessantes, pois pode-se trabalhar maiúsculas, minúsculas e letras capitalizadas.

- text-transform: capitalize, apresenta a primeira letra de cada palavra em maiúscula.
- text-transform: uppercase, apresenta todas as letras em maiúsculas.
- text-transform: lowercase, apresenta todas as letras em minúsculas.
- text-transform: none, resseta qualquer estilo neste contexto.

Exemplo:

```
body {  
  font-family: verdana, helvetica, sans-serif;  
  font-size: 0.8em;  
}  
  
h1 {  
  font-size: 3em;  
}  
  
h2 {  
  font-size: 2.5em;  
}  
  
a {  
  text-decoration: none;  
}  
  
strong {  
  font-style: italic;  
  text-transform: uppercase;  
}
```




- **text-spacing**

Cuida do espaçamento, indentação e alinhamento entre letras e palavras.

- letter-spacing e word-spacing – espaçamento entre letras e palavras respectivamente. O padrão é normal.
- line-height – configura a altura entre as linhas. tal como ocorre com o espaçamento entre parágrafos no editor de texto. Múltiplos do tamanho da fonte ou %. O padrão é normal.
- text-align – alinha o texto pela direita, ao centro, à esquerda ou justificado. Opções: left, right, center e justify.
- text-indent – indica a indentação da primeira linha do parágrafo.

Exemplo:

```
p {  
letter-spacing: 0.75em;  
word-spacing: 2em;  
line-height: 2.0;  
text-align: justify;  
}
```

11 Margens e espaçamentos

São as propriedades mais comumente utilizadas no espaçamento de elementos, em que as margens são espaços fora do elemento e o padding são os espaços dentro do elemento.

Exemplo:

```
p {  
letter-spacing: 0.75em;  
word-spacing: 2em;  
line-height: 2.0;
```

```
text-align: justify;
```

```
}
```

Repare como ficou a aplicação deste código em um elemento h2. Existe uma margem e um espaçamento entre o elemento e a borda da margem.

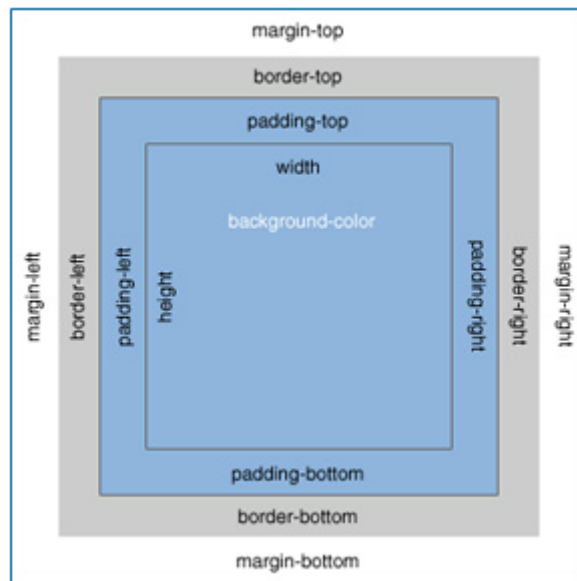


Os 4 lados podem ser trabalhados de forma individual utilizando-se as seguintes propriedades:

margin-top, margin-right, margin-bottom, margin-left, padding-top, padding-right, padding-bottom e padding-left

Elas são explicadas, mais facilmente, pela figura abaixo.

Exemplo de Layout com as propriedades:



12 Bordas

Para adicionarmos uma borda ao redor do elemento, basta trabalharmos com a propriedade border-style cujos valores são:

solid, dashed, dotted, groove, ridge, double, além de inset e outset.

Além dessas propriedades para border-style, devemos trabalhar a espessura da mesma com a propriedade border-width.

Veja nos exemplos abaixo:

```
h2 {  
border-style: dashed; /* estilo de borda pontilhado */  
border-width: 3px; /* espessura de borda 3px */  
border-left-width: 25px; /* espessura da borda a esquerda de 25px */  
border-right-width: 10px; /* espessura da borda a direita de 10px */  
border-color: navy; /* cor da linha da borda azul */  
}
```



```
h1 {  
font-size: 5em; /* fonte tamanho 5 vezes o tamanho da fonte corrente */  
color: #ccff00; /* cor da fonte esverdeada */  
background-color: #006400; /* cor do fundo verde escuro */  
border-top: 2px solid #ffcc00; /* borda superior com espessura de 2px estilo sólido e cor laranja */  
border-bottom: 2px solid #ffcc00; /* borda inferior com espessura de 2px estilo sólido e cor laranja */  
}
```

A imagem mostra um elemento de interface com o texto "Cabeçalho Página" em uma fonte verde amarela. O elemento possui um fundo verde escuro e bordas superior e inferior sólidas de cor laranja, com uma espessura de 2 pixels.

13 Listas de links

Para se criar uma barra de navegação como um menu, faz-se necessário que se conheça mais alguns detalhes. Na verdade, um menu de navegação nada mais é do que uma lista de links.

Veja o código ao lado:

```
p {  
  letter-spacing: 0.75em;  
  word-spacing: 2em;  
  line-height: 2.0;  
  text-align: justify;  
}
```

Existe uma propriedade na definição da lista chamada `list-style-type` que determina como a lista vai se comportar quanto à existência de marcadores. Neste caso, o valor está como `none` tendo em vista que uma barra de navegação não precisa de marcadores.

Além disso, as margens e espaçamentos estão zerados com o intuito de remover os valores definidos nos navegadores Web.

Para listas verticais, o código é outro, pois somente é preciso definir um estilo para a tag `<a>` a fim de se orientar como os links irão se comportar. No exemplo abaixo, foi definido um bloco por ser um menu vertical.

```
a  
{  
  display: block;  
  width: 100px;  
}
```

Acompanhe no código completo:

```
<head>  
<style type="text/css">  
ul  
{  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
}
```

```

a
{
display:block;
width:100px;
background-color:orange;
}
</style>
</head>
<body>
<ul>
<li><a href="#home">Universidade</a></li>
<li><a href="#news">Cursos</a></li>
<li><a href="#contact">Fale conosco</a></li>
<li><a href="#about">Créditos</a></li>
</ul>
</body>

```



Para o caso de Itens alinhados teremos que utilizar um estilo para li.

```

li
{
display:inline;
}

```

Para se deixar os itens lado a lado, pode-se flutuá-los à esquerda utilizando:

```
li  
  
{  
  
float:left;  
  
}
```

14 Na prática

Agora que você já conhece os principais recursos de CSS, que tal juntar tudo isto e mais um pouco em uma página organizando um menu para construir uma página exemplo? É só digitar o código abaixo e acompanhar, no seu navegador, o exercício. Tente alterar os valores das propriedades e veja o resultado.

Vamos começar analisando um exemplo bastante simples e, depois, outro um pouco maior e mais complexo:

<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>>

<http://www.w3.org/1999/xhtml>" xml:lang="en" lang="en">

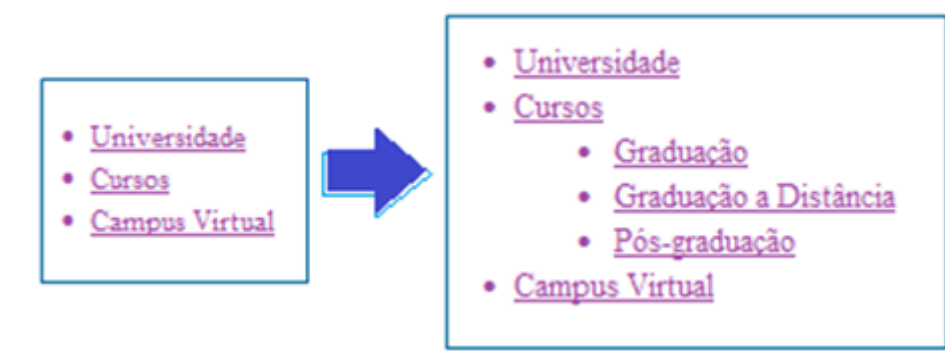
```
<head>  
  
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>  
  
<title>Exemplo</title>  
  
<style type="text/css" media="screen">  
ul li ul {  
display: none;  
}  
ul li:hover ul {  
display: block;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  
<li>  
  
<a href="http://www.universidade.br/menu/" title="Página Principal" class="current">Universidade</a>  
  
</li>  
  
<li>  
  
<a href="http://www.universidade.br/menu/" title="Cursos">Cursos</a>
```

```

<ul>
<li>
<a href="http://www. universidade.br/menu/" title="Graduação">Graduação</a>
</li>
<li>
<a href="http://www. universidade.br/menu/" title="Graduação a Distância">Graduação a Distância</a>
</li>
<li>
<a href="http://www. universidade.br/menu/" title="Pós-Graduação">Pós-Graduação</a>
</li>
</ul>
</li>
<li>
<a href="http://www. universidade.br/menu/" title="Campus Virtual">Campus Virtual</a>
</li>
</ul>
</body>
</html>

```

O resultado é muito interessante. Construímos um menu com sub-menu sensível ao mouse. Quando se passar o mouse sobre o item 'Cursos', ele explodirá os sub-itens. Acompanhe a sequência de imagens.



Viu como é fácil? Agora, analisemos o exemplo mais complexo. Não fique preocupado com o tamanho, pois são mais itens no menu. Lembre-se de revisar a aula e buscar, no site do W3C, sua complementação.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

```

```

<html>

<head>

<title>Menu</title>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>Menu</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<style type="text/css">

/* Aqui cuidamos estilo para o bloco que irá representar o Menu */

#bloco {

padding: 25px 15px 0 15px;

background: #67A897;

}

/* Aqui cuidamos do estilo da lista do Menu */

ul#menu {

list-style-type: none;

width: 100%;

position: relative;

height: 27px;

font-family: "Trebuchet MS", Arial, sans-serif;

font-size: 13px;

font-weight: bold;

margin: 0;

padding: 11px 0 0 0;

}

/* Aqui cuidamos do estilo do item da lista do Menu */

ul#menu li {

display: block;

float: left;

margin: 0 0 0 4px;

height: 27px;

```



```

}

/* Aqui cuidamos do estilo do item da lista no Menu */
ul#menu li.left {
margin: 0;
}

/* Aqui cuidamos do estilo do link do item da lista no Menu */
ul#menu li a {
display: block;
float: left;
color: #fff;
background: #4A6867;
line-height: 27px;
text-decoration: none;
padding: 0 17px 0 18px;
height: 27px;
}

/* Aqui cuidamos do estilo do link do item da lista no Menu */
ul#menu li a.right {
padding-right: 19px;
}

/* Aqui cuidamos do estilo do hover no link do item da lista no Menu */
ul#menu li a:hover {
background: #2E4560;
}

/* Aqui cuidamos do estilo link corrente do item da lista no Menu */
ul#menu li a.current {
color: #2E4560;
background: #fff;
}

/* Aqui cuidamos do estilo do hover no link corrente do item da lista no Menu */
ul#menu li a.current:hover {
color: #2E4560;
}

```

```

background: #fff;

}

/* Aqui cuidamos do estilo da cor link do item da lista no Menu */
ul#menu li a {
color: #80ff00;
}

#

</style>

</head>

<body>

/* Aqui definimos um bloco do menu com a tag div (divisão), separando-o do resto da página. Este recurso é
muito utilizado para substituir o recurso de tabela nos posicionamentos e lay-outs */

<div id="bloco">

<ul id="menu">

<li>

<a href="http://www.universidade.br/menu/" title="Página Principal" class="current">Universidade</a>

</li>

<li>

<a href="http://www.universidade.br/menu/" title="Admissão">Admissão</a>

</li>

<li>

<a href="http://www.universidade.br/menu/" title="Cursos">Cursos</a>

</li>

<li>

<a href="http://www.universidade.br/menu/" title="Unidades">Unidades</a>

</li>

<li>

<a href="http://www.universidade.br/menu/" title="Serviços">Serviços</a>

</li>

<li>

<a href="http://www.universidade.br/menu/" title="Alunos">Alunos</a>

</li>

```

```
<li>

<a href="http://www.universidade.br/menu/" title="Empresa">Empresa</a>

</li>

<li>

<a href="http://www.universidade.br/menu/" title="Campus Virtual">Campus Virtual</a>

</li>

</ul>

</div>

<h1>Novos Cursos em EAD</h1>

<h2>Gestão da Tecnologia da Informação</h2>

O curso habilita o aluno para que atue em organizações de qualquer setor (público ou privado), indústria, comércio e empresas que utilizam as tecnologias da informação e comunicação como diferencial estratégico para suas operações e decisões. Além disso, o gestor da tecnologia da informação é um profissional apto a prestar assessoria e consultoria nas diversas áreas das organizações.<h2>Análise e Desenvolvimento de Sistemas</h2>

O profissional da área de Análise e Desenvolvimento de Sistemas administra e mantém softwares que apoiam o funcionamento de uma organização. Possui conhecimento para a criação de projetos de automação de processos administrativos e para a gestão do desenvolvimento de sistemas. Também aprende as técnicas para documentar, testar e implantar o sistemas de informação como também para usar ferramentas metodológicas de garantia de qualidade em sistemas de informação.<h2> Sistemas de Informação </h2>

O Bacharel em Sistemas de Informação estará apto a atuar com inovação, planejamento e gerenciamento da informação e da infraestrutura de tecnologia da informação alinhados às estratégias organizacionais como também com o desenvolvimento e evolução de sistemas de informação e da infraestrutura de informação e comunicação, ambos aplicados aos processos organizacionais.

</body>

</html>
```

Saiba mais



Quer conhecer as outras propriedades? Ainda com dúvida? Aprenda mais com o W3C em: www.w3.org/TR/CSS2/visuren.html#display-prop.