

HTML 6

Estes exercícios estão baseados no tutorial disponível em www.w3schools.com.

Exercício 1

Digite o texto a seguir com o nome de cidade.html e veja o resultado

```
Existem diversas cidades no estado do Paraná, mas vai-se falar aqui da capital.
<div style="background-color:green;color:white; margin:20px;padding:20px;">
<h2>Curitiba</h2>
<p>Curitiba é a capital do Estado do Paraná. É a cidade mais populosa do estado. Contando com a região metropolitana, conta com 3 milhões de habitantes </p> </div>
Outras cidades importantes são Londrina, Maringá, Ponta Grossa e Paranaguá.
```

Em HTML existem elementos de bloco (*block level*) e elementos em linha (*inline*). Os elementos de bloco normalmente começam e terminam por uma nova linha (*newline*) quando mostrados pelos browsers. Como exemplo: <h1>, <p>, , <table>. Já os elementos em linha são normalmente mostrados sem quebrar a linha. Por exemplo , <td>, <a>, .

O elemento HTML <div> é um elemento de bloco que é usado como container para conter outros elementos HTML. Ele não tem um propósito específico, mas serve para marcar blocos do arquivo que sofrerão algum tratamento especial (como a cidade de Curitiba no exemplo acima). Ele não tem atributos obrigatórios, mas style e class são comuns.

O elemento é um elemento em linha que é usado como um container de texto. Ao contrário de <div> que sempre é formatado com *newlines*, não sofre nenhuma formatação automática. Veja um exemplo

```
<h1>Meu cabeçalho <span style="color:red">
Importante</span> está aqui</h1>
```

Aplicando classes a elementos HTML pode-se definir estilos CSS aplicados a todos os elementos da classe. A regra é

```
Estilos iguais para classes iguais ou
estilos diferentes para classes diferentes.
```

Dentro do estilo (tag <style> dentro da tag <head>) o formato dos seletores de classe é

```
seletor.classe {propriedade: valor;}
```

Se a classe tem nome único ou se aplica a todos, o nome do seletor pode ser omitido. Compare com o exemplo acima

```
<head> <style>
.cidades {background-color:green;color:white; margin:20px;padding:20px;} </style></head>
<body>
Existem diversas cidades no estado do Paraná, mas vai-se falar aqui da capital.
<div class="cidades">
<h2>Curitiba</h2>
<p>Curitiba é a capital do Estado do Paraná. É a cidade mais populosa do estado. Contando com a região metropolitana, conta com 3 milhões de habitantes </p> </div> </body>
```

Agora todas as cidades do texto que estiverem dentro de uma <div> classificada como cidades sofrerão a mesma formatação.

Da mesma maneira, pode-se usar um estilo para . Veja como

```
<head> <style>
span.verdao {color:green;}
</style> </head> <body>
<h1>Meu cabeçalho<span class="verdao">
Importante</span> está aqui</h1> </body>
```

Veja a seguir como pode-se criar um layout multicolumnado usando <div>.

```
<div id="header">
<h1>Galeria de Cidades</h1>
</div>

<div id="nav">
Londres<br>
Paris<br>
Toquio<br>
</div>

<div id="section">
<h1>Londres</h1>
<p> Londres é a capital da Inglaterra. É a cidade mais populosa do país. </p>
<p> Situada ao lado do Rio Tâmis a cidade foi fundada por romanos, com nome de Londinium. </p> </div>

<div id="footer">
Viaje e crie cultura
</div>
```

Agora o CSS

```
<style>
#header {background-color:black; color:white; text-align:center; padding:5px; }
#nav {line-height:30px; background-color:#eeeeee; height:300px; width:100px; float:left; padding:5px;}
#section {width:350px; float:left; padding:10px; }
#footer {background-color:black; color:white; clear:both; text-align:center; padding:5px; }
</style>
```

Um detalhe: O HTML5 prevê tags semânticas para ajudar a construir o layout. São elas: <header>, <nav>, <section>, <article>, <aside>, <footer>, <details> e <summary>. Se quiser usá-las, o conjunto e o CSS ficaria

```
\b<header>
<h1>Galeria de Cidades</h1>
</header>
...
<style>
header {
background-color:black;
color:white;
text-align:center;
padding:5px; }
...
```

O conceito agora é o do RWD (Responsive Web Design). Trata-se de uma técnica para criar layouts que se adaptam adequadamente a diversos tamanhos de tela, já que páginas web hoje podem ser acessadas por PCs, tablets e smartphones.

Existem diversos jeitos de fazer isso. Um, é programar na unha. Veja o arquivo [responsive1.html](#) e modifique o tamanho da tela para ver a resposta que ele dá. Estude os comandos empregados.

Outra maneira é usar um framework de CSS pronto. O mais popular é o [bootstrap](#) (HTML, CSS, JS) para criar web responsivo. Olhe o arquivo [responsive2.html](#) e veja o seu comportamento.

Exercício 2

Crie um arquivo de nome [pracas.html](#) falando sobre 3 praças de Curitiba. Inclua pelo menos uma foto em cada praça. Faça um layout responsivo que se adapte a telas estreitas.

Nosso próximo conceito é o de [Iframe](#). O [Iframe](#) é usado para abrir uma página web dentro de uma página web. A sintaxe é

```
<iframe src="url" width="tamanho" height="tamanho" frameborder="0"></iframe>
```

Os tamanhos podem ser estabelecidos à vontade e o [frameborder](#) diz se há ou não borda no frame. Pode-se ainda estabelecer um nome no [iframe](#) para referenciá-lo em um link como em

```
<iframe width="100%" height="300px" src="www.up.com.br" name="ifra"></iframe>
<p><a href="http://www.up.com.br" target="ifra">UP</a></p>
<p>Quando o target acha o nome de um iframe, o link passa a abrir o site lá</p>
```

Exercício 3

Crie um [iframe](#) dentro de [pracas.html](#) apresentando 3 sites (escolhidos via links diferentes): [up.com.br](#), [curitiba.pr.gov.br](#) e um terceiro site escolhido por você.

A seguir vamos ver o conceito de [XHTML](#). Trata-se de um redesenho do HTML para adaptá-lo ao padrão XML. A sigla significa “Extensible HTML” e ele é mais estrito do que o HTML. A razão para ele é que pela história da Internet, muitos browsers ainda aceitam (e sempre aceitarão) HTML mal-formado. Entretanto, esta miscelânea aponta para problemas devido a proliferação de dispositivos onde o HTML deve ser interpretado. O padrão XML exige documentos bem-formados e portanto XHTML idem. Eis as principais diferenças entre HTML e XHTML.

- o elemento DOCTYPE é obrigatório.
- o atributo xmlns de <html> é obrigatório
- <html>, <head>, <title> e <body> são obrigatórios
- elementos XHTML devem ser apropriadamente aninhados e corretamente encerrados, mesmo os elementos vazios, por exemplo
- elementos e atributos devem ser escritos em minúsculo.
- todos os valores devem estar entre aspas.

Existe um validador de XHTML disponível em <http://validator.w3.org/>. Basta ir lá e digitar o endereço de sua página. O validador dar-lhe-á um parecer sobre a sua página.

Exercício 4

Verifique visualmente se [pracas.html](#) atende aos requisitos de XHTML. Corrija-o se necessário. Depois submeta-o ao validador e capture o resultado dele.

📝 Para você fazer

- Escreva aqui e se não couber continue no verso desta folha um resumo das tags estudadas nesta aula:

- Imprima os arquivos pedidos (cidade, [pracas.html](#)) e a captura da saída do validador grameie-os nesta folha e devolva tudo ao professor.

