

Cascade Style Sheets

Estudo sobre as propriedades do CSS

Estilizando

Arquivo de apoio:
estilo-fontes.html

FONTES!!!

Estilizando fontes

<code>font-size: 50px;</code>	Tamanho da fonte
<code>color: #7733aa;</code>	Cor da fonte
<code>font-style: italic;</code>	Aplica o itálico na fonte
<code>font-weight: bold;</code>	Aplica negrito na fonte.
<code>font-family: serif;</code>	Define a “fonte” (arial, georgia, times, etc)

Estilizando fontes

<code>text-indent: 40px;</code>	Define um espaçamento antes do texto
<code>text-transform: uppercase;</code>	Transforma o texto em caixa-alta. Auxilia no uso de leitores de tela, como ChromeVox ou FoxVox.
<code>line-height: 100px;</code>	Define a altura da linha do texto. Ajuda no alinhamento vertical.
<code>text-align: center;</code>	Alinhamento horizontal. Por padrão, é definido como <i>left</i> , mas pode receber <i>center</i> , <i>right</i> ou <i>justify</i> (para justificar).
<code>text-shadow: 2px 2px 10px green;</code>	Aplica sobre o texto. Parâmetros: posicionamento horizontal e vertical, difusão e cor da sombra.

Algumas fontes podem não ser reconhecidas

Algumas fontes são particulares de uma versão ou família de sistema operacional. Caso o sistema não reconheça a fonte, o browser colocará uma fonte default. Isso pode prejudicar o layout da página.

Para contornar este problema, é importante definir uma lista de fontes (separadas por vírgula), conforme abaixo:

```
font-family: Georgia, 'Times New Roman', Times, serif;
```

Google Fonts

O Google mantém um site de fontes que podem ser aplicadas ao site após sua especificação no *<head>*.

Para utilizá-las, entre em <https://fonts.google.com/>, clique no botão de adicionar (+) da fonte escolhida e veja a tag *<link>* e o *font-family* que devem ser utilizados para configurar sua página.

Estilizando Tags

Arquivos de apoio:

estilo-div1.html

estilo-div2.html

estilo-div3.html

estilo-div4.html

estilo-div5.html

estilo-div6.html

estilo-div7.html

exe-p.html

<div>

<h1>, <h2>, ...

<p>

<a>

Vamos começar estilizando uma <div> (*estilo-div1.html*)

<code>width: 400px;</code> <code>height: 200px;</code>	Definem o comprimento e a altura da DIV. Estão em pixels, mas existem outras medidas, por exemplo, % (adiante veremos outras).
<code>background-color: #434366;</code>	Define a cor de fundo da DIV.
<code>border: 3px solid #996622;</code> <code>border: 4px dotted orange;</code> <code>border: 1px dashed rgb(40,70,200);</code>	Define a borda da DIV. Este comando recebe a espessura da borda, o tipo da linha e a cor. Esses dados poderiam ser definidos separadamente pelos comandos: <code>border-width: 3px;</code> <code>border-style: solid;</code> <code>border-color: #996622;</code>

A <div> ficou legal, mas podemos melhorar (*estilo-div2.html*)

Note que a <div> ficou posicionada no canto superior esquerdo da página e que o texto interno ficou muito próximo das bordas. E agora?



O Box Model do HTML/CSS

https://www.w3schools.com/css/css_boxmodel.asp



Estilizando a <div> (*estilo-div2.html*)

<pre>padding: 50px; padding: 10px 20px; padding: 5px 5px 5px 5px; padding-top: 10px; padding-right: 10px; padding-bottom: 10px; padding-left: 10px;</pre>	<p>Define o padding (espaçamento entre o conteúdo e a borda do elemento).</p> <p>Se forem informados 2 parâmetros, os paddings serão aplicados respectivamente às partes de cima e de baixo, e à direita e esquerda.</p> <p>Se forem informados 4 parâmetros, a ordem será: cima, direita, baixo e esquerda.</p> <p>Existem também atributos particulares para cada caso.</p>
<pre>margin: 30px;</pre>	<p>Define a margem do elemento. As definições de 2 e 4 parâmetros também são possíveis (semelhante à definição do padding). Também há atributos particulares para cada posição (cima, direita, baixo e esquerda).</p>

Estilizando a <div> (*estilo-div3.html*)

https://www.w3schools.com/cssref/css3_pr_border-radius.asp

`border-radius: 25px;`

`border-radius: 10px 50px;`

`border-radius: 5px 15px 25px 40px;`

Define a curvatura do canto da borda. O valor passado por parâmetro define o **raio** do círculo formador da curvatura.



Se forem informados 2 valores, o primeiro se aplica aos cantos superior esquerdo e inferior direito e o segundo aos cantos superior direito e inferior esquerdo.

Se forem informados 4 valores, a ordem será canto superior esquerdo, superior direito, inferior direito e inferior esquerdo.

Estilizando a <div> (*estilo-div3.html*)

<code>box-shadow: 20px 20px 10px #ff0000;</code>	Este atributo possui comportamento semelhante ao text-shadow, definindo o posicionamento (2 primeiros parâmetros), a difusão e a cor da sombra.
<code>opacity: .2;</code>	Define uma transparência para o elemento, sendo 0 totalmente transparente e 1 totalmente visível. Note que o conteúdo do elemento também ficará transparente.
<code>background-color: rgba(0, 0, 0, .2);</code>	Uma alternativa para não tornar o conteúdo transparente é utilizar o valor RGBA, onde A representa o <i>alpha</i> , que vai de 0 (totalmente transparente) e 1 (totalmente visível).

Posicionando a <div> (*estilo-div4.html*)

`float: right;`

`float: left;`

`float: none;`

O atributo *float* altera o posicionamento padrão do elemento (que normalmente é um abaixo do outro) e o desloca de acordo com o valor informado (direita ou esquerda), tornando-o **flutuante** na tela.

O valor *none* anula o comportamento do *float*.

Cuidado: o próximo elemento da página subirá e ocupará o lugar do elemento flutuante.



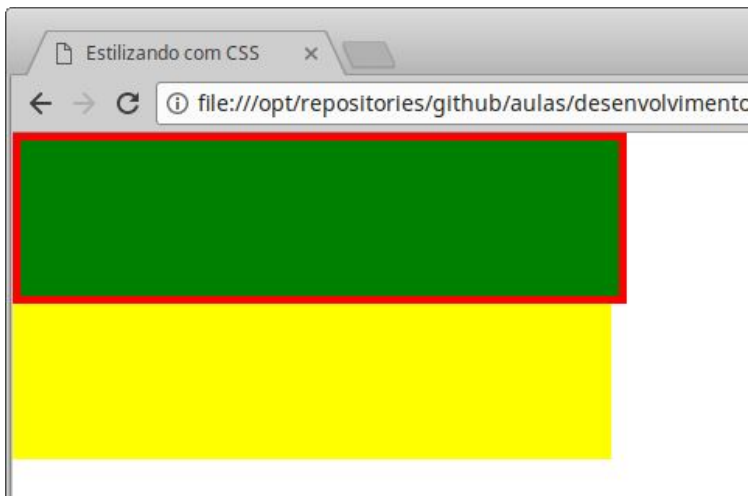
Mais sobre Box Model (*estilo-div5.html*)

Vamos agora definir 2 DIVs, ambos flutuantes do lado esquerdo da tela. Ao invés de terem comprimento em pixels, definiremos comprimento em %.



O que acontece se definirmos valores para *border* ou *padding*?

Mais sobre Box Model (*estilo-div5.html*)



Isso acontece porque os valores do *border* e *padding* são considerados integrantes do comprimento e altura total do elemento.

Como resolver? Com o *box-sizing*.

box-sizing: border-box;

Através desse atributo, com o valor *border-box*, os valores do *border* e *padding* são automaticamente incluídos no comprimento e altura total do componente. O valor do *margin* **não** é considerado!

Estilizando um elemento (*estilo-div6.html*)

Vamos agora estilizar um elemento . Ele pode assumir todos os mesmos atributos apresentados para o elemento <div>.

```
span {  
  width: 200px;  
  height: 200px;  
  background-color: #55ff11;  
  margin-left: 30px;  
  margin-top: 100px;  
}
```

Após estilizar, temos... nada?

Por que isso está acontecendo?

Elementos *inline* e *block*

A primeira coisa que temos que ter em mente na hora de estilizar os elementos é que alguns possuem comportamentos semelhantes a textos (*inline*) e outros são apresentados como blocos (*block*).

As tags *inline* assumem o comprimento e largura de seus conteúdos. Se não houver conteúdo, eles terão 0 pixels. Na prática isso significa que as propriedades *width*, *height*, e distanciamento *margin-top* e *margin-bottom* não são aplicados.

Além desse comportamento, uma tag *inline* não “quebra a linha” para um próximo parágrafo - enquanto as tags *block* vão interromper a sequência textual (ou estrutural) para serem exibidas abaixo.

Tags inline

Exemplos de tags inline: ``, `<big>`, `<i>`, `<small>`, ``, `<sub>`, `<sup>`, `<button>`, `<input>`, `<label>`, `<select>`, `<textarea>`, entre outros.

Lembre-se: as tags *inline* não aceitam estilos de posicionamento, comprimento, altura, etc. As tags “encolhem” para caber apenas o conteúdo. Exemplo:

Texto com `link` sem quebrar linha.

Texto com link sem quebrar linha.

Tags block

Exemplos de tags block: <article>, <aside>, <blockquote>, <canvas>, <div>, <fieldset>, <footer>, <form>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <header>, <hr>, , , , <main>, <nav>, <noscript>, <p>, <section>, <table>, entre outras.

Essas tags aceitam estilos de posicionamento, comprimento e altura.

Texto com <div>um elemento block</div>

Texto com
um elemento block

Ajustando comportamento *inline* e *block* (*estilo-div6.html*)

É possível alterar o comportamento padrão dessas tags transformando-as de *inline* para *block* ou vice-versa. Isso é pode ser realizado através da propriedade *display*.

display: *block*;

display: *inline*;

É possível também configurar um elemento *inline* para aceitar estilos de posicionamento ou tamanho através do valor *inline-block*.

display: *inline-block*;

Nenhuma exibição (*estilo-div7.html*)

Uma outra possibilidade interessante para o atributo *display* é o valor *none*, que simplesmente **não** exibe o componente na tela.

display: none;

Isso pode parecer inútil, porém há casos reais em que é necessário deixar o componente escondido até que determinado evento seja disparado pelo navegador (seja pelo CSS ou JavaScript).

E as outras tags?

Todos esses atributos apresentados podem ser aplicados às tags `<p>`, `<h1>`, `<h2>`, ..., `<h6>`, `<a>`, ``, ``, entre outras. Essas propriedades também são válidas para `<table>`, mas estudaremos outros detalhes particulares às tabelas adiante.

Exercício em sala (*exe-p.html*)

Crie um parágrafo com a tag `<p>` que fique parecido com a imagem a seguir. **Dica:** use a abuse das Google Fonts e do Materialize Colors.

Obs: o texto não precisa ser o mesmo.

Cascading Style Sheets (CSS) é um simples mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) a um documento web. Em vez de colocar a formatação dentro do documento, o CSS cria um link para uma página que contém os estilos. Quando quiser alterar a aparência do portal, basta modificar apenas um arquivo

Obrigado!