

# PROGRAMAÇÃO BÁSICA

NATANAEL ANTONIOLI

# 1 FUNDAMENTOS DO HTML



# Programação Básica - Fundamentos do HTML

Por: Fábrica de Noobs

## *Índice*

- 1) Prefácio
- 2) Introdução ao HTML
- 3) Programas Recomendados
- 4) Funcionamento do HTML
- 5) Estrutura básica da página em HTML
- 6) Salvamento de páginas
- 7) Cabeçalho
  - a. Tag <meta>
    - i. Atributo charset
    - ii. Atributos description e name
  - b. Tag <title>
- 8) Formatação de Texto I
  - a. Tags <h1> até <h6>
  - b. Tag <p>
  - c. Tag <b>
  - d. Tag <i>
  - e. Tag <u>
  - f. Tag <s>
  - g. Tag <sub>
  - h. Tag <sup>
  - i. Tag <hr>
- 9) Inserção de Imagens
  - a. Tag <img>
    - i. Atributo src
    - ii. Atributo height e width
    - iii. Atributo align na tag <p>.
- 10) Formatação de Texto II
  - a. Subclasse style
    - i. Atributo font-size
    - ii. Atributo font-family
    - iii. Atributo color
    - iv. Atributo font-align
  - b. Tag <font>
    - i. Atributo size
    - ii. Atributo face
    - iii. Atributo color
- 11) Links
  - a. Tag <a>
    - i. Atributo href

- ii. Atributo target

## 12) Listas

- a. Tag <li>
- b. Tag <ul>
- c. Tag <ol>

## 13) Tabelas

- a. Tag <table>
  - i. Atributo border
  - ii. Atributo bordercolor
- b. Tag <tr>
- c. Tag <td>
- d. Tag <thead>
- e. Tag <tbody>
- f. Tag <th>
  - i. Atributo colspan
  - ii. Atributo bgcolor

## 14) Divisões na Página

- a. Tag <div>
- b. Tag <span>

## 15) Conclusão

## 1) Prefácio

Se você está lendo isso, então é porque meu projeto deu certo e decidi publicar o curso. Antes de irmos propriamente ao conteúdo, vou falar um pouco sobre o objetivo dele.

Todo o conteúdo deste curso é voltado ao público iniciante, para aquele que está interessado no HTML não para sair trabalhando com ele amanhã, mas para ter uma noção básica de como a coisa funciona, e usá-la no futuro ou em alguma outra área – em determinados conteúdos de hacking, pode ser importante, por exemplo – além disso, o curso é voltado para o contexto atual, onde já temos programas que automatizam boa parte do processo, tornando o ato de programar algo menos monótono.

Já vi alguns cursos de HTML (pagos, inclusive) que obrigavam o aluno a programar no bloco de notas, e repetir a mesma coisa centenas de vezes – isso quando não pediam para copiar códigos – é exatamente o oposto que você encontrará aqui.

Pretendo futuramente gravar todo o conteúdo desse curso no canal Fábrica de Noobs, mas por enquanto será apenas por texto.

Recomendo que não fique preso somente ao conteúdo dele, vá por conta própria, faça experiências, procure por outros exercícios.

Pois bem, é isso. Agradeço por ter se interessado pelo meu curso, não sou muito experiente em dar esse tipo de aulas, então sinta-se à vontade para qualquer tipo de sugestão e crítica.

Bons estudos!

Meus agradecimentos especiais à

***Luan Vinícius***

***Carlene Gomes***

***Joao Paulo Assis***

***@BR\_Kizuna***

Que se disponibilizaram a revisar esse curso.

## 2) Introdução ao HTML

Ao navegar na internet, entramos em contato com HTML o tempo todo.. O HTML é basicamente o “esqueleto” de uma página da web, composto de marcações, que orientam o navegador a transformar uma série de códigos em um site amigável, que seja bonito para o usuário.

Não é exatamente uma linguagem de programação, mas sim de marcação – uma vez que não possui recursos como variáveis, e o usuário apenas sinaliza para o navegador como cada elemento da página deve ser interpretado. Porém, é uma linguagem interessante para quem deseja ter uma noção de como páginas da web são construídas, solucionar desafios ou, caso trabalhe com hacking, criar e identificar phishing scams.

Atualmente, a maioria dos sites utilizam CSS e complementações para o HTML (a não ser que se trate do site da prefeitura da sua cidade, provavelmente), mas aqui trabalharemos o básico da programação – que ainda é muito útil – e em outro curso, entraremos em CSS.

O CSS é o que dá um visual melhor ao HTML. Pense no HTML como o esqueleto, e o CSS como a maquiagem.

## 3) Programas Recomendados

Para criar códigos em HTML é necessário qualquer editor de texto comum – como o bloco de notas. Mas é algo realmente monótono e cansativo de usar, então vou apresentar alguns programas que podem ser úteis para você:

- **Notepad++**: é um editor de texto um pouco mais bem trabalhado, com cores e tabulações melhores. Recomendo usá-lo no início, para entender o básico que é normalmente automatizado em outros programas.
- **Adobe Dreamweaver**: indico para quem já conhece os fundamentos – ou seja, para você daqui a alguns capítulos – e quer agilizar a programação dos códigos. Sua grande vantagem é que não é necessário escrever os elementos por inteiro, eles já aparecem intuitivamente. Isso significa maior facilidade na programação, mas requer um pouco mais de prática. O programa é pago, porém você pode baixá-lo pirata aqui <http://fabricadenoobs.wix.com/home#!Adobe-Dreamweaver-CSS/c7i/55972ff50cf2604da3499ddd>, ou caso tenha baixado o curso completo, ele vem como material complementar.

- **Brackets:** semelhante ao Dreamweaver, em versão grátis. Não conta com tantos recursos e possui uma mecânica um tanto diferente, mas não deixa de ser interessante testá-lo.

#### 4) Funcionamento do HTML

O HTML trabalha na base de tags (ou etiquetas), onde marcamos um elemento para mostrar o que a página deve exibir. Esse sistema funciona assim:

```
<tag> Elemento qualquer </tag>
```

Como pode-se perceber no exemplo, uma vez que a tag é aberta, deve-se fechá-la (salvo algumas exceções).

Uma tag HTML pode conter outra tag, e assim por diante:

```
<tag>  
    <outra-tag> Texto </outra-tag>  
</tag>
```

Além disso, uma tag pode conter determinados valores – os chamados atributos. No exemplo abaixo, damos ao título (<h1>) o atributo estilo (*style*), e em seguida, o igualamos a sua cor (*color*).

```
<h1 style="color:red"> Título em Vermelho </h1>
```

Não se preocupe por não entender a função de cada elemento acima. Isso será explicado mais adiante. No momento, o objetivo é apenas compreender a sintaxe do HTML.

Se necessário, podemos também colocar comentários em meio ao nosso código. Estes serão desconsiderados no momento de compilar o script, uma vez que servirão apenas para orientar o programador. Fazemos isso utilizando <!-- -->, como no exemplo abaixo:

```
<h1> qualquer coisa <h1>  
<!-- o que temos acima é um parágrafo sobre qualquer coisa -->
```

Observe que o texto inserido entre as tags de comentário não fará parte do código final.

Também é importante notar alguns fatores ao se escrever em HTML:

- A linguagem não é case-sensitive. Isso significa que não importa se algo é escrito em minúsculas ou maiúsculas, terá sempre o mesmo valor. Porém, para melhor organização, recomendo usar apenas uma formatação em todo o código.

- Procure manter seu documento indentado (isto é, usando tabulações) para facilitar a manutenção.

## 5) Estrutura básica da página em HTML

A primeira coisa que precisamos fazer ao construir um código HTML é informar o navegador de que o mesmo deve interpretá-lo como um código HTML. Fazemos isso utilizando inicialmente a tag `<!doctype: html>`, depois `<html>` e `</html>`, para marcar o código de ponta-a-ponta.

Logo em seguida, temos o cabeçalho, ou topo da página. Nele, colocaremos informações para serem interpretadas pelo navegador, incluindo sistema de caracteres, título da página, código css, palavras-chave, entre outros.

Usamos as tags `<head>` e `</head>` para marcar o início e o fim do cabeçalho do documento.

Por fim, usamos as tags `<body>` e `</body>` para delimitar o corpo da página. É nele que inseriremos todos os elementos presentes na página da web.

Portanto, todo documento HTML deve apresentar, primordialmente, o seguinte formato:

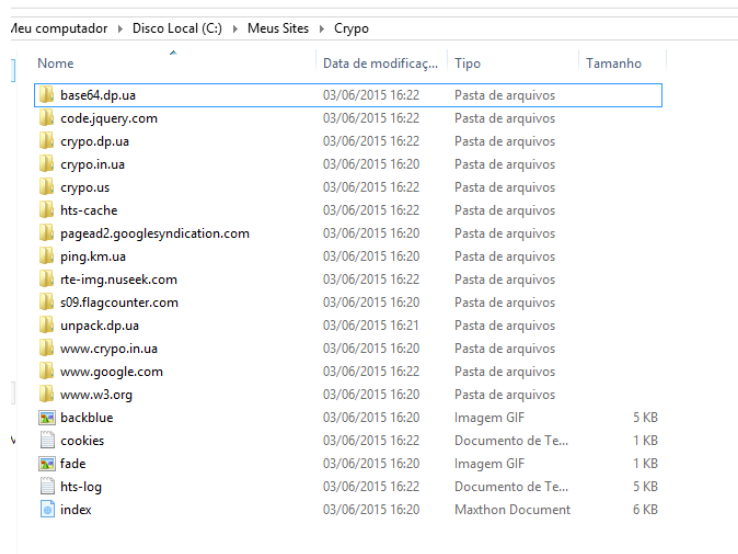
```
<!doctype: html>  
<html>  
<head>  
<meta charset="utf-8">  
<title> Título do Documento </title>  
</head>  
<body>  
</body>  
</html>
```

De início, atente-se apenas a parte colorida. As demais informações serão explicadas posteriormente.

## 6) Salvamento de Páginas

Uma vez terminado, o documento deve ser salvo. Normalmente, criamos uma pasta para o projeto, dentro da qual inserimos todos os seus arquivos.

Por convenção, a home page é chamada de `index`. As demais pastas e conteúdos podem ser nomeados como preferir, ao passo que todas as páginas em HTML devem conter a extensão `.html`. Veja o exemplo:



Nome	Data de modificaç...	Tipo	Tamanho
base64.dp.ua	03/06/2015 16:22	Pasta de arquivos	
code.jquery.com	03/06/2015 16:22	Pasta de arquivos	
crypto.dp.ua	03/06/2015 16:22	Pasta de arquivos	
crypto.in.ua	03/06/2015 16:20	Pasta de arquivos	
crypto.us	03/06/2015 16:22	Pasta de arquivos	
hts-cache	03/06/2015 16:22	Pasta de arquivos	
pagead2.google syndication.com	03/06/2015 16:20	Pasta de arquivos	
ping.km.ua	03/06/2015 16:20	Pasta de arquivos	
rte-img.nuseek.com	03/06/2015 16:22	Pasta de arquivos	
s09.flagcounter.com	03/06/2015 16:20	Pasta de arquivos	
unpack.dp.ua	03/06/2015 16:21	Pasta de arquivos	
www.crypto.in.ua	03/06/2015 16:20	Pasta de arquivos	
www.google.com	03/06/2015 16:22	Pasta de arquivos	
www.w3.org	03/06/2015 16:20	Pasta de arquivos	
backblue	03/06/2015 16:20	Imagem GIF	5 KB
cookies	03/06/2015 16:22	Documento de Te...	1 KB
fade	03/06/2015 16:20	Imagem GIF	1 KB
hts-log	03/06/2015 16:22	Documento de Te...	5 KB
index	03/06/2015 16:20	Maxthon Document	6 KB

**Exercício 1 – Criando seu primeiro esqueleto HTML:** utilizando os conhecimentos adquiridos, crie uma página contendo, além das tags fundamentais, os seguintes códigos:

**Cabeçalho:**

```
<meta charset="utf-8">
```

```
<title> Página em HTML </title>
```

**Corpo:**

```
<h1> Primeiro texto </h1>
```

```
<h2> Segundo Texto </h2>
```

**Importante:** caso pretenda experimentar outros códigos, talvez encontre alguns erros envolvendo acentuação e caracteres especiais. Não se preocupe, isso será explicado a seguir.

## 7) Cabeçalho



Após a identificação do documento, o próximo passo é preencher o cabeçalho. Ele deve conter informações adicionais a serem interpretadas pelo browser.

Primeiro, informamos ao navegador qual tipo de codificação ele usará em seus caracteres. Sem esse procedimento, ao compilar o código, a palavra “Canção” poderia sair como “CanÃ§Ã£o”, por exemplo.

A codificação mais usada atualmente é a UTF-8, apesar de existirem exceções. Para informar ao editor que o código é feito em determinada função, usamos

```
<meta charset="nome_da_codificação">
```

Nisso, podemos ainda apresentar o uso de atributos, onde `<meta>` é a tag principal, e `charset`, seu atributo. Atributos terão inúmeras finalidades no decorrer do curso.

A parte em azul representa a tag `<meta>`, que também serve para outras determinações.

A parte em vermelho é o atributo `charset`, que indica a codificação escolhida.

A parte grifada em verde, entre os sinais de aspas e igual, é o nome da codificação a ser utilizada.

Logo, caso queiramos deixar a página codificada em UTF-8, devemos construir a tag da seguinte forma:

```
<meta charset="utf-8">
```

Alguns editores HTML, como o [Adobe Dreamweaver](#) já possuem a configuração em UTF-8 como padrão. Para alterá-la, vá em [Modificar](#) → [Propriedades da Página](#) → [Título/Codificação](#).

Porém, o mesmo não acontece no [Notepad++](#) e com o [Bloco de Notas](#). Nesses editores, além da tag, é preciso uma configuração adicional. Veja abaixo o procedimento para cada um deles:

No [Notepad++](#), deve-se marcar a opção [HTML](#) no menu [Linguagem](#) e marcar a [Codificação em UTF-8](#) no menu [Formatar](#).

Já no [Bloco de Notas](#), ao salvar o arquivo, fique atento a caixa [Codificação](#) que aparece no canto inferior da tela ao clicar em [Salvar](#).

Até o momento, a aparência do cabeçalho deve ficar como:

```
<head>
```

```
<meta charset="utf-8">
```

```
</head>
```

Agora, vamos ao próximo passo. Para facilitar a localização da página por mecanismos de busca (como o Google), é conveniente usar meta-tags – que são como variáveis, onde atribuímos nomes e valores. Mostrarei como inserir uma breve descrição do site, e também palavras-chave.

Para inserir uma descrição, devemos usar:

```
<meta name="description" content="Descrição da sua página">
```

Não é recomendado que o texto da descrição tenha mais de 160 caracteres.

No código acima, utilizamos a tag `<meta>` para criar uma nova meta-tag, a qual nomeamos `description` através do atributo `name`, e inserimos seu conteúdo na área grifada em verde, através do atributo `content`.

Para inserir palavras-chave, devemos usar:

```
<meta name="keywords" content="palavra1, palavra2, palavra3">
```

Mais uma vez, fazemos o mesmo procedimento do item anterior, substituindo apenas o nome da meta-tag e seu conteúdo. A única diferença é que devemos separar as palavras-chave por vírgulas.

Observe que `description` e `keywords` são usados como uma convenção para facilitar a indexação das páginas.

Por fim, falta darmos um nome para a página, o qual será exibido na guia superior do navegador. Esse nome é dado usando as tags `<title>` `</title>`, colocando entre elas o nome desejado:

```
<title> Nome da Página </title>
```

Resumindo, supondo que criássemos uma página para o meu canal, ela usaria esse cabeçalho:

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="description" content="Página do Fábrica de Noobs">
```

```
<meta name="keywords" content="Deep Web, gameplays, programação, tutoriais, hacking">
```

```
<title>Fábrica de Noobs</title>
```

```
</head>
```

Ao contrário de alguns cursos, aqui, apenas o título e a codificação será inserida no início da página, uma vez que os demais não tem muita finalidade

em um documento de testes. A partir do próximo capítulo, as tags de identificação básicas já estarão inseridas.

**Exercício 2 – Criando seu primeiro cabeçalho:** crie uma página de sua escolha, contendo

- *Codificação em UTF-8*
- *Título*
- *Descrição*
- *Palavras-chave*

## 8) Formatações de Texto I

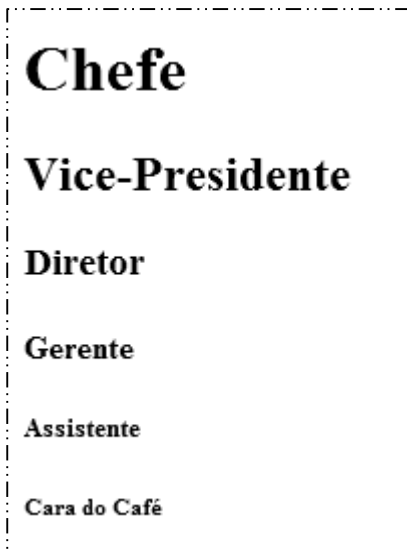
Uma vez visto como configurar o cabeçalho de seu documento HTML, vamos para o corpo. Antes de mais nada, essa área é delimitada pelas tags `<body>` e `</body>`.

Começaremos com as formatações do texto. A primeira coisa que todo documento costuma ter, é um título. Este, em HTML, possui infinitas personalizações.

De início, o HTML nos fornece 6 opções prontas para os títulos, representadas com as tags de `<h1> </h1>` até `<h6> </h6>`. Observe o código abaixo:

```
<body>
<h1> Chefe </h1>
<h2> Vice-Presidente </h2>
<h3> Diretor </h3>
<h4> Gerente </h4>
<h5> Assistente </h5>
<h6> Cara do Café </h6>
</body>
```

Como deve ter imaginado, o texto é maior em `<h1>`, e vai reduzindo gradativamente, até o `<h6>`. Veja sua compilação:

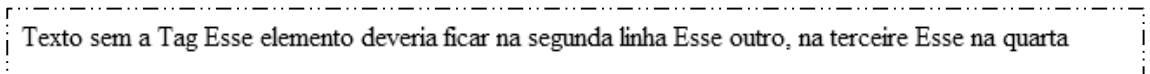


Saindo agora dos títulos e seguindo para o texto em si, outra tag bastante utilizada é a `<p> </p>`, usada para caracterizar um novo parágrafo.

Ela existe porque, se simplesmente colocarmos o texto solto dentro do código HTML (mesmo pulando as linhas com “enter”), o navegador o colocaria todo na mesma linha. Veja no código abaixo:

```
<body>  
Texto sem a Tag  
Esse elemento deveria ficar na segunda linha  
Esse outro, na terceira  
Esse na quarta  
</body>
```

Ao ser compilado, obteríamos isso:



Porém, utilizando a tag `<p>`, resolvemos o problema:

```
<body>  
<p> Texto com a Tag </p>  
<p> Esse elemento deveria ficar na segunda linha </p>  
<p> Esse outro, na terceira </p>  
<p> Esse na quarta </p>  
</body>
```

Tendo, portanto:

Texto com a Tag  
Esse elemento deveria ficar na segunda linha  
Esse outro, na terceira  
Esse na quarta

Observe que a quebra de linha natural (isto é, a que ocorre quando acaba o espaço da página), será feita normalmente pelo navegador, independente do uso dessas tags.

Caso queiramos apenas descer uma linha, sem necessariamente representar um novo parágrafo, podemos utilizar a tag `<br>`. Ela não admite fechamento, nem conteúdo. Coloque-a apenas em meio ao texto para sinalizar que deve-se descer uma linha.

Recomendo usá-la apenas em casos especiais, como para escrever versos de um poema, por exemplo. Nos demais casos, prefira por usar apenas a tag `<p>`.

Também podemos adicionar outras formatações de texto:

- **Negrito** com as tags `<b>` e `</b>`
- *Itálico* com as tags `<i>` e `</i>`
- Sublinhado com as tags `<u>` e `</u>`
- ~~Tachado~~ com as tags `<s>` e `</s>`
- Subscrito com as tags `<sub>` e `</sub>`
- Sobrescrito com as tags `<sup>` e `</sup>`

Essas tags podem ser usadas em qualquer texto, independente de ele já estar ou não dentro de outra tag.

Também podemos criar uma linha horizontal que se estende por toda a página com a tag `<hr>`.

**Exercício 3 – Formatações de texto:** copie e cole o trecho abaixo no corpo da página (isto é, entre as determinadas tags) em um editor HTML de sua preferência.

### Sobre o Kali Linux

**Kali Linux** é uma avançada distribuição Linux especializada em Testes de Intrusão e Auditoria de Segurança.

### Funcionalidades do Kali Linux

Kali é uma reconstrução completa do **BackTrack Linux**, que adere totalmente aos padrões de desenvolvimento do Debian. Uma infraestrutura completamente nova foi montada, todas as ferramentas foram revistas e empacotadas, e nós utilizamos Git como nosso Sistema de Controle de Versões.

Agora, utilizando as tags estudadas, faça com que o texto tenha as seguintes características:

- **Funcionalidades do Kali Linux** seja um subtítulo em <h2>.
- **Sobre o Kali Linux** seja um título em <h1>.
- Além do título e subtítulo, o texto deve ter outros 2 parágrafos.
- Após o primeiro parágrafo, deve haver uma linha divisória.
- **Kali Linux** deve estar em negrito.
- **BackTrackLinux** deve estar em itálico.
- Infraestrutura completamente nova deve estar sublinhada.

## 9) Inserção de Imagens

Uma vez visto o básico sobre textos, vamos falar de imagens.

A primeira coisa que devemos ter em mente é que uma imagem, para ser inserida no corpo, pode vir de dois lugares: de uma página da web, ou do diretório local.

No primeiro caso, devemos apenas usar seu URL, sem tê-la na pasta do projeto. Porém, esse método não é recomendado, uma vez que a imagem é hospedada por terceiros e pode sair do ar a qualquer momento.

O melhor é colocar a imagem na pasta do projeto, e utilizar seu diretório como fonte.

De início, para informar ao navegador que temos uma imagem, devemos usar a tag <img>. Observe que ela não requer fechamento.

Em seguida, precisamos dar alguns atributos a ela. O primeiro (e obrigatório) é o `src`. Ele indica a fonte, ou *source* de onde a imagem foi retirada.

Como todos os outros atributos, o inserimos com um espaço, e depois igualamos usando igual e aspas. A sintaxe fica assim:

```

```

Caso decida por inserir a imagem a partir da URL, o código ficaria:

```

```

No segundo caso, ficaria

```

```

A partir de agora, todos os atributos listados são opcionais.

Podemos também especificar uma borda para a imagem, usando o atributo `border` e o igualando para algum valor em pixels. Por exemplo:

```

```

Recomendo que faça alguns exemplos em algum editor para verificar as diversas espessuras de borda.

Outro atributo importante são os `height` e `width`. (altura e comprimento, respectivamente). Se eles não forem especificados, a imagem será mantida no tamanho original. Porém, ao definir ambos os atributos, a imagem é achatada ou alongada (nas direções horizontal e vertical) conforme o valor em pixels especificado. Caso apenas um dos atributos for determinado, a imagem será alterada com base na mesma proporção.

Por exemplo:

```

```

No caso acima, a imagem será redimensionada para entrar em acordo com tais atributos, o que poderia deixá-la fora de proporção.

Já aqui,

```

```

Apenas a altura da imagem será alterada para 500 pixels. O comprimento será alterado também, mas dentro da mesma escala.

Caso queiramos alinhar uma imagem dentro de um documento, devemos colocá-la dentro de tags `<p>` `</p>`, e usar o atributo `align` (com os mesmos valores listados acima) em `<p>`. Por exemplo:

```
<p align="center">  </p>
```

Também existem inúmeros atributos em <p> que podem ser explorados, ficando a seu critério.

**Exercício 4 – Testes com imagens:** utilizando a imagem disponível na pasta do exercício (ou qualquer outra – da internet ou do seu computador), faça 3 códigos, dentro dos quais:

1. A imagem possua o tamanho original, com borda 4.
2. A altura da imagem seja alterada para 300 e o comprimento para 700.
3. A imagem tenha 300 de altura, e o comprimento seja ajustado proporcionalmente.

*Não se esqueça de colocar cada imagem dentro de um parágrafo para que cada uma fique em sua linha.*

## 10) Formatação de Texto II

Além das marcações de estilo, as quais fazemos com o uso de tags (negrito, itálico, etc), temos ainda os atributos. Como vimos no capítulo anterior utilizando imagens, também podemos definir atributos como fonte e cor para os textos.

Antes de prosseguir, considere que utilizando CSS ou divisões, podemos fazer o mesmo processo de forma mais rápida.

Observe o código abaixo, e sua compilação:

```
<body>  
  
<h1 style="text-align:center"> Testando Atributos em HTML </h1>  
<p style="font-family:Impact"> Esse texto está em Impact </p>  
  
<p style="color:green"> Esse em verde </p>  
  
<p style="font-size:50px"> Esse em tamanho 50 </p>  
  
</body>
```



## Testando Atributos em HTML

Esse texto está em Impact

Esse em verde

Esse em tamanho 50

Veja que utilizamos 4 cores diferentes para a sintaxe do código. Vamos analisar cada uma delas.

```
<p style="color:green"> Esse em verde </p>
```

Temos uma nova categoria, representada em verde: uma subclasse. No caso, a subclasse `style`. Subclasses não são tão comuns em HTML. Nesse caso, ela é usada pois temos várias possibilidades de atributos (como `color`, `font-size`, `font-family`) para o mesmo contexto.

Assim, quando desejamos fazer tais alterações em um texto, devemos utilizar a subclasse `style`, igualar, abrir aspas, e, dentro delas, informar seus atributos utilizando dois pontos para determinar o valor. Caso tenhamos mais de um atributo, os separamos por ponto e vírgula. Assim, a sintaxe para usar o `style` é:

```
<tag style="atributo_1:valor_1; atributo_2:valor_2"> </tag>
```

Onde a parte grifada em azul representa a paragrafação ou forma do texto. Por exemplo: `<h1>`, `<h3>`, `<p>`.

A parte em verde é a subclasse `style`.

A parte grifada em amarelo representa os atributos. Por exemplo: `color`, `font-size`, `font-family`, `text-align`.

E, por fim, a parte grifada em verde corresponde aos valores referentes a cada atributo. Por exemplo: `blue`, `30px`, `Arial`, `center`.

Vejam agora como usar cada um deles. Recomendo colocar os códigos de exemplo no editor, e fazer alguns testes.

- **font-size** – Tamanho

Usamos `font-size` para determinar um tamanho ao texto que se segue. O valor é dado pelo número em pixels, seguido por `px`. Veja os exemplos:

```
<h1 style="font-size:50px"> Texto </h1>
```

```
<p style="font-size:12px"> Texto </p>
```

- **font-family** – Tipo da fonte

Para uma fonte ser exibida no navegador, precisamos primeiro que ela esteja instalada no computador do usuário. Assim, usamos `font-family` para determinar uma ou mais fontes que aparecerão dentro daquela tag, sendo que a primeira fonte listada será a “primeira opção”.

Vejamos como fica a sintaxe no exemplo:

```
<p style="font-family: Cambria"> Texto </p>
```

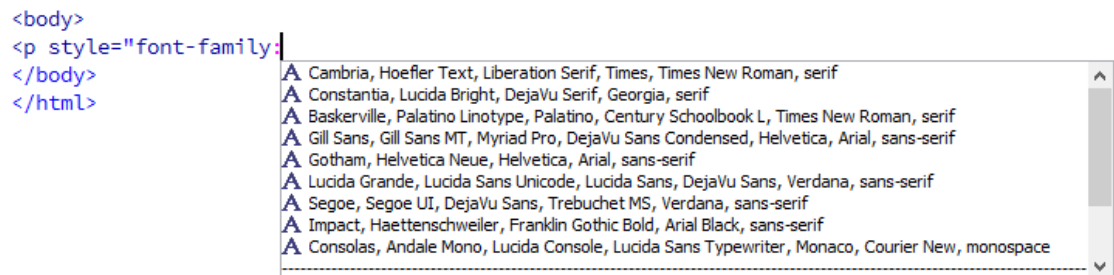
Nesse exemplo, definimos que o parágrafo deve ser escrito na fonte Cambria.

Podemos escolher também várias fontes para o navegador utilizar caso não encontre a anterior, separando-as com vírgula.

```
<p style="font-family: Cambria, 'Hoefler Text', 'Liberation Serif', Times, 'Times New Roman', serif">
```

Porém, tal processo não costuma ser necessário, já que os computadores atuais possuem a maioria dessas fontes por padrão.

Programas como o Adobe Dreamweaver até agrupam as fontes por família para facilitar a escolha:



- **font-color** – Cor da fonte

Também podemos alterar a cor da fonte, utilizando o atributo `color`. Sua sintaxe é a mesma das anteriores, só que possui uma forma um pouco diferente de valores:

```
<p style="color: #3F9"> Verde meio azul, sei lá que cor é essa </p>
```

```
<p style="color: red"> Vermelho </p>
```

Podemos indicar a cor de duas formas: em hexadecimal, ou em texto. É evidente que temos uma gama bem maior de opções ao informar as cores em base hexadecimal.

Para tanto, podemos usar as tabelas de cores, que você encontra com algumas buscas no Google, ou que até já vem nos programas, como é o caso do Adobe Dreamweaver. Nesse site <http://html-color-codes.info/> podemos encontrar um formador de cores.

- **text-align** – Alinhamento do texto

Utilizando o atributo text-align, podemos determinar o alinhamento do texto em relação a página. Observe o exemplo:

```
<h1 style="text-align:center"> Texto Centralizado </h1>
```

```
<h1 style="text-align:left"> Texto na Esquerda </h1>
```

Dispensa dizer que informamos o tipo de alinhamento ao expressar o valor. Aqui vão suas correspondências:

center – Centro

right – Direita

left – Esquerda

justify - Justificado

Os atributos que estudamos acima são aplicáveis apenas a um elemento inteiro, como um parágrafo. Caso desejamos atribuir determinado estilo apenas a um trecho de texto, podemos usar a tag <font> </font>, seguida dos seguintes atributos:

- **size** para determinar o tamanho da fonte. Utilizamos como valor, os números de -6 a +6. Faça algumas experiências para verificar os resultados.
- **face** para escolher a fonte. Mesma sintaxe dos valores estudados no atributo font-family.
- **color** para determinar a cor. Mesma sintaxe dos valores estudados no atributo color.

Vamos agora à um exemplo:

```
<p> Texto em tamanho normal, <font size="+2"> agora maior </font>, e <font color="#006699"> agora em azul </font>, voltando ao normal em seguida </p>
```

Texto em tamanho normal, agora maior , e agora em azul , voltando ao normal em seguida

Como vimos, podemos abrir e fechar a tag <font> no meio de um texto, assim como faríamos com uma formatação especial (como negrito ou sublinhado).

Assim como na subclasse style, podemos listar vários atributos em sequência na tag <font>, separando-os com vírgula.

**Exercício 5 – Formatando parte de um texto:** o objetivo desse exercício é verificar alguns conceitos sobre a sub-classe `style` e seus atributos, além de reforçar o uso de formatações especiais, como **negrito**. Recomendo fazer outros testes por conta própria para fixação.

Na frase abaixo, faça com que:

Por definição, **Exploit** é qualquer **programa**, comando ou sequência de dados que se **proveite da vulnerabilidade** de um sistema para invadi-lo.

1. O texto como um todo esteja em um parágrafo.
2. A palavra **Exploit** esteja em **negrito** e em azul.
3. A palavra **programa** esteja na fonte Arial.
4. As palavras **proveite da vulnerabilidade** estejam em vermelho, com fonte +2.

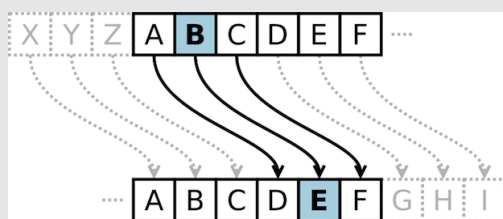
**Exercício 6 – Formatando todo um conjunto:** vamos agora utilizar a sub-classe `style` para alterar a composição de todo um parágrafo, título, ou subtítulo.

No texto abaixo, faça o que se pede:

**O que é criptografia?**

A criptografia é um conjunto de técnicas para esconder informação de acesso não autorizado.

O objetivo da criptografia é transformar um conjunto de **informação legível**, como um e-mail, por exemplo, em um emaranhado de caracteres impossível de ser compreendido.



A figura acima mostra uma cifra de César.

1. O título **O que é a criptografia** deve ser colocado em h1, centralizado, com a palavra **criptografia** em azul.
2. O primeiro parágrafo deve estar na fonte Arial, tamanho 16px.
3. O segundo parágrafo deve estar em vermelho, sendo que apenas as palavras **informação legível** devem continuar pretas. **Dica:** existem duas formas de fazer isso.
4. A imagem da cifra de César (disponível na pasta do exercício) deve estar centralizada, e sua largura deve ser de 320px.
5. A legenda da imagem deve estar centralizada, em verde e *itálico*.

Recomendo praticar os conteúdos acima.

## 11) Links

Agora que já trabalhamos a formatação de texto mais avançada e a inserção de imagens, veremos como funcionam os links.

Um link não é nada mais que uma ligação de um elemento da página para outro lugar – que pode ser uma página interna, externa ou arquivo.

Ao criar um link, temos que considerar duas coisas: a ligação, ou seja, para onde ele vai, e também a “máscara”, ou seja, o elemento usado para fazer essa ligação.

O primeiro – e único novo – passo é indicarmos ao programa qual elemento da página ele deve usar como ligação, e para onde.

Assim, inserimos a tag `<a>` no início do futuro link, e `</a>` no seu final.

Logo, se quisermos deixar o texto assim:

Quer um curso de HTML gratuito? [Clique Aqui!](#)

Devemos usar o seguinte código:

Quer um curso de HTML gratuito? `<a>` Clique Aqui! `</a>`

Porém, ainda precisamos indicar para onde o link deve ir. Fazemos isso com o atributo `href`, igualando-o ao link (seja ele um arquivo do projeto, ou um link externo). Por exemplo:

Quer um curso de HTML gratuito? `<a href="https://fabricadenoobs.wix.com/home">` Clique Aqui! `</a>`

Dessa forma, ao clicar em “Clique Aqui!” o usuário será redirecionado para o link do site especificado.

É importante lembrar de sempre utilizar a sintaxe `https://` antes do link, uma vez que, se for representado só por `www` ou equivalente, você corre o risco de seu código redirecionar para uma página (inexistente) do projeto, e não um link externo.

Se quisermos fazer com que o link seja aberto em uma nova página, usamos o atributo `target` e o igualamos a `_blank`. Existem outras sintaxes para `target`, mas elas fogem do foco deste curso. Veja o exemplo:

`<a href="https://twitter.com/" target="_blank">` Twitter `</a>`

Também podemos colocar uma imagem entre as tags <a> </a>, ou ainda usar a tag <font> para especificar um estilo diferente do padrão para o link.

**Exercício 7 – Experiências com links:** aqui, vamos testar alguns dos conhecimentos aprendidos neste capítulo sobre o uso de links.

Na frase abaixo, faça com que:

Acessando o [site oficial do canal](#), você pode encontrar download de programas, tutoriais e matérias sobre [Deep Web](#), além de explicações sobre [esteganografia](#).



1. O texto e a imagem estejam em um parágrafo.
2. [Site oficial do canal](#) deve ser um link para <http://fabricadenoobs.wix.com/home>.
3. [Deep Web](#) deve ser um link em vermelho para [https://pt.wikipedia.org/wiki/Deep\\_web](https://pt.wikipedia.org/wiki/Deep_web) (a matéria em si é horrível, mas não quero usar o mesmo site duas vezes).
4. [Esteganografia](#) deve ser um link para o arquivo esteganografia que está dentro da pasta do exercício.
5. A imagem (também na pasta) deve estar com largura 100 e ser um link para <https://www.youtube.com/channel/UCGObNjKNjo1OUPLm8BTb3A>.

## 12) Listas

Outro conteúdo de grande importância dentro do HTML são as listas. Com elas, podemos construir índices, ordenar valores e criar listas (sério?!).

Antes de irmos para a prática, vale lembrar que a indentação é uma ferramenta importante ao lidar com listas. Se o código não for bem tabulado, vai parecer uma orgia de tags no final. Procure fazer com que o próprio código-fonte já tenha uma aparência de lista, isso vai facilitar para você, e para quem for manipular o script.

Criar uma lista pode ser dividido em 2 passos: determinar o tipo da lista e criar seus elementos.

Existem 3 tipos de listas: as não ordenadas, ordenadas e de definição. Essa última é pouco usada, portanto, não mencionarei neste curso.

Uma lista não ordenada é organizada por símbolos, e não por numerais ordinais. Dessa forma, os itens não tem uma ordem de importância dentro da lista. No HTML, o símbolo utilizado para as listas é •.

Para determinar uma lista não ordenada, usamos as tags `<ul>` e `</ul>` no começo e fim da mesma. É como as tags `<body>`, indicando o fim e início do corpo.

Em seguida, usamos as tags `<li>` e `</li>` em cada elemento da lista. Não é obrigatório, mas é interessante colocar cada item em uma linha do código.

Vejam o exemplo:

```
<ul>
  <li> HTML </li>
  <li> CSS </li>
  <li> PHP </li>
  <li> JavaScript </li>
</ul>
```



Observe a forma como a lista está paragrafada. Poderíamos simplesmente colocar todos os elementos na mesma linha, mas ficaria estranho.

Outra lista que podemos criar é a ordenada. Nela, listamos vários itens, com uma ordem de importância. Para isso, usamos as tags `<ol>` e `</ol>`, da mesma forma que as anteriores. Veja no exemplo:

```
<ol>
  <li> Top </li>
  <li> Jungler </li>
  <li> Mid </li>
  <li> AdCarry </li>
  <li> Suporte </li>
</ol>
```

1. Top
2. Jungler
3. Mid
4. AdCarry
5. Suporte

Além disso, também podemos inserir uma lista dentro de outra, colocando-a dentro das tags `<li>` e `</li>`. Reforço ainda a necessidade de usar a indentação em situações como essa. Veja o código:

```
<ol>
```

```
  <li> Jogos de FPS
```

```
    <ul>
```

```
      <li> Battlefield </li>
```

```
      <li> Warface </li>
```

```
    </ul>
```

```
  </li>
```

```
  <li> Jogos de RPG
```

```
    <ul>
```

```
      <li> Perfect World </li>
```

```
      <li> World of Warcraft </li>
```

```
    </ul>
```

```
  </li>
```

```
</ol>
```

Ao compilar tal código, esse é o resultado esperado:

1. Jogos de FPS
  - Battlefield
  - Warface
2. Jogos de RPG
  - Perfect World
  - World of Warcraft

Vamos reproduzi-lo novamente para observar a sintaxe:

```
<ol>
```

```
  <li> Jogos de FPS
```



```
<ul>
  <li> Battlefield </li>
  <li> Warface </li>
</ul>
</li>
<li> Jogos de RPG
  <ul>
    <li> Perfect World </li>
    <li> World of Warcraft </li>
  </ul>
</li>
</ol>
```

Com o auxílio das cores, tente observar que o próprio código já está tabulado em forma de lista.

Primeiro, usamos as tags `<ol>` `</ol>` para iniciar uma lista ordenada.

Depois, abrimos e fechamos as tags `<li>` `</li>` para cada elemento da lista, deixando sua aparência inicial assim:

```
<ol>
  <li> Jogos de FPS</li>
  <li> Jogos de RPG</li>
</ol>
```

Com isso, tínhamos apenas uma lista ordenada.

Agora, depois do elemento (“Jogos de FPS”, por exemplo), abrimos e fechamos as tags `<ul>` `</ul>` entre as tags `<li>` `</li>` já inseridas.

Por fim, criamos os elementos da nova lista não ordenada com `<li>` `</li>`:

```
<li> Jogos de FPS
  <ul>
    <li> Battlefield </li>
    <li> Warface </li>
  </ul>
```

</li>

Outro ponto interessante é que também podemos aplicar a sub classe `style` e todos seus atributos nos elementos `<ul>` ou `<ol>` para alterar toda a lista, ou apenas nos elementos `<li>`, para alterar apenas o texto entre aquelas tags.

Também podemos aplicar as formatações especiais e links em toda a lista, ou em elementos específicos.

**Exercício 8 – Organizando uma lista:** crie uma lista com as seguintes estrutura e valores. Em seguida, faça com que:

Atores e filmes

1. Nicolas Cage
  - National Treasure
  - Snowden
2. Morgan Freeman
  - Now You See Me
  - Oblivion
3. Leonardo DiCaprio
  - Inception
  - Shutter Island

1. O título da lista deve estar em *h3*
2. Toda a lista Nicolas Cage deve ficar azul.
3. Apenas a palavra Morgan Freeman deve estar em *Itálico*.
4. Now You See Me deve estar em **negrito e verde**.
5. Toda a lista Leonardo DiCaprio deve estar em fonte *Arial* e em **negrito**.
6. Island (apenas Island) deve ficar **vermelho**.

## 13) Tabelas

Vamos agora para as tabelas. Elas são usadas para armazenar dados tabulares de forma que fiquem fáceis de ler.

Quando queremos apresentar informações em uma tabela com linhas e colunas, utilizamos inicialmente a tag `<table>` `</table>` para marcar o início e fim de uma tabela.

Também é importante especificarmos o tamanho da borda da tabela. Usamos o atributo `border` e o igualamos a algum valor em pixels. Podemos, com o atributo `bordercolor`, determinar uma cor para a borda da tabela (não é normalmente usado).

Vamos construir uma tabela gradualmente, para que você entenda melhor o processo. Assim, temos de início:

```
<table border="1px">  
  
</table>
```

Agora, vamos adicionar as linhas. Fazemos isso utilizando as tags `<tr>` e `</tr>` para cada linha que desejamos adicionar. Assim, se nossa tabela tem 3 linhas, nós utilizaremos essas tags 3 vezes.

```
<table border="1px">  
  <tr> </tr>  
  <tr> </tr>  
  <tr> </tr>  
</table>
```

Em HTML, não adicionamos colunas exatamente, mas sim vários elementos (ou células) na mesma linha, de forma que funcionem como colunas. Separamos cada elemento com as tags `<td>` `</td>`.

Sendo assim, vamos adicionar um elemento `<td>` a cada elemento `<tr>` e observar o resultado.

```
<table border="1px">  
  <tr> <td> Um </td> </tr>  
  <tr> <td> Dois </td> </tr>  
  <tr> <td> Três </td> </tr>  
</table>
```



Um
Dois
Três

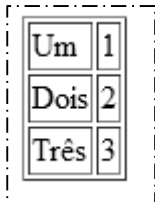
Criamos uma nova coluna para a tabela. Agora, vamos adicionar outra. Para tanto, inserimos uma nova sequência de elementos `<td>`, um em cada linha.

```
<table border="1px">
```

```
<tr> <td> Um </td> <td> 1 </td> </tr>
<tr> <td> Dois </td> <td> 2 </td> </tr>
<tr> <td> Três </td> <td> 3 </td> </tr>
```

</table>

Esse é o resultado:



Um	1
Dois	2
Três	3

Você provavelmente deve estar achando isso confuso, e com razão. Organizei o código dessa maneira para que pudéssemos compreender o conceito de linhas e colunas claramente.

Existem outras formas menos intuitivas, porém mais organizadas de se indentar uma tabela. Veja o exemplo:

```
<table border="1px">
  <tr>
    <td> Um </td>
    <td> 1 </td>
  </tr>

  <tr>
    <td> Dois </td>
    <td> 2 </td>
  </tr>

  <tr>
    <td> Três </td>
    <td> 3 </td>
  </tr>
</table>
```

De início, parece mesmo complicado. Mas com alguma prática, fica fácil identificar a formação final da tabela apenas pelo código.

Podemos deixar a tabela com uma aparência melhor adicionando atributos ao seu cabeçalho ou corpo.

Primeiramente, identificamos cada parte da tabela (cabeçalho e corpo) com `<thead>` `</thead>` e `<tbody>` `</tbody>`, respectivamente. Assim, temos:

```
<table border="1px">
  <thead>
</thead>
  <tbody>
    <tr>
      <td> Um </td>
      <td> 1 </td>
    </tr>

    <tr>
      <td> Dois </td>
      <td> 2 </td>
    </tr>

    <tr>
      <td> Três </td>
      <td> 3 </td>
    </tr>
  </tbody>
</table>
```

Como ainda não possuímos um cabeçalho, apenas colocamos as tags `<thead>` abaixo da tag `<table>`. Quanto as tags `<tbody>`, inserimos de forma que os valores da tabela fiquem dentro delas.

Vamos agora inserir no cabeçalho uma linha superior, aquela que dá nome a cada coluna. Fazemos isso inserindo primeiramente uma nova linha, com `<tr>`.

Depois, definimos cada elemento com `<th>` `</th>` (ao invés de `<td>`).

Como precisamos apenas de uma linha, iremos usar o conjunto `<tr>` `</tr>` uma única vez. O número de elementos `<th>` que usarmos vai depender do número de colunas que a tabela possui.

```
<thead>
  <tr>
    <th> Escrita </th>
    <th> Decimal </th>
  </tr>
</thead>
```

Por padrão, o cabeçalho final ficará em negrito, como abaixo. Porém, podemos usar a sub classe `style` para alterar a formatação de qualquer valor da tabela.

Por exemplo, se quisermos mudar a cor de todo o cabeçalho, ou de toda uma linha, basta adicionar esse atributo a tag `<tr>`.

Escrita	Decimal
Um	1
Dois	2
Três	3

Antes de prosseguir com mais detalhes, vamos a um exercício de fixação.

**Exercício 9 – Criando sua primeira tabela:** reproduza a tabela abaixo, tendo 1px como borda, o cabeçalho em vermelho e toda a tabela em fonte Arial.

Item	Preço
Mouse	R\$ 21,00
Teclado	R\$ 34,00
Monitor	R\$ 320,00

Também podemos colocar um título, o qual ocuparia, em uma linha, toda a extensão da tabela. É o mesmo que criar uma linha com apenas uma coluna, só que ocupando o espaço todo.

Inserimos o título como uma segunda linha no cabeçalho, acima da nomenclatura das colunas, usando as tags `<th>` e `<tr>`.

Para informar ao editor que a nova linha é na verdade um título, deve-se usar o atributo `colspan` na linha em questão (indicada pela tag `<th>`) e o igualar ao número de colunas da tabela.

Esse seria o cabeçalho com título da tabela trabalhada anteriormente:

```
<thead>
  <tr>
    <th colspan="2"> Valores </th>
  </tr>
  <tr>
    <th> Escrita </th>
    <th> Decimal </th>
  </tr>
</thead>
```

Ao passo que esse seria o resultado compilado (juntamente com o corpo da tabela):

Valores	
Escrita	Decimal
Um	1
Dois	2
Três	3

Além da sub classe `style`, também podemos colocar imagens como valores da tabela. Fazemos isso utilizando a tag `<img>` dentro das tags `<td>`.

Acompanhe a criação do código abaixo:

Temos essas 3 imagens, e queremos distribuí-las em uma galeria de fotos 1x3.



Primeiro, criamos um documento HTML, e nele inserimos o cabeçalho da tabela:

```
<table border="1px">
<thead>
  <tr>
    <th colspan="3"> Galeria de Fotos </th>
  </tr>
</thead>
```

Note que desejamos um título que ocupe todas às 3 colunas da futura tabela. Para simplificar, vou dispensar formatações de texto adicionais.

Vamos agora ao corpo. Como se trata de uma tabela 1x3, adicionaremos 1 tag <tr>, e 3 tags <td> dentro dela.

```
<tbody>
  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
</tbody>
</table>
```

Agora, focaremos apenas em um elemento <td>. Dentro dele, abrimos a tag <img> e determinamos a fonte com o atributo src. Conseguimos enquadrar a imagem na galeria, porém, se achamos ela grande ou pequena demais, podemos determinar seus atributos width ou heigth (nunca ambos, para não “estourar” a imagem ao colocar valores desproporcionais). Veja o resultado:

```
<td>  </td>
```

Fazemos o mesmo processo com as demais, lembrando que, caso for determinada a largura ou altura em uma delas, todas as outras devem ter os mesmos atributos.

O código final é:

```
<table border="1px">
  <thead>
    <tr>
```



```
<th colspan="3"> Galeria de Fotos </th>
</tr>
</thead>
<tbody>
<tr>
<td>  </td>
<td>  </td>
<td>  </td>
</tr>
</tbody>
</table>
```

E seu resultado:



Podemos ainda utilizar links nas tabelas, envolvendo o elemento entre as tags <a> </a>, e determinando a ligação com o atributo href. Veja o exemplo:

```
<td> <a href="https://www.youtube.com/">  </a> </td>
```

**Exercício 10 – Painel de redes sociais:** para verificar os conceitos aprendidos, vamos construir uma galeria de redes sociais – composta por ícones e links para seu perfil em diversas redes. Siga o exemplo e reproduza a tabela abaixo. Se preferir, use apenas o link de cada uma das redes. Os ícones estão na pasta do exercício.



1. A borda da tabela deve ter valor 1.
2. O título deve estar em azul, na fonte Arial.
3. Todos os links devem ser abertos em uma nova guia.

Vamos agora a nossa última parte sobre tabelas. Outro atributo legal que podemos usar é o `bgcolor`, que especifica uma cor de fundo.

Tal atributo pode ser usado tanto na tabela toda (na tag `<table>`), em uma linha (na tag `<tr>`), ou em uma única célula (nas tags `<th>` ou `<td>`).

Partindo da seguinte tabela, vamos deixá-la colorida:

Decimal e Binário	
Decimal	Binário
0	0
1	1
2	10

O código para sua construção segue abaixo. Recomendo construir a tabela por conta para praticar, mas se preferir, apenas cole-o no editor.

```
<table border="1px">
  <thead>
    <tr>
      <th colspan="2"> Decimal e Binário </th>
    </tr>
```

```

        <tr>
            <th> Decimal </th>
            <th> Binário </th>
        </tr>
</thead>
<tbody>
    <tr>
        <td> 0 </td>
        <td> 0 </td>
    </tr>

    <tr>
        <td> 1 </td>
        <td> 1 </td>
    </tr>

    <tr>
        <td> 2 </td>
        <td> 10 </td>
    </tr>
</tbody>
</table>

```

Agora, vamos fazer alguns procedimentos para deixar a tabela estilizada.

O primeiro passo é determinarmos uma cor para o fundo. Deixar ela cinza é uma boa opção.

Ao consultar a tabela de cores, vemos que cinza corresponde à #CCCCCC. Logo, temos:

```
<table border="1px" bgcolor="#CCCCCC">
```

Decimal e Binário	
Decimal	Binário
0	0
1	1
2	10

Próximo passo é mudar a cor do título da tabela para amarelo (ou verde, sei lá). Assim:

```
<th colspan="2" bgcolor="#CCFF66"> Decimal e Binário </th>
```

Decimal e Binário	
Decimal	Binário
0	0
1	1
2	10

Colorimos, então, (isso tá parecendo aqueles livrinhos infantis) a outra parte do cabeçalho da tabela. Observe que não é necessário atribuir o bgcolor para cada célula, apenas para a linha, na tag <tr>.

```
<tr bgcolor="#66FF66">
  <th> Decimal </th>
  <th> Binário </th>
</tr>
```

Decimal e Binário	
Decimal	Binário
0	0
1	1
2	10

E se desejarmos colorir todo o corpo da tabela, sem necessariamente envolver alguma linha em particular? Isso só vai ser possível com CSS. Apenas dentro do HTML, precisamos colorir cada uma das linhas, mesmo que todas tenham a mesma cor.

Dispensarei essa parte para evitar uma repetição.

Porém, vamos ainda alterar somente uma célula da tabela. Fazemos isso atribuindo bgcolor a alguma tag <td>, como no exemplo:

```

<tr>
    <td bgcolor="#996666"> 2 </td>
    <td> 10 </td>
</tr>

```





Decimal e Binário	
Decimal	Binário
0	0
1	1
2	10

Também podemos definir uma imagem (ao invés de uma cor) como plano de fundo da nossa tabela. Fazemos isso utilizando o atributo `background` ao invés de `bgcolor`, e o igualando ao link (externo ou local) da imagem relacionada.

Tais atributos podem ser usados em outros locais da página, como na tag `<body>`.

Nesse caso, recomendo criar suas próprias imagens de fundo baseadas no tamanho em questão (usando Photoshop ou similares). Caso contrário, sua página pode ficar com uma péssima aparência.

**Exercício 11 – Tabela de cores:** por fim, faremos uma tabela relacionando o nome de uma determinada cor, seu equivalente em HTML e a cor propriamente dita. Crie a sua, com base no exemplo:

Tabela de Cores em HTML		
Nome	Em HTML	Cor
Azul	blue	
Verde	green	
Amarelo	yellow	
Vermelho	red	

1. A borda da tabela deve ter valor 1.

## 14) Divisões na Página

Temos ainda algumas tags aplicadas para estilizar a página mais facilmente dentro do CSS. Como o foco do curso é HTML, irei apenas citar seus usos.

A tag `<div> </div>` pode ser usada para estilizar uma grande área da página, sem a necessidade de alterar elemento por elemento.

Por exemplo, temos toda uma página HTML que precisa ser escrita em Arial. Mas é inviável criar um atributo `style` para cada tag. Então, abrimos e fechamos `<div> </div>` em toda a área afetada, e atribuímos seu estilo:

```
<div style="font-family:Arial; color:blue" >
    <h1> Título </h1>
    <h3> Subtítulo </h3>
    <p> Parágrafo </p>
</div>
```

**Título**

Subtítulo

Parágrafo

O contorno pontilhado é adicionado no Dreamweaver para diferenciação, mas não aparece no resultado final.

Também temos a tag `<span> </span>`, que, dentro do HTML, tem o mesmo funcionamento da tag `<font>`: o de estilizar um pequeno espaço de texto. Irei aprofundá-la no curso de CSS.

## 15) Conclusão

Por fim, concluímos nosso curso de HTML. Agradeço a você pela paciência, peço desculpas por qualquer falha, e estou sempre aberto a sugestões.

Para os que pretendem se aprofundar no assunto, recomendo esperar o lançamento do próximo curso, que será sobre CSS.