
PROGRAMAÇÃO DE SCRIPTS

HTML

RESUMO DAS AULAS

RESUMO DAS AULAS DE HTML

PARTE 1 - Introdução	3
1 - Conceitos Básicos.....	3
2 - Estrutura de uma Página Web HTML	4
3 - Primeiros Elementos	4
4 - Parágrafos.....	5
5 - Configurando Textos	6
PARTE 2 - Links e Imagens	7
6 - Documentos com <i>Hiperlinks</i>	7
7 - Exibição de Imagens	7
PARTE 3 - Configurando Listas	8
8 - Listas.....	8
PARTE 4 - Melhorando a Aparência	9
9 - Cores.....	9
10 - Formatando Parágrafos	9
PARTE 5 - Âncoras	10
11 - Âncoras	10
PARTE 6 - Tabelas	10
12 - Utilizando Tabelas.....	10

Hiper Text Markup Language

Este material é apenas um resumo para acompanhamento dinâmico das aulas ministradas, sendo necessário o acompanhamento das aulas e as orientações do professor.

PARTE 1 - Introdução

1 - Conceitos Básicos

- a. **Página web** é um documento composto basicamente de textos e códigos especiais chamados **tags** (etiquetas), que possibilitam a exibição do documento em um **navegador** (*browser*) na internet. Pode conter imagens, sons, animações, vídeos, links e formulários. Basicamente são escritos em linguagem de marcação de hipertextos **HTML**, apesar de receberem implementos em linguagens de *scripts* - como **Javascript** por exemplo - linguagens dinâmica de programação - como **PHP** por exemplo - dentre outras.
- b. **World Wide Web** é a rede mundial de computadores, conhecida como **Web**, **www** ou **W₃**, é um conjunto de documentos multimídia conectados através de *hiperlinks* que permitem o deslocamento entre documentos na internet. O ato de seguir hiperligações é chamado "navegar" ou "surfar" na Web.
- c. **URL (Uniform Resource Locator - Localizador Uniforme de Recursos)** é o endereço na Web no qual se encontram recursos como uma página web, um arquivo de computador ou um dispositivo periférico (impressora, equipamento multifuncional, unidade de rede etc.). Essa rede pode ser a internet, uma rede corporativa (intranet). Um URL completo possui a seguinte estrutura:

```
protocolo://domínio:porta/caminho/recurso  
?parâmetro=argumento#fragmento
```

- **protocolo** pode ser HTTP, HTTPS, FTP, etc.
- **domínio** é o endereço do servidor que hospeda o documento/recurso solicitado, sendo **:porta** opcional como um ponto lógico no qual se executa a conexão com o servidor.
- O **caminho** especifica o local (pasta, sub-pastas, diretórios) onde se encontra o **recurso**, dentro do servidor.
- **parâmetro=argumento** é um conjunto de pares de parâmetros com seus respectivos argumentos (senha=123, por exemplo). É opcional, sendo uma string enviada ao servidor para que seja possível filtrar o recurso.
- O **fragmento** é uma parte opcional, uma âncora, ou posição específica dentro do recurso.

- d. **Endereço IP (Internet Protocol ou Protocolo de Internet)**, é uma identificação de um dispositivo (computador, impressora, etc.) em uma rede, sendo este o meio em que as máquinas usam para se comunicar na Internet. Cada conexão em uma rede possui um IP único. Para facilitar a navegação, comumente utilizamos endereços de domínio, tal como **www.google.com**, sendo este convertido em um endereço IP pelo **DNS (Domain Name System)**. Os endereços IP são divididos em estruturas de tamanho fixo chamadas de **classes de endereço**. As principais são as classes A, B e C:

Classe	Gama de Endereços	Qtde. de Endereços
A	1.0.0.0 até 127.0.0.0	16 777 216
B	128.0.0.0 até 191.255.0.0	65 536
C	192.0.0.0 até 223.255.255.0	256

- Endereços da **Classe A** permitem menos **redes** e mais **hosts** por rede, enquanto que os endereços da **Classe C** permitem mais **redes** e menos endereços (**hosts**) disponíveis em cada rede.
- e. **DNS (Domain Name System - Sistemas de Nomes de Domínio)** é um sistema de gerenciamento de nomes para computadores, serviços ou qualquer recurso conectado à internet. Comumente, traduz os nomes de domínios - que são facilmente memorizáveis pelos usuários - para os endereços IP numéricos. Este processo chama-se **resolução de nome**. O banco de dados onde se armazenam **DNS** é distribuído, assim seu tamanho é ilimitado e o desempenho não se degrada muito quando se adicionam mais servidores.
- f. **TCP/IP** é um conjunto de protocolos de comunicação entre computadores em rede, sendo **TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão)** e o **IP (Internet Protocol - Protocolo de Internet)**. O conjunto de protocolos pode ser visto como um modelo de camadas onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas, estão logicamente mais perto do usuário e lidam com dados mais abstratos, sendo os protocolos de camadas mais baixas para tarefas de menor nível de abstração. A **Tabela 1** mostra as camadas e suas aplicações.

Camadas	Aplicações
1ª e 2ª Interface com Rede	Camadas físicas pois tratam-se das tecnologias usadas para as conexões. Exemplos: Wi-Fi, Modem.
3ª Internet	Camada responsável pela interconexão entre as redes locais. Exemplo IP.
4ª Transporte	Camada que controla a comunicação ponto-a-ponto, host-to-host. Exemplo TCP.
5ª, 6ª e 7ª Aplicação	Camadas que contêm os protocolos para um serviço específico de comunicação de dados no nível processo-a-processo, como um navegador deve se comunicar com um servidor da web. Exemplos: HTTP, HTTPS, FTP, DNS.

Tabela 1

- g. **HTTP (Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto)** é um protocolo de comunicação de requisição-resposta no modelo computacional cliente-servidor (em uma das camadas de aplicação), utilizado para sistemas de informação de hipermídia. Ele é a base para a comunicação de dados da *World Wide Web*. **Hipertexto** é o texto estruturado que utiliza ligações lógicas (hiperlinks) entre pontos que contenham texto. Um navegador pode ser o cliente e uma aplicação em um computador que hospeda um *site* da web pode ser o servidor. O cliente submete uma mensagem de requisição **HTTP** para o servidor. O servidor retorna uma mensagem resposta para o cliente.

1.1 - Visualizando Páginas Web

Para visualizar uma página web é preciso digitar uma URL em um navegador instalado na máquina onde se deseja exibir a página web. Primeiramente, a parte da URL referente ao servidor de rede (**domínio**) é separada e transformada em um endereço IP, pelo processo de resolução de nome (DNS). Assim, o navegador estabelece uma conexão TCP-IP com o servidor web localizado no endereço IP retornado.

O próximo passo é o navegador enviar uma requisição HTTP ao servidor para obter o recurso indicado pela parte restante da URL. Em seguida, o texto HTML é recebido e **interpretado** pelo navegador. **Não é compilado**. A partir daí, requisições adicionais são feitas para carregar figuras, arquivos de formatação (como .CSS), arquivos de *script* (como .JS) entre outros recursos que compõem a página. O navegador, então, renderiza (constrói) a página na tela.

1.2 - Utilizando um Navegador

Não é preciso estar conectado à internet para se utilizar um navegador, portanto, é muito simples testar as páginas web que estiverem sendo editadas, bastando para isso executar o navegador em um micro e carregar o programa editado. Às vezes, acontece de uma tag ser reconhecida por um determinado navegador e não ser por outro. Para que isso seja testado, é interessante que se tenha instalado em seu micro pelo menos os dois principais Navegadores do mercado, e testar sua Página Web nos dois. Em caso de incompatibilidade, o melhor é escrever os comandos duas vezes (uma vez com a sintaxe de um navegador e a outra vez com a sintaxe do outro).

2 - Estrutura de uma Página Web HTML

Para escrever uma página web em HTML, o programador deve digitar o código-fonte em um editor de textos qualquer, desde que permita salvar com a extensão .HTM ou .HTML, seguindo as regras de sintaxe do HTML.

O código-fonte de uma página HTML pode ser digitado com caracteres maiúsculos ou minúsculos, pois o HTML não é *key sensitive*. As *tags* podem ser digitadas na mesma linha ou em linhas subsequentes. O programador deve sempre estar atento com abertura e fechamento de *tags* e aspas. O HTML possui **estrutura linear e não compilada**, portanto não haverá lista de erros.

2.1 - Sintaxe das Tags

Podem ser especificadas em pares ou individuais e estão sempre envolvidas pelos sinais < > para abertura da tag e </> para fechamento, também chamados de containeres.

<TAG> texto </TAG>.

Exemplo:

Texto em **negrito** .

Nem todas as tags precisam de fechamento, como por exemplo, a tag para quebra de linha:

Texto em uma linha.

Texto na linha seguinte.

Neste resumo de aulas, o estudo do HTML considera a utilização da versão **HTML 4.01**, mas inclui também a nova versão **HTML 5**. A maioria das tags do HTML 4.01 se aplicam ao HTML 5, porém é necessária a utilização de estilos em CSS para configurar seus parâmetros. Para *tags* ou parâmetros que não se aplicam no HTML 5 o símbolo ☒ é indicado.

2.2 - Estrutura Básica da Página Web

Nem todas as *tags* são obrigatórias, mas basicamente o código fonte HTML segue a seguinte estrutura:

```
<!DOCTYPE html>
<HTML>
<HEAD>
    <!-- comandos de cabeçalho e estilo -->
    <TITLE>Título da Página</TITLE>
</HEAD>
<BODY>
    <!-- montagem do corpo da página -->
</BODY>
</HTML>
```

3 - Primeiros Elementos

a. <!DOCTYPE html>

Todo documento HTML deve conter um DOCTYPE em seu início, antes da tag <HTML>, para informar ao navegador qual é a versão do HTML que está sendo utilizado. Se não for declarado, o navegador tentará adivinhar qual é o DOCTYPE, podendo provocar uma renderização defeituosa. No HTML5 o DOCTYPE é referenciado como um cabeçalho inútil mas necessário, apenas para ativar o modo padrão de renderização em navegadores comuns, isso porque o HTML5 não é baseado em SGML, usando o DOCTYPE apenas para seleção de modo sem utilizar DTD (Declaração do Tipo de Documento). A sintaxe é apenas:

```
<!DOCTYPE html>
```

Até o HTML 4.01, o DOCTYPE necessitava-se fazer referência a um DTD, podendo ser DOCTYPE **Strict**, **Transitional** ou **Frameset**, como segue:

- **HTML 4.01 Strict:** código precisa ser escrito de forma perfeita, sem erros ou esquecimentos. Não utiliza tags e atributos de apresentação como , <BODY BGCOLOR> por exemplo, mas sim utilizar CSS para definir estilos de apresentação.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **HTML 4.01 Transitional:** para ajudar os web designers na aprendizagem, menos sensível a alguns erros e esquecimentos ao usar *tags* e atributos de apresentação. É como uma transição entre a forma antiga de codificar para a forma nova e correta.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **HTML 4.01 Frameset:** seguindo a mesma ideia do transicional, porém para a construção de páginas utilizando utilização de FRAMES.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN""http://www.w3.org/TR/html4/
frameset.dtd">
```

b. <HTML> ... </HTML>

Especifica que dentro dessas tags há comandos HTML. Representa a raiz de um documento HTML. Todos os outros elementos são descendentes de <HTML>.

- **lang** = define o idioma (pt-br)

```
<HTML lang="pt-br">
```

c. <HEAD> ... </HEAD>

Área destinada às tags relacionadas com a identificação da página e formatação de estilos. Sua utilização não é obrigatória, entretanto, há tags que necessitam ser declaradas nesta área, como o <TITLE> por exemplo. Também no <HEAD> são incluídos metadados, <META>,

que são dados de alto nível que não são exibidos na tela, mas exercem influência na execução da página.

```
<HEAD>
  <META charset="UTF-8">
</HEAD>
```

O exemplo acima, mostra a definição do conjunto de caracteres UTF-8 que permite exibir caracteres da língua portuguesa.

d. <TITLE>...</TITLE>

Especifica o título da página que aparecerá na aba do navegador. A digitação incorreta desta tag acarretará problemas na exibição da página.

e. <BODY> ... </BODY>

Especifica o início e o fim do corpo principal da página web. Há apenas um único <BODY> em cada página HTML. Até a versão HTML 4.01 possui os seguintes parâmetros opcionais de configuração de cores da página web:

- **bgcolor** = cor do fundo. Padrão é branco.
- **text** = cor das fontes da página inteira. Padrão é preto.
- **link** = cor dos links. Padrão é azul.
- **vlink** = cor dos links visitados. Padrão é marrom.
- **alink** = cor dos links em acesso. Padrão é azul.

```
<BODY bgcolor="lightblue" text="darkblue"
link="red" vlink="orange" alink="pink">
```

No HTML 5 é necessário utilizar <style> em CSS.

```
<HEAD>
<style>
  a:link {
    color: red;    //para link não visitado
  }
  a:visited {
    color: green;  //para link visitado
  }
  a:hover {
    color: hotpink; //ao passar o mouse sobre
  }
  a:active {
    color: blue;   //ao clicar sobre o link
  }
</style>
</HEAD>
```

- **background** = imagem no fundo, que pode ser multiplicada na tela. Não deve ser utilizada junto com **bgcolor**.

```
<BODY background="foto1.jpg" text="green"
link="brown" vlink="cian" alink="magenta">
```

No HTML 5 é necessário utilizar <style> em CSS.

```
<HEAD>
<style>
body {
  background-image:    url("foto1.jpg");
  background-repeat:   no-repeat;
  background-position: right top;
  background-attachment: fixed;
}
</style>
</HEAD>
```

Ou ainda, CSS no modo curto:

```
<HEAD>
<style>
body {
  background: #00ff00 url("smiley.gif") no-repeat
fixed center;
}
</style>
</HEAD>
```

f. <!-- comentários -->

Permite digitar comentários em qualquer parte do programa. Esses comentários não serão exibidos.

```
<!-- Elaborado pelo Prof. Celso Gallão -->
```

4 - Parágrafos

O navegador, ao executar o HTML, não reconhece o fim de um parágrafo apenas porque o programador escreveu o código em linhas diferentes. Mesmo que o programador tenha pressionado [ENTER] no código fonte HTML, o navegador não reconhecerá a quebra de linha ou parágrafo. É necessário utilizar tags específicas, a saber:

a. <P>

Ao utilizar apenas <P>, especifica a quebra de parágrafo e **insere** uma linha em branco após.

```
<BODY>
Texto do 1º parágrafo <P>
Texto do 2º parágrafo
</BODY>
```

b. <P>...</P>

Pode ser utilizado com o seu par, marcando o início e o fim de um parágrafo. Até a versão HTML 4.01 possui o seguinte parâmetro opcional:

- **align** = alinhamento (left, center, right)

```
<BODY>
<P align="right">Este parágrafo será exibido
centralizado na tela.</P>
</BODY>
```

**c.
**

Especifica uma simples quebra de linha **sem inserir** uma linha em branco após.

```
<BODY>
Texto na 1ª linha <BR>
Texto na 2ª linha
</BODY>
```

d. <PRE> ... </PRE>

Permite que um texto seja reconhecido pelo navegador com sua formatação original, reconhecendo a tecla [TAB] para tabulação e [ENTER] para final de linha. A fonte utilizada é a **Courier** pois mantém cada caracter com a mesma largura.

```
<PRE>
Agora sim,
o HTML irá reconhecer a tecla [ENTER]
para quebra de linha.
    Até as margens com [TAB] são reconhecidas.
</PRE>
```

e. <DIV> ... </DIV>

Marca o início e o fim de um bloco de texto. Utilizada para agrupar e formatar elementos em um bloco. Por padrão, os navegadores inserem quebra de linha antes e após o <DIV>, mas isso pode ser alterado com CSS. Até a versão HTML 4.01 possui o seguinte parâmetro opcional:

- **align** = alinhamento (left, center, right, justify)

```
<DIV align="right">
Textos do bloco que serão visualizados à direita.
</DIV>
```

No HTML 5 é necessário utilizar <style> em CSS.

```
<DIV style="color:#0000FF">
Textos do bloco que serão exibidos com a cor azul.
</DIV>
```

f. **Textos em Cabeçalhos** <H1> até <H6>

É um tipo de formatação de texto que pode ser utilizada em qualquer lugar da página, com 6 configurações diferentes. Por padrão, são 6 tamanhos de fonte, sendo a maior com <H1> até a menor com <H6>. Automaticamente será adicionada uma linha em branco antes e outra depois. Até a versão HTML 4.01 possui o seguinte parâmetro opcional:

- **align** = alinhamento (left, center, right, justify)

```
<H1 align = "right">Texto Centralizado</H1>
```

No HTML 5 é necessário utilizar <style> em CSS.

```
<HEAD>
<style>
h1 {
    display:      block;
    font-size:    2;
    margin-top:   0.67;
    margin-bottom: 0.67;
    margin-left:  0;
    margin-right: 0;
    font-weight:  bold;
}
</style>
</HEAD>
```

g. <HR>

Até a versão HTML 4.01, <HR> traça uma linha horizontal na tela e possui os seguintes parâmetros opcionais:

- **size** = espessura da linha.
- **width** = comprimento da linha.
- **align** = alinhamento (left, center, right).
- **color** = cor da linha.

```
<HR SIZE=2 WIDTH=50 ALIGN="left" COLOR="red">
```

No HTML 5 é necessário utilizar <style> em CSS.

```
<HEAD>
<style>
hr {
    display:      block;
    margin-top:   0.5;
    margin-bottom: 0.5;
    margin-left:  auto;
    margin-right: auto;
    border-style:  inset;
    border-width:  1px;
    border-color:  black;
    width:        50%;
    text-align:    left;
}
</style>
</HEAD>
```

5 - Configurando Textos

As tags que alteram as configurações de textos devem ser especificadas marcando o início e o fim da aplicação. Até a versão HTML 4.01 as tags eram:

- | | | | |
|----|---------|---------------------------|----------|
| a. | | texto em negrito | |
| b. | <I> | texto em itálico | </I> |
| c. | <U> | texto sublinhado | </U> |
| d. | ⊗ <BIG> | aumenta a fonte + negrito | </BIG> |
| e. | <SMALL> | reduz a fonte | </SMALL> |
| f. | ^{| texto sobrescrito |} |
| g. | _{| texto subscrito |} |

No HTML 5 opcionalmente pode-se utilizar <style> em CSS para configurar os parâmetros, conforme segue:

```
<HEAD>
<style>
b {
    font-weight: bold;
}
i {
    font-style: italic;
}
u {
    text-decoration: underline;
}
small {
    font-size: smaller;
}
sub {
    vertical-align: sub;
    font-size: smaller;
}
sup {
    vertical-align: super;
    font-size: smaller;
}
</style>
</HEAD>
```

No HTML 5, todas as *tags* anteriores (exceto <BIG>) são aceitas. A seguir, uma série de *tags* que já existiam no HTML 4.01, mas que só têm sentido se configuradas com <style> em CSS:

- | | | | |
|----|----------|--------------------------|-----------|
| h. | | texto destacado | |
| i. | | texto enfatizado | |
| j. | <MARK> | texto realçado (pintado) | </MARK> |
| k. | <CITE> | texto de citação | </CITE> |
| l. | <DFN> | texto de definição | </DFN> |

À seguir exemplos da configuração padrão destas tags:

```
<HEAD>
<style>
strong {
    font-weight: bold;
}
em {
    font-style: italic;
}
mark {
    background-color: yellow;
    color: black;
}
cite {
    font-style: italic;
}
dfn {
    font-style: italic;
}
</style>
</HEAD>
```

Editar o EXERCÍCIO 01 do Caderno de Exercícios

PARTE 2 - Links e Imagens

6 - Documentos com Hiperlinks

São documentos que se vinculam à outros documentos através de ligações que podem ser textuais ou mesmo por imagens. São os chamados **LINKS**, ou seja, ligações de uma página web com outra.

6.1 - Link para Página Web Local

Pode ser texto ou imagem que, ao ser clicada, fará ligação com o documento que for mencionado, bastando para isso relacionar o endereço local do documento ao *hiperlink*. A tag utilizada é `<A>...` com o parâmetro `href`:

```
<A href="path\pagina.htm">hiperlink</A>
```

```
<A href="index.htm">Página Principal</A>
```

Quando este *link* for clicado é exibido o documento (neste caso uma página web) chamado `index.htm`, que deverá estar no mesmo diretório da página web que o acessou, caso contrário deverá ser indicado o *path* (caminho) completo do documento, com as barras invertidas `"\"`.

```
<A href="c:\arquivos\index.htm">
Voltar à Página Principal</A>
```

Para subir um nível na hierarquia dos diretórios (pastas) utilize a marcação com ponto-ponto-barras `"..\"`.

```
<A href="..\arquivos\pagina2.htm">Página 2</A>
```

6.2 - Link para Página Web Externa

Neste caso o *link* fará ligação com um documento externo, ou seja, que está conectado a uma rede de computadores como internet ou intranet. A tag utilizada é `<A>...` com o parâmetro `href`. Entretanto, o endereço tem de ser a URL completa do documento de destino:

```
<A href="URL">hiperlink</A>
```

```
<A href="http://www.fatecsaocaetano.edu.br/">Site
da Fatec São Caetano</A>
```

6.3 - Definindo onde Abrir a Página Web

A tag `<A>` possui diversos parâmetros que serão abordados mais adiante. Entretanto um deles é o responsável por informar onde a página web de destino do *link* deverá ser exibida. O parâmetro chama-se **target** e permite direcionar para `_blank`, `_parent`, `_self`, `_top` ou ainda para um nome de frame.

```
<A href="http://www.google.com" target="_blank">
Site do Google</A>
```

No HTML 5 pode-se utilizar `<style>` em CSS.

```
<HEAD>
<style>
a:link, a:visited {
    color: red;
    text-decoration: underline;
    cursor: auto;
}
</style>
</HEAD>
```

7 - Exibição de Imagens

A tag utilizada é `` juntamente com o parâmetro obrigatório `src="URL"`. Não possui tag de fechamento. A seguir, alguns parâmetros de ``.

- **src** = URL do arquivo de imagem.
- **alt** = texto alternativo sobre a imagem.
- **width** = largura da imagem (pixels ou %).
- **height** = altura da imagem (pixels ou %).
- **border** = borda da imagem.
- **align** = alinhamento em relação ao texto.

```
<IMG src="foto1.jpg" alt="Foto de Teste" width=200
border=1 align="top">
```

Para exibir uma imagem na página web é necessário que esta esteja em um dos formatos aceitos pelo navegador:

- **GIF (Graphics Interchange Format)**: É o formato padrão original dos navegadores. Permite GIF animado - que foram muito explorados no passado - e *background* transparente. Suporta apenas 256 cores (8 bits), sendo pouco recomendado imagens muito coloridas ou com texturas.
- **JPEG (Joint Photographic Engineering Group)**: É o formato mais utilizado pela sua maior capacidade de compactação. Suporta 16.777.216 cores (24 bits), sendo recomendado para fotografias digitais de alta resolução. Também é o formato padrão da maioria das máquinas digitais e celulares com câmera digital.
- **PNG (Portable Network Graphics)**: É o formato que permite imagem com alta compactação e sem perda de qualidade. Também suporta 16.777.216 cores (24 bits), permite animação e *background* transparente.

7.1 - Imagem Utilizada como Link

Para exibir uma imagem usando-a como um *link*, basta utilizar a tag `<A href>` em conjunto com ``.

```
<A href="pagina01.htm"> <IMG src="foto.jpg"> </A>
```

No exemplo acima, quando o usuário clicar em `foto.jpg` será carregado o documento `pagina01.htm`.

7.2 - Link para Enviar E-Mail

Para que o usuário envie e-mail apenas clicando em um hiperlink, utiliza-se a tag `<A href>`, mas especifica-se no URL o parâmetro `mailto`: seguido do e-mail de destino.

```
<A href="mailto:celso.gallao@fatec.sp.gov.br">
Clique aqui para enviar e-mail </A>
```

No exemplo acima, quando o usuário clicar no hiperlink será carregado o aplicativo para envio de e-mail que estiver instalado na máquina, como o *Outlook* por exemplo, e o e-mail descrito em `mailto`: já estará especificado como e-mail de destino.

Também pode-se enviar outras partes da mensagem, como **Assunto**, **Corpo**, **Texto** e ainda especificar o modo de envio `cc` (com cópia) e `bcc` (com cópia oculta). Adiciona-se o símbolo `?` (interrogação) após o endereço de e-mail seguido dos parâmetros abaixo:

- **subject** = assunto do e-mail
- **body** = mensagem no corpo do e-mail
- **cc** = e-mail para receber cópia
- **bcc** = e-mail para receber cópia oculta

Todos os parâmetros acima devem estar após o símbolo `?` (interrogação) e separados pelo `&` (e comercial).

```
<A href="mailto:gallao@fatec.sp.gov.br?
subject=Felicidade&body=Estou muito feliz com esta
aula&cc=celso@teste.com&bcc=gallao@teste.com">
Clique aqui para enviar e-mail </A>
```

PARTE 3 - Configurando Listas

8 - Listas

As listas facilitam a criação de itens seriais. Podem ser **Ordenadas** (com itens numéricos) ou **Não Ordenadas** (com marcadores). Ao definir sub-listas tem-se então **Listas Aninhadas**, podendo ser **Mistas** ou **não**, conforme segue:

8.1 - Listas Não Ordenadas

A tag ` ... ` define o início e o fim de listas **não ordenadas**. Cada **marcador** será exibido na posição da tag ``.

```
<UL>
  <LI>Item 1</LI>
  <LI>Item 2</LI>
  <LI>Item 3</LI>
</UL>
```

A partir da estrutura acima, será exibido:

- Item 1
- Item 2
- Item 3

8.2 - Listas Ordenadas

A tag ` ... ` define o início e o fim de listas **ordenadas**. Cada **número** é exibido na posição da tag ``.

```
<OL>
  <LI>Item 1</LI>
  <LI>Item 2</LI>
  <LI>Item 3</LI>
</OL>
```

A partir da estrutura acima, será exibido:

1. Item 1
2. Item 2
3. Item 3

8.3 - Listas Aninhadas Mistas

Basta iniciar e finalizar *tags* de lista dentro de outras *tags* de lista.

```
<OL>
  <LI>Tecnologia em Jogos Digitais</LI>
  <LI>Tecnologia em ADS</LI>
    <UL>
      <LI>Engenharia de Software</LI>
      <LI>Programação de Scripts</LI>
        <OL>
          <LI>HTML</LI>
          <LI>Javascript</LI>
          <LI>PHP</LI>
        </OL>
      <LI>Inteligência Artificial</LI>
    </UL>
  <LI>Tecnologia em Segurança da Informação</LI>
</OL>
```

A partir do exemplo acima, será exibido:

1. Tecnologia em Jogos Digitais
2. Tecnologia em ADS
 - Engenharia de Software
 - Programação de Scripts
 1. HTML
 2. Javascript
 3. PHP
 - Inteligência Artificial
3. Tecnologia em Segurança da Informação

8.4 - Complementos

Parâmetros complementares podem ser aplicados às listas, conforme segue:

- **type** = tipo de marcador (1, a, A) em ``.
- **start** = posição inicial do marcador em ``.
- **reversed** = ordem decrescente em ``.

```
<OL reversed type="A" start="3" >
  <LI>Tecnologia em Jogos Digitais</LI>
  <LI>Tecnologia em ADS</LI>
  <LI>Tecnologia em Segurança da Informação</LI>
</OL>
```

A partir do exemplo acima, será exibido:

- C. Tecnologia em Jogos Digitais
- B. Tecnologia em ADS
- A. Tecnologia em Segurança da Informação

No HTML 5 pode-se utilizar `<style>` em CSS.

```
<HEAD>
<style>
ol {
  display: block;
  list-style-type: decimal;
  margin-top: 5px;
  margin-bottom: 5px;
  margin-left: 0;
  margin-right: 0;
  padding-left: 40px;
}
ul {
  display: block;
  list-style-type: disc;
  margin-top: 5px;
  margin-bottom: 5px;
  margin-left: 0;
  margin-right: 0;
  padding-left: 40px;
}
</style>
</HEAD>
```

Pode-se definir listas sem marcadores, utilizando, as *tags* `<DL> ... </DL>` como início e fim da lista, `<DT> ... </DT>` para o 1º nível e `<DD> ... </DD>` para o 2º nível:

```
<DL>
  <DT>Html</DT>
  <DD>Html Básico</DD>
  <DT>Aprender</DT>
</DL>
```

A partir do exemplo acima, será exibido:

- Html
 - Html Básico
- Aprender

[Editar o EXERCÍCIO 02 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 03 do Caderno de Exercícios](#)

PARTE 4 - Melhorando a Aparência

9 - Cores

Os navegadores possibilitam até 256 tons para cada uma das cores padrão RGB (Red-Green-Blue). Portanto, cada cor é representada por 3 dígitos, de 000 a 255. O HTML só reconhecerá esses dígitos se estiverem convertidos na base numérica **Hexadecimal**, ou seja, de 00 a FF, no formato **#RRGGBB**. Também pode-se declarar as cores especificando seus nomes em inglês (tabela RGB). A seguir, uma pequena amostra da tabela de cores RGB.

Cor	Textual em inglês	hexadecimal
Preto	black	#000000
Branco	white	#FFFFFF
Azul	blue	#0000FF
Amarelo	yellow	#FFFF00
Verde	green	#008000
Lima	lime	#00FF00
Marrom	brown	#800000
Oliva	olive	#808000
Azul celeste	aqua	#00FFFF
Lilás	fuchsia	#FF00FF
Cinza	gray	#808080
Azul naval	navy	#000080
Roxo	purple	#800080
Verde naval	teal	#008080
Prata	silver	#C0C0C0
Vermelho	red	#FF0000

10 - Formatando Parágrafos

O padrão de alinhamento dos parágrafos em HTML é à esquerda mas podemos modificá-los, assim como o tipo e a cor da fonte, utilizando alguns parâmetros, conforme segue:

10.1 - O Parâmetro align

É utilizado nas tags <P>, <DIV> ou <H1> até <H6>.

- **align** = right, center ou left.

```
<P align = "center"> texto centralizado </P>
<DIV align = "right"> texto à direita </DIV>
<H1 align = "right"> texto à direita </H1>
```

Mas atenção, porque **align** é apenas um parâmetro das tags acima. Jamais utilize **align** como uma tag separada.

Não existe a tag <align>!

10.2 - A Tag <CENTER>...</CENTER>⊗

Até a versão HTML 4.01 era possível centralizar textos, como um marcador de início e fim, através da tag <CENTER>...</CENTER>⊗. O HTML 5 não o reconhece.

```
<CENTER> texto centralizado </CENTER>
```

10.3 - A Tag ...⊗

Até a versão HTML 4.01 era possível alterar o tipo, tamanho, e a cor das fontes através da tag ...⊗ e seus parâmetros abaixo. Não é reconhecido pelo HTML 5.

- **face** = tipo da fonte
- **size** = tamanho da fonte
- **color** = cor da fonte

```
<FONT face="arial" size=5 color="#FF0000">
```

No HTML 5 pode-se utilizar <style> em CSS.

```
<HEAD>
<style>
H1 {
    color:        black;
    font-weight:  bold;
    font-size:    15px;
    font-family:  arial, verdana;
    text-align:   center
}
</style>
</HEAD>
```

10.4 - Texto em Movimento

A tag <MARQUEE>...</MARQUEE> faz com que o texto fique em movimento horizontal na tela. Esta tag não faz parte da documentação oficial do HTML por ser considerado efeito visual, portanto deve ser gerenciado pelo CSS. Não é reconhecido por alguns os navegadores. O padrão de movimento é da direita para a esquerda, mas pode ser alterado com os seus parâmetros:

- **behavior**: define a forma de rolagem do texto:
 - **scroll** - rolagem continua em um mesmo sentido.
 - **slide** - o texto para ao chegar ao final da rolagem.
 - **alternate** - o texto desliza de um lado para o outro, sem parar.
- **direction**: define a direção de rolagem do texto:
 - **left** - para a esquerda.
 - **right** - para a direita.
- **loop**: define o número de vezes que o texto deslizará:
 - **infinite** (padrão), define uma rolagem constante.
- **scrollamount**: define a velocidade de rolagem do texto, em milissegundos.
- **scrolldelay**: define a quantidade de pixels a percorrer em cada quadro.

```
<marquee behavior = "alternate"
              direction = "right"
              loop = 5
              scrollamount = 100
              scrolldelay = 5>
texto em movimento
</marquee>
```

11 - O Parâmetro class

Especifica um ou mais nomes de classe para um elemento. Refere-se a uma classe em CSS, sendo utilizada para configurar um elemento em diversas tags diferentes. Na versão HTML 4.01, o atributo de classe não pode ser usado com as tags <base>, <head>, <html>, <meta>, <script>, <style> e <title>. No HTML5, o atributo de classe pode ser usado em qualquer elemento. Pode-se definir como uma classe específica de uma tag:

```
<style>
H1.intro {
    color: blue;
}
</style>
```

Pode-se definir como uma classe independente:

```
<style>
.importante {
    background-color: orange;
}
</style>
```

Na página web, utiliza-se a classe como parâmetro das tags:

```
<H1 class="intro">Introdução</H1>
<P class="importante">Isso é importante!</P>
<H1 class="intro importante">Tudo junto</H1>
```

PARTE 5 - Âncoras

11 - Âncoras

As âncoras são utilizadas para fazer ligações entre partes de uma página web, através da tag `<A>` com seu parâmetro **name**. Para isso, é preciso determinar um ponto de referência na página, que será o destino a ser buscado pelo *link*. Este ponto de referência é a âncora e receberá um nome que será mencionado pelo *link* que o acessará. É bastante útil para se movimentar por textos longos na página web, como nos celulares por exemplo.

11.1 - Definindo uma Âncora

Até a versão HTML 4.01 era possível definir uma âncora com a tag `<A name>...` no ponto de destino na página web. O nome deve ser definido sem o símbolo # e o texto entre as marcas de início e fim não aparecem na tela.

```
<a name="inicio">Início da Página</a>
```

11.2 - Link para acessar a Âncora

Utiliza-se um texto ou uma figura que, ao ser clicada, fará ligação direta com a âncora que foi definida, fazendo com que a página web “seja movida” até a âncora. A tag usada para criar um *link* é `<A href>...`. O nome da âncora deve ser referenciado com o símbolo #, indicando que o destino é uma âncora na página atual.

```
<a href="#inicio">
Voltar ao Início da Página</a>
```

11.3 - Link para acessar Âncora Externa

Para acessar uma âncora externa, ou seja, definida em uma outra página web, basta referenciar a URL da página de destino imediatamente antes do símbolo # descrito no parâmetro **href** da tag `<A>`.

```
<a href="pagina2.htm#inicio">
Voltar ao Início da Página 2</a>
```

PARTE 6 - Tabelas

12 - Utilizando Tabelas

A utilização de tabelas melhora muito a aparência de uma página web, mas requer muita atenção, paciência e testes contínuos, para que se obtenha um bom resultado. As tags são simples, mas a estruturação delas pode ser bastante complexa, dependendo da aplicação requerida:

```
<TABLE>
<CAPTION>Título da Tabela</CAPTION>
<TR>
    <TD>...</TD>
    <TH>...</TH>
</TR>
</TABLE>
```

Até a versão HTML 4.01 os parâmetros abaixo eram utilizados. Na versão HTML 5 nem todos os parâmetros podem ser utilizados, sendo necessária a formatação com estilos em CSS.

- a. A tag `<TABLE>...</TABLE>` define o início e o fim de uma nova tabela, com os seguintes parâmetros:
- **align** = alinhamento horizontal (left, center, right).
 - **bgcolor** = cor do fundo.
 - **border** = 1 ou 0.
 - **cellspacing** = espaço em pixels entre as células.
 - **cellpadding** = espaço em pixels entre o conteúdo e a borda da célula.
 - **width** = comprimento da tabela em pixels ou %.

No HTML 5 pode-se utilizar `<style>` em CSS.

```
<HEAD>
<style>
table {
    display: table;
    border-collapse: separate;
    border-spacing: 2px;
    border-color: gray;
}
</style>
</HEAD>
```

- b. A tag `<TR>...</TR>` inclui uma linha na tabela, mas não recebe conteúdo.
- **align** = alinhamento horizontal (left, center, right).
 - **valign** = alinhamento vertical (top, middle, bottom).
 - **bgcolor** = cor do fundo.

No HTML 5 pode-se utilizar `<style>` em CSS.

```
<HEAD>
<style>
tr {
    display: table-row;
    vertical-align: inherit;
    border-color: inherit;
}
</style>
</HEAD>
```

c. Os conteúdos da tabela devem ser inseridos nas células que são definidas pelas tags `<TD>...</TD>` e `<TR>...</TR>`. A diferença é que `<TR>` exibe os conteúdos com alinhamento centralizado e em negrito.

- **colspan** = qtde. de colunas mescladas
- **rowspan** = qtde. de linhas mescladas
- **align** = alinhamento horizontal (left, center, right).
- **valign** = alinhamento vertical (top, middle, bottom).
- **bgcolor** = cor do fundo.
- **border** = 1 ou 0.
- **width** = comprimento da célula em pixels ou %.
- **height** = comprimento da célula em pixels ou %.
- **cellspacing** = espaço em pixels entre as células.
- **cellpadding** = espaço em pixels entre o conteúdo e a borda da célula.
- **nowrap**, se declarado, não permite quebrar o texto.

No HTML 5 pode-se utilizar `<style>` em CSS.

```
<HEAD>
<style>
td {
    display: table-cell;
    vertical-align: inherit;
}
</style>
</HEAD>
```

d. A tag `<CAPTION>...</CAPTION>` inclui um título que será exibido centralizado e acima da tabela. Deve ser declarado imediatamente após a tag `<TABLE>`.

- **align** = alinhamento horizontal (left, center, right).

No HTML 5 pode-se utilizar `<style>` em CSS.

```
<HEAD>
<style>
caption {
    display: table-caption;
    text-align: center;
}
</style>
</HEAD>
```

12.1 - Tabela com Células Simples

ABC Paulista

Cidade	Área (Km ²)	População (2010)
Santo André	175	568.538
São Bernardo do Campo	406	736.466
São Caetano do Sul	15,33	149.263

```
<TABLE border=1 width=100%>
<caption>ABC Paulista</caption>
<TR>
    <TH>Cidade</TH>
    <TH>Área (Km<sup>2</sup></TH>
    <TH>População (2010)</TH>
</TR>
<TR>
    <TD>Santo André</TD>
    <TD>175</TD>
    <TD>568.538</TD>
</TR>
<TR>
    <TD>São Bernardo do Campo</TD>
    <TD>406</TD>
    <TD>736.466</TD>
</TR>
<TR>
    <TD>São Caetano do Sul</TD>
    <TD>15,33</TD>
    <TD>149.263</TD>
</TR>
</TABLE>
```

12.2 - Tabela com Células Mescladas

Times do Futebol Paulista

Palmeiras		São Paulo
Corinthians	Portuguesa	São Caetano
Santos	Santo André	

```
<TABLE border=2 width=100% bgcolor="yellow">
<CAPTION>Times do Futebol Paulista</CAPTION>
<TR>
    <TH colspan=2 bgcolor=green>Palmeiras</TH>
    <TD height=50>São Paulo</TD>
</TR>
<TR>
    <TD valign=bottom height=50>Corinthians</TD>
    <TH align=right>Portuguesa</TH>
    <TH rowspan=2>São Caetano</TH>
</TR>
<TR>
    <TD height=50 valign=top align=center>
        Santos</TD>
    <TD bgcolor=blue><font color=white>Santo
        André</font></TD>
</TR>
</TABLE>
```

[Editar o EXERCÍCIO 08 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 09 do Caderno de Exercícios](#)

Referências

<https://en.wikipedia.org>
<https://www.w3schools.com>
<http://www.tutorialspark.com>