





**Apostila de  
Lógica de Programação,  
HTML e JavaScript**

ESTA OBRA PODE SER REPRODUZIDA E DISTRIBUÍDA PARCIAL OU  
INTEGRALMENTE DESDE QUE CITADA A FONTE.  
MATERIAL COPYLEFT - VENDA PROIBIDA

Todo material desenvolvido pela Coordenadoria do Governo  
Eletrônico é resultado de um processo coletivo de produção,  
que se iniciou em 2001 e que é permanente.  
Agradecemos a todos que colaboraram e que queiram contribuir.

CGE COORDENADORIA DO GOVERNO ELETRÔNICO  
Equipe de Treinamento Técnico

Aparecido Quesada  
Adriana Tosta  
Eder Moura Dourado  
Simone Leal dos Santos  
Thyago Akira de Moraes Ribeiro  
Yuri Robinson de Souza

Contato  
treinamento\_cge@prefeitura.sp.gov.br  
telecentros@prefeitura.sp.gov.br

PALÁCIO DO ANHANGABAÚ  
VIADUTO DO CHÁ Nº 15  
CEP 01002-000 SÃO PAULO  
TEL: 3113-8938 FAX 3113-8939

## Índice

1) Lógica de Programação .....	pg. 11
Algoritmo .....	pg. 13
Exercícios .....	pg. 13
Constantes e Variáveis .....	pg. 15
Exercício .....	pg. 16
Fluxogramas .....	pg. 17
Exercício .....	pg. 18
Operadores .....	pg. 19
Operadores Matemáticos .....	pg. 19
Operadores Relacionais .....	pg. 19
Operadores Lógicos .....	pg. 20
Exercícios .....	pg. 22
Estrutura de Controle .....	pg. 27
Seqüência simples .....	pg. 27
Seleção .....	pg. 27
Exercícios .....	pg. 29
Repetição .....	pg. 33
Enquanto ... Faça: .....	pg. 33
Até que ... Faça: .....	pg. 34
Faça Enquanto ...: .....	pg. 34
Faça Até que ...: .....	pg. 34
Para ... até ... faça .....	pg. 35
Exercícios .....	pg. 35
Lógica Escrita .....	pg. 37
Exercícios .....	pg. 38
2) HTML .....	pg. 39
Introdução .....	pg. 41
Redes .....	pg. 41
Provedores de Acesso .....	pg. 41
Internet .....	pg. 41
Como surgiu a Internet .....	pg. 42
Endereço IP .....	pg. 42
Domínio .....	pg. 42
WWW .....	pg. 43

O que são Browsers? .....	pg. 44
Protocolo HTTP .....	pg. 44
URL .....	pg. 45
Páginas Web .....	pg. 45
Home Page .....	pg. 45
Como Planejar seu Site .....	pg. 46
Determinando a Finalidade de um Site .....	pg. 46
O que é HTML? .....	pg. 47
O que você pode fazer com o HTML? .....	pg. 47
O que é necessário para trabalhar com HTML? .....	pg. 48
Sintaxe dos comandos HTML .....	pg. 48
Como Exibir o Código HTML de uma Página da Internet ....	pg. 49
Estrutura Básica do HTML .....	pg. 49
Elementos Básicos do HTML .....	pg. 49
Atributo BACKGROUND .....	pg. 50
Atributo BGCOLOR: .....	pg. 50
Atributo TEXT .....	pg. 50
Atributos LINK, ALINK, VLINK .....	pg. 50
Títulos e Subtítulos .....	pg. 51
Exercício .....	pg. 51
Acentuação .....	pg. 52
Formatação de Textos .....	pg. 53
Comando <font> .....	pg. 53
Os principais Comandos de Estilo de Texto são: .....	pg. 54
Comando <marquee> .....	pg. 55
Exercício .....	pg. 55
Formatação de Parágrafos .....	pg. 56
Comando <P ALIGN=...> .....	pg. 56
Comando <CENTER> .....	pg. 57
Comando <BLOCKQUOTE> .....	pg. 57
Linhas Horizontais .....	pg. 57
Hyperlinks .....	pg. 57
Definir um Link Dentro de uma Página .....	pg. 58
Definir um Link para uma Página Interna .....	pg. 59
Definir um Link "Mailto" .....	pg. 59
Definir um Link para Outro Site .....	pg. 60

	Definir um Link em uma Imagem .....	pg. 60
Imagens	.....	pg. 60
Principais Diferenças entre as Imagens GIF e JPEG .....		pg. 61
Qual Formato Escolher? .....		pg. 61
GIFs Animados .....		pg. 62
Definir uma Imagem no Segundo Plano .....		pg. 62
	O comando <! ... —> .....	pg. 63
	Exercício .....	pg. 63
Listas	.....	pg. 64
	Listas Ordenadas .....	pg. 65
	Exercício .....	pg. 65
	Listas Não Ordenadas .....	pg. 66
	Exercício .....	pg. 67
	Lista de Definição .....	pg. 68
	Exercício .....	pg. 68
Tabelas	.....	pg. 69
	Elementos de uma Tabela .....	pg. 70
	Comando <CAPTION> .....	pg. 70
	Comando <TR> .....	pg. 70
	Comando <TD> .....	pg. 70
	Comando <TH> .....	pg. 70
	Exercício .....	pg. 71
Parâmetros	.....	pg. 72
	Parâmetros do Elemento <TABLE> .....	pg. 72
	border .....	pg. 72
	width .....	pg. 73
	cellspacing .....	pg. 73
	cellpadding .....	pg. 73
	align .....	pg. 73
	bgcolor .....	pg. 74
	Exercício .....	pg. 74
	Parâmetros do Elemento <TR> .....	pg. 74
	align .....	pg. 74
	valign .....	pg. 75
	bgcolor .....	pg. 75
	Parâmetros dos Elementos <TD> e <TH> .....	pg. 75

align .....	pg. 75
valign .....	pg. 75
bgcolor .....	pg. 75
nowrap .....	pg. 75
colspan .....	pg. 76
rowspan .....	pg. 76
Exercício .....	pg. 76
Formulários .....	pg. 78
Elementos HTML para Formulários .....	pg. 78
<form> .....	pg. 78
<input> .....	pg. 79
Tipos do Elemento INPUT .....	pg. 79
<input type=text> .....	pg. 79
<input type=password> .....	pg. 80
<input type=radio> .....	pg. 80
<input type=checkbox> .....	pg. 80
<input type=reset> .....	pg. 81
<input type=submit> .....	pg. 81
<input type=image> .....	pg. 82
<input type=hidden> .....	pg. 82
<select> .....	pg. 82
<textarea> .....	pg. 83
Exercício .....	pg. 83
Hospedando seu Site .....	pg. 85
Sites de Hospedagem Gratuita .....	pg. 86
Sites de Hospedagem Paga .....	pg. 86
Hospedando seu Site no Geocities .....	pg. 86
3) Java Script .....	pg. 88
Operadores Lógicos .....	pg. 90
Operadores Matemáticos .....	pg. 90
Controles Especiais .....	pg. 91
Comandos Condicionais .....	pg. 91
Comando IF: .....	pg. 92
Comando FOR: .....	pg. 92
Comando WHILE: .....	pg. 92



Eventos .....	pg. 93
Criando Variáveis .....	pg. 94
Escrevendo no Documento .....	pg. 94
Mensagens .....	pg. 95
Apenas Observação: .....	pg. 95
Mensagem que retorna confirmação de OK ou CANCELAR: .....	pg. 95
Recebe mensagem via caixa de texto Input .....	pg. 95
Criando Funções .....	pg. 96
Interagindo com o usuário .....	pg. 97
Objeto Input TEXT .....	pg. 97
Objeto Input PASSWORD .....	pg. 97
Objeto Input CHECKBOX .....	pg. 98
Objeto Input RADIO .....	pg. 98
Objeto Input BUTTON .....	pg. 98
Objeto Input SUBMIT .....	pg. 99
Objeto TEXTAREA .....	pg. 99
Objeto SELECT .....	pg. 99
Abrindo novas janelas .....	pg. 99
Abrindo um Documento .....	pg. 100
Escrevendo no Documento .....	pg. 100
Fechando a Janela .....	pg. 100
Fechando o Documento .....	pg. 100



PARTE I  
**Lógica de Programação**



Esta apostila visa auxiliar àqueles que pretendem ingressar na área de programação e desenvolvimento de sistemas. Aprenderemos como ordenar ações de forma consistente e assim formar uma seqüência lógica.

Serão demonstrados conceitos generalizados, que poderão ser aplicados nos programas se forem devidamente adaptados.

## **Algoritmo**

O Algoritmo é a parte fundamental para qualquer programação. Ele é uma seqüência lógica e predefinida para a realização de ações. Mas afinal, o que é isso?

Quando estamos com fome, executamos várias tarefas para poder se alimentar:

Alguém está faminto então:

Vai até a cozinha;

Abre a geladeira;

Prepara a refeição;

Almoça.

O exemplo acima são as ações que se devem seguir para saciar a fome. É o algoritmo de como acabar a fome de alguém.

A construção de um algoritmo deve seguir certas regras, para que a tarefa seja concluída com êxito. Devemos ser simples e objetivo, pois quem interpretar este código não poderá ter dúvidas para executá-lo.

## **EXERCÍCIOS**

1- Crie um algoritmo das rotinas que você executa para realizar a tarefa de acordar.

2- Crie um algoritmo das rotinas que você executa para procurar um emprego.

3- Crie um algoritmo das rotinas que você executa para se deslocar para este treinamento.

### **Constantes e Variáveis**

Em um algoritmo, sempre haverá alguém que estará executando a tarefa. Por exemplo:

Thiago compra o jornal;  
Alguém assiste televisão;  
Ana veste o vestido;

Nos exemplos acima, as ações são executadas por alguém, ou específico (Thiago e Ana), ou indeterminado (Alguém).

Quando a ação é executada por alguém específico, esse alguém é denominado constante. Isso quer dizer que ele não irá mudar.

Quando a ação é executada por alguém indeterminado, esse alguém é denominado variável. Isso quer dizer que essa variável pode tomar várias formas diferentes. No exemplo acima:

Alguém assiste televisão. (José assiste televisão; Maria assiste televisão).

As variáveis lhe proporcionam uma infinidade de possibilidades. Por isso que elas são classificadas de forma a restringir a sua variação.

As variáveis podem ser:

Numéricas (1,2,45,98);

Caracteres (a,B,F,z,Y);

Alfanuméricas (adc, eduardo, 5d6f);

Booleana ou Lógica (Verdadeiro, Falso);

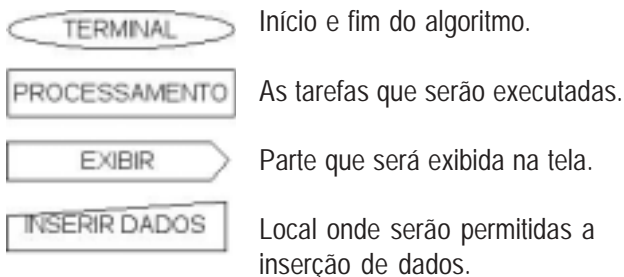
## EXERCÍCIO

1- Selecione dos algoritmos dos exercícios anteriores alguns itens que podem ser classificados como constantes ou como variáveis. Classifique-os de acordo com o tipo.

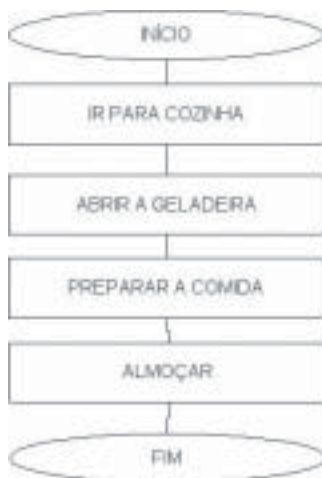


## Fluxogramas

Os fluxogramas têm o objetivo de facilitar o entendimento do algoritmo. Estes fluxogramas são padronizados em diferentes formas, que auxiliam na visualização de sua respectiva função:



Existem outros objetos no fluxograma, mas isso nós veremos futuramente. Agora, montaremos o fluxograma da ação comer.



Com esse fluxograma, você percebe que só poderá executar uma ação após executar a outra.

## EXERCÍCIO

- 1- Crie o fluxograma dos algoritmos dos exercícios anteriores.

## Operadores

### OPERADORES MATEMÁTICOS

Sempre serão utilizados cálculos para a realização de inúmeras tarefas. Os principais operadores matemáticos são:

+ (soma)  
- (subtração)  
\* (multiplicação)  
/ (divisão)

É muito importante saber que, como na matemática, as operações na lógica de programação seguem uma ordem. Existe uma ordem de precedência entre estes operadores:

Primeiro: ( ) - Parênteses.

Segundo: \*(multiplicação) ou /(divisão), o que aparecer primeiro.

Terceiro: +(soma) ou -(subtração), o que aparecer primeiro.

Exemplo:

$5 + 2 * 3 = 11$  , pela precedência, lê-se o mesmo que:  $5 + (2 * 3)$

$(5 + 2) * 3 = 21$  .

### OPERADORES RELACIONAIS

Os operadores relacionais são utilizados para comparar dois valores, que podem ser variáveis ou constantes, do mesmo tipo. São exemplos de operadores relacionais:

= (igual a)  
> (maior que)

< (menor que)  
<> (diferente de)  
>= (maior ou igual a)  
<= (menor ou igual a)

Esses operadores são muito utilizados. Com o resultado destas comparações implica num valor lógico (verdadeiro ou falso):

10 >= 5 (verdadeiro)  
11 <> 11 (falso)  
0,5 < (-1)(falso)  
A = A (verdadeiro)

## OPERADORES LÓGICOS

Os operadores lógicos são combinados às expressões. Desta forma, conseguimos um valor verdadeiro ou falso. Os operadores lógicos principais são:

E - conjunção  
Ou - disjunção  
Não - negação

O "E" e o "Ou" são operadores binários. Isso quer dizer que eles servem para combinar duas expressões. O "Não" é um operador unário. Isso quer dizer que ele serve para alterar o valor de uma expressão. Para entendermos melhor como eles são utilizados, vejam as seguintes expressões:

- 1.Tem energia elétrica.
- 2.O computador está ligado na tomada

Para que o computador ligue, ele tem que obedecer as duas expressões anteriores.

Imagine a seguinte relação:

(Expressão 1)      E      (Expressão 2)

Se uma destas expressões não for verdadeira, a condição é falsa (O computador não liga). O computador somente ligará se as duas forem verdadeiras. A tabela de possibilidades de uma condição é chamada Tabela Verdade:

Expressão 1	<i>E</i>	Expressão 2
FALSO	<b>FALSO</b>	FALSO
FALSO	<b>FALSO</b>	VERDADEIRO
VERDADEIRO	<b>FALSO</b>	FALSO
VERDADEIRO	<b>VERDADEIRO</b>	VERDADEIRO

Agora veja estas expressões:

- 1.Tenho dinheiro no banco.
- 2.Tenho cartão de crédito.

Para efetuar uma compra, não é necessário seguir as duas expressões acima.

Imagine a relação abaixo:

(Expressão 1)      Ou      (Expressão 2)

Agora, se uma destas expressões for verdadeira, a condição é verdadeira (Posso comprar). Se as duas forem falsas, a condição é falsa (Não posso comprar). Veja a tabela verdade desta relação:

Expressão 1	<i>Ou</i>	Expressão 2
VERDADEIRO	<b>VERDADEIRO</b>	VERDADEIRO
VERDADEIRO	<b>VERDADEIRO</b>	FALSO
FALSO	<b>VERDADEIRO</b>	VERDADEIRO
FALSO	<b>FALSO</b>	FALSO

Desta forma, já temos conhecimento das cadeias verdades dos dois operadores lógicos principais. Ainda existe o operador unário. Veja as seguintes expressões:

1. Não tenho dinheiro.
2. Tenho dinheiro.

Se eu negar as afirmações acima, quais serão os resultados:

1. Negar que não tenho dinheiro => Tenho dinheiro.
2. Negar que tenho dinheiro => Não tenho dinheiro.

Conseguiremos a seguinte tabela verdade:

Não	Expressão
FALSO	VERDADEIRO
VERDADEIRO	FALSO

Os operadores lógicos podem ser combinados de inúmeras formas. Como os operadores matemáticos, os operadores lógicos seguem uma ordem de precedência (prioridade):

- Primeiro: ( ) - Parênteses.  
Segundo: Não - Negação.  
Terceiro: E - Conjunção.  
Quarto: Ou - Disjunção.

## EXERCÍCIOS

1- Considere a tabela a seguir:

Torcedor	Corintianos		Palmeirenses		São Paulinos		Santistas	
Sexo	M	F	M	F	M	F	M	F
Quantidade	40	10	30	5	36	8	25	2

a) Crie uma afirmação que tenha como resultado todos os torcedores palmeirenses.

Torcedor=Palmeirense

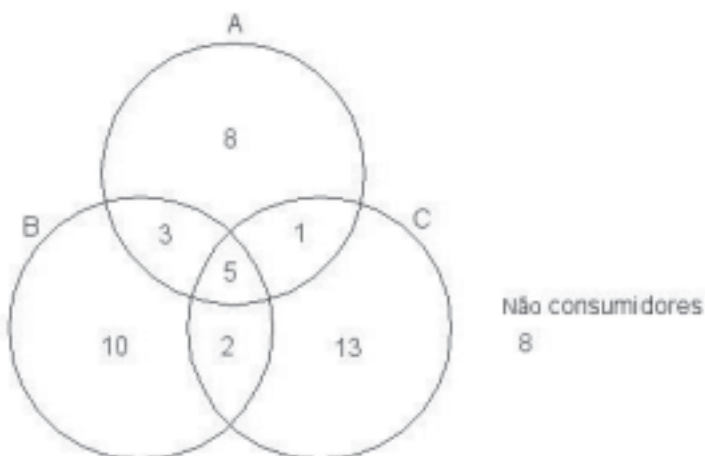
Resultado: 35 torcedores

b) Crie uma afirmação que tenha como resultado todas as torcedoras mulheres corintianas, palmeirenses e santistas.

c) Crie uma afirmação que tenha como resultado todos os torcedores de times auvi-negros.

2- Considere o diagrama a seguir:

Pesquisa qual chocolate você consome?



a) Crie uma afirmação que tenha como resultado dos consumidores das 3 marcas de chocolate.

Chocolate=A E Chocolate=B E Chocolate=C  
Resultado: Consumidores=5

b) Crie uma afirmação que tenha como resultado dos consumidores de duas marcas de chocolate.

c) Crie uma afirmação que tenha como resultado todos os consumidores de apenas uma marca de chocolate (ou somente de A ou somente de B ou somente de C).

d) Crie uma afirmação que tenha como resultado os consumidores do chocolate A ou do chocolate B e os consumidores que comem os 3 chocolates.



e) Crie uma afirmação que tenha como resultado todos os pesquisados.



## Estrutura de Controle

Existem três estruturas básicas de controle, que são:

- Sequência simples
- Seleção
- Repetição

As ações são efetuadas com base nestas três estruturas.

### SEQUÊNCIA SIMPLES

É uma tarefa que sempre será executada. Ela obedecerá a ordem da esquerda para direita, de cima para baixo.

Um exemplo de sequência simples é o utilizado anteriormente:

Vai até a cozinha;  
Abre a geladeira;  
Prepara a refeição;  
Almoça.

### SELEÇÃO

Esta estrutura é utilizada quando há uma necessidade de escolha entre duas opções.

Por exemplo, quando perguntamos se você está com fome, você responde sim ou não. Esta decisão acarretará na execução ou não de algumas tarefas.

A estrutura de seleção utiliza a forma "Se..... então", ou "Se ..... então..... senão". Para entendermos melhor isto, veja o exemplo:

Se você está com fome então você come.

Esta é uma forma simples de exemplificar uma seleção. Caso você esteja com fome, você come. Senão, você não come.

Se a afirmação for dividida em expressões:

- 1.Você está com fome.
- 2.Você come.

Então a tabela verdade dela seria a seguinte:

Se (Expressão 1)	Então (Expressão 2)	Resultado
V	V	Você come (V)
V	F	Você não come(F)
F	V	Você come (V)
F	F	Você não come(V)

Ainda existe a possibilidade de colocar outra opção:

Se você está com fome você come, senão você dorme.

Nesta afirmação, você pode separá-la em três expressões:

- 1.Você está com fome
- 2.Você come
- 3.Você dorme

Esta condição implica na tabela verdade abaixo:

Se (Exp 1)	Então (Exp 2)	Senão (Exp 3)	Resultado
V	V	V	Você come(V)
V	V	F	Você come (V)
V	F	V	Você dorme (F)
V	F	F	Você não come nem dorme (F)
F	V	V	Você dorme (V)
F	V	F	Você não come nem dorme (V)
F	F	V	Você dorme (V)
F	F	F	Você não come nem dorme (V)

Agora veremos um fluxograma de uma estrutura de seleção:



Este é um exemplo de um fluxograma com uma seleção composta.

#### EXERCÍCIOS

1- Crie uma condição para uma pessoa atravessar a rua. Considere o farol como uma variável.

<i>Se farol=verde faça</i> Atravesse a rua; <i>Senão</i> Aguarde o farol verde;	<i>Se farol=vermelho faça</i> Aguarde o farol verde; <i>Senão</i> Atravesse a rua
--	--

2- Crie uma condição para uma pessoa entrar na sua casa.

3- Veja a seguinte tabela de preços:

Gabinete	- R\$ 1.000,00
Monitor	- R\$ 400,00
Teclado	- R\$ 50,00
Mouse	- R\$ 30,00
Caixa de Som	- R\$ 50,00
Impressora	- R\$ 400,00

a) Baseando-se na tabela acima, crie uma condição para uma pessoa que pretende comprar as peças de um computador, obedecendo a ordem de importância (Gabinete, Monitor, Teclado, Mouse, Caixa de Som e Impressora).

b) Usando a condição criada acima, veja o que comprará cada uma das pessoas:

João - R\$1.500,00;  
Paulo - R\$1.000,00;  
Pedro - R\$850,00;  
Mário - R\$200,00;  
Eduardo - R\$2.000,00.

4- Crie o fluxograma dos exercícios anteriores:



## REPETIÇÃO

As estruturas de repetição servem para realizar uma rotina inúmeras vezes. Essa repetição pode ser finita ou infinita (looping infinito). Deve-se tomar muito cuidado com os comandos de repetição, pois executando um looping infinito você pode travar seu computador.

Enquanto ... Faça:

Este comando é conhecido como "While (condição) Do". Esta repetição testa a condição antes de executar o looping. Vejamos o exemplo:

```
Enquanto Dinheiro>10 faça:  
    Compre pão;  
    Compre leite;  
    Compre ovos.
```

Neste exemplo, temos uma variável (Dinheiro), que é testada antes, pois se não houver dinheiro suficiente, não será possível

efetuar a compra. Então, enquanto a condição for verdadeira, será feita a compra.

Até que ... Faça:

Este comando é conhecido como "Until (condição) Do". Esta repetição é muito parecida com a anterior. Vejam o exemplo:

Até que Dinheiro<=10 faça:

    Compre pão;

    Compre leite;

    Compre ovos.

Neste exemplo temos a mesma variável (Dinheiro), que também é testada antes. A única diferença do exemplo anterior é que o looping só será efetuado enquanto a condição for falsa.

Faça Enquanto ...:

Este comando é conhecido como "Do.... While (condição)". Esta repetição testa a condição após ter executado uma vez o looping. Vejam o exemplo:

Faça

    Dar uma colherada;

Enquanto Prato<>Vazio.

Neste exemplo, temos a variável prato, que é testada após ser efetuada uma vez a ação (Dar uma colherada). Enquanto a condição citada for verdadeira, será efetuada a repetição.

Faça Até que ...:

Este comando é conhecido como "Do... Until (condição)". Esta repetição também testa a condição após ter efetuado uma vez os comandos. Vejamos o exemplo:

Faça  
    Dar uma colherada;  
Até que Prato=Vazio.

Da mesma forma que o exemplo anterior, a variável prato é testada após a ação. Já neste exemplo, o looping é efetuado enquanto a condição for falsa.

Para ... até .... faça:

Este comando é conhecido como "For (variável) to (valor) do". Utilizamos esta repetição quando precisamos que o looping seja executado "X" vezes, isto é, quando há um número que determina a quantidade de loopings. Veja o exemplo:

Para Degrau=0 até 10 faça:  
    Suba um degrau;

Este exemplo utiliza-se de uma variável, que recebe seu valor inicial (degrau=0) e do objetivo a ser alcançado (10). Desta forma, a cada repetição é efetuado o incremento de uma unidade na variável.

## EXERCÍCIOS

1- Crie uma repetição para a ação de comer:

Enquanto prato<>vazio faça: Dê uma colherada;	Até prato=vazio faça: Dê uma colherada;
Faça: Dê uma colherada;	Faça: Dê uma colherada;
Enquanto prato<>vazio.	Até prato=vazio.

2- Crie uma repetição para a ação de lavar louça:

3- Você tem uma venda de frutas. Você tem 50 frutas para vender. Crie uma condição para você vender estas frutas.

4- Após você ter aberto a venda, apareceram os seguintes clientes e a quantidade de frutas que cada um desejava:

João .....	8
Manoel .....	10
Joaquim .....	4
Maria .....	4
Julio .....	11
Mário .....	9
Patricia .....	3
Carolina .....	7
Ana .....	6

Considerando que você só fecharia a venda após vender todas a frutas, após qual cliente você foi embora?

## Lógica Escrita

Agora trabalharemos utilizando-se dos fundamentos da lógica. A lógica tem uma simbologia própria para os operadores lógicos:

**Ou:**  $\vee$  (disjunção)      **E:**  $\wedge$  (conjunção)      **Não:**  $\neg$  (negação)

Também trabalharemos as variáveis de uma forma diferente. Será atribuídas a elas uma letra ou uma abreviatura, para simplificar a visualização.

Desta forma, a visualização de uma expressão será a seguinte:

$A \wedge B$

$a \vee b$

Toda expressão tem sua tabela verdade:

A	$\wedge$	B
V	V	V
V	F	F
F	F	V
F	F	F

A	$\vee$	B
V	V	V
V	V	F
F	V	V
F	F	F

$\neg$	A
V	F
F	V

### EXERCÍCIOS

1- Transforme as afirmações criadas nos exercícios de operadores lógicos na Lógica Escrita e crie uma tabela verdade para as expressões criadas.

Exercício1: Torcedor=T      T= palmeirense

Exercício 2: Chocolate=Ch       $(Ch=A) \wedge (Ch=B) \wedge (Ch=C)$

PARTE II  
**HTML**





## Introdução

Neste curso apresentaremos a linguagem utilizada na definição de páginas na Internet. Com o **HTML (Hyper Text Markup Language)** é possível criar páginas incríveis, conforme a criatividade de cada um. Além disso, você aprenderá tudo a respeito da World Wide Web, dos browsers e das homepages.

## Redes

Uma rede é a conexão entre dois ou mais computadores. Através da rede, os computadores podem compartilhar periféricos, informações e programas. Uma impressora, por exemplo, pode ser utilizada por vários equipamentos ao mesmo tempo, reduzindo assim os custos.

Os provedores da Internet são um exemplo de rede, constituída pela interligação entre computadores e um servidor central, através de uma linha telefônica.

## Provedores de Acesso

Os **Provedores de Acesso** são empresas que mantêm computadores conectados de forma permanente à Internet. O provedor faz um investimento em linhas telefônicas, computadores, software e na própria conexão permanente com a Internet. Depois provê acesso à Internet aos usuários. Assim o usuário passa também a ter acesso à Internet pagando os pulsos telefônicos das chamadas, e as mensalidades dos provedores em alguns casos.

## Internet

A Internet é uma rede composta por milhões de redes interconectadas mundialmente, compreendendo redes comerciais, acadêmicas, governamentais e militares. Cada rede individual conectada à Internet pode ser administrada por uma

entidade governamental, uma empresa ou uma instituição educacional. Mas a Internet, como um todo, não tem um poder centralizado, ou seja, ninguém é dono da Internet.

## **Como surgiu a Internet**

A Internet surgiu em **1969**. Vivia-se o auge da guerra fria e os laboratórios militares americanos sentiam a necessidade de compartilhar de forma segura informações sigilosas, armazenadas em computadores espalhados pelo país. Foi criada então uma rede de comunicação interligando esses computadores. Para evitar que um ataque nuclear soviético interrompesse essa comunicação, desenvolveram um esquema para a transmissão em que as informações seriam divididas em pacotes. Esses pacotes tomariam caminhos diferentes para chegar ao mesmo local. Caso um trecho de comunicação fosse destruído, os pacotes pegariam outro caminho e chegariam ao mesmo destino.

## **Endereço IP**

Nesse momento você deve estar se perguntando como tantos computadores conseguem se comunicar uns com os outros. Cada um consegue ser exclusivo, porque cada um tem seu próprio endereço. Os endereços IP (Internet Protocol/Protocolo Internet) são formados por números de 4 bytes. São valores únicos e cada um indica um computador na Internet.

Ex: 200.225.162.12

## **Domínio**

Se um número for muito extenso teremos dificuldade em memorizá-lo. A partir dessa dificuldade, os idealizadores da Internet criaram o conceito de "Domínio". Domínio é uma tabela na qual

cada endereço IP é associado a um nome. Dessa forma, o internauta não precisa ficar preocupado se não lembrar do endereço de um determinado site, basta mandar o navegador ou browser procurar pelo domínio.

Existem algumas regras para a atribuição de domínios, como o uso de letras minúsculas, a proibição de sinais gráficos como +, - ou \* e a extensão final, que deve refletir o caráter da instituição.

Temos as seguintes extensões:

<b>.mil</b>	Organizações Militares
<b>.gov</b>	Organizações Governamentais
<b>.com</b>	Empreendimentos Comerciais ou Pessoais
<b>.edu</b>	Instituições de Ensino
<b>.net</b>	Redes Mundiais
<b>.org</b>	ONGs e semelhantes

Os nomes de domínios são solicitados através de uma organização chamada INTERNIC, que coordena todos os pedidos, verificando se já não há registro desse domínio.

Em todos os países, com exceção dos EUA, ficou definido que os domínios da Internet terminam com a sigla do país em que está localizado. Assim, google.com.br é diferente de google.com.

## WWW

A **World Wide Web (WWW)** é a parte mais popular da Internet. Devido à sua facilidade de uso, logo se espalhou pelo mundo. A "Grande Teia Mundial" (World Wide Web) é formada por um conjunto sempre crescente de computadores. O que diferencia de uma rede convencional é o fato das informações estarem conectadas umas às outras, formando o conceito de **hipertexto**.

## O que são Browsers?

A ferramenta básica para a exibição de documentos na WWW são os **browsers**, ou como são conhecidos, os “**navegadores**”.

Com o passar do tempo os browsers foram ficando cada vez mais sofisticados, com inclusão de imagens, novos métodos de formatação de texto, reconhecimento de novos formatos de dados, capacidade de comunicação por voz via Internet, e recepção de vídeo e áudio.

Há dezenas de browsers diferentes. Os mais usados no Linux são o Mozilla e o Galeon, e no Windows é o Internet Explorer.

## Protocolo HTTP

Os **protocolos** são “línguas” usadas entre o computador que acessa a informação e o que a transmite. É uma forma de padronizar os processos; assim evita-se que cada programa use uma linguagem diferente.

O **HTTP**, isto é, o **Hiper Text Transfer Protocol**, Protocolo de Transferência de Hipertexto, é a linguagem usada na WWW. Através dele, qualquer browser pode se comunicar com qualquer servidor, independente dos fabricantes dos softwares envolvidos.

Os browsers mais modernos são compatíveis com vários tipos de protocolo, além do HTTP. Entre eles está o **FTP**, de **File Transfer Protocol**. É um método prático de ter acesso a arquivos em servidores de todo o mundo.

Para acessar um servidor de FTP via Mozilla, basta digitar, no campo de URL, o endereço na forma: **FTP://<nome do servidor de ftp>**.

## URL

**URL** significa **Universal Resource Locator** ou Localizador Universal de Recursos. A URL é um conjunto de informações que descreve um recurso qualquer na Internet, seja e-mail, FTP, telnet ou WWW. Assim, os programas sabem que protocolos devem utilizar e podem também orientar o usuário, quando o recurso solicitado estiver além das possibilidades do programa. Veja abaixo, exemplos de algumas URLs válidas:

`http://pobox.com/~cardoso`  
`ftp://ftp.unikey.com.br`  
`telnet:bbs.unikey.com.br`  
`mailto:cardoso@pobox.com`

## Páginas Web

Páginas Web ou documentos HTML ou mesmo páginas na Internet são documentos localizados em um servidor específico, mas acessíveis mundialmente, através da Internet. As páginas podem ser simples, com algumas linhas de texto, ou ter milhares de páginas num único site.

A publicação de uma página, ao contrário do que se pensa, não é complicada, tampouco dispendiosa.

Em uma página pessoal, você pode disponibilizar o que quiser: seus poemas, fotos do seu gato de estimação, desenhos dos filhos, comentários sobre o seu trabalho, etc.. O limite é sua imaginação e o espaço em disco.

## Home Page

**Home Page** é a página inicial ou a página de entrada de um site.

## Como Planejar seu Site

Projetar um site demanda um considerável planejamento prévio. Você não somente tem de decidir qual será seu público e o que incluir no site, mas terá de planejar como apresentar essa informação no site. Planejar seu site é o passo mais importante. Sem um bom plano, seu site poderá não ser bem-sucedido, o que significa, no pior dos casos, que ninguém irá visitá-lo. Fazer a si mesmo algumas perguntas importantes e desenvolver um plano baseado em suas respostas irá ajudá-lo a criar um site bem-sucedido.

### Determinando a Finalidade de um Site

Primeiro, você deve determinar a finalidade de seu site. Antes de começar a criar seu site, você deve fazer as seguintes perguntas:

1. Por que estou criando esse site? O que eu quero que as pessoas façam ou saibam após visitarem meu site?

É fundamental identificar o principal objetivo do site e depois manter essa meta em mente enquanto o projeta. Se estiver criando um site para negócios, certifique-se de obter a opinião de todos os envolvidos no projeto e nas decisões sobre o conteúdo.

2. Qual é o público que quero atingir?

Você deve decidir quem irá visitar seu site e depois planejar um site que atraia essas pessoas. Para focalizar seu público, faça a si mesmo algumas perguntas mais específicas. Qual é a formação de seus amigos, de sua família ou de seus clientes? Quais grupos etários estão representados? Quais são seus interesses? O que eles fazem?

3. Quais informações quero compartilhar com os visitantes de meu site?

Muitas informações estão disponíveis sobre você como pessoa, sua empresa ou organização, ou sobre seu tópico. Determine quais informações você quer tornar disponíveis aos seus visitantes. Seja muito cuidadoso ao determinar qual informação você não deseja compartilhar. A melhor maneira de decidir o que você quer no seu site é observar outros sites na Internet.

#### 4. Que tipo de informação você quer obter dos visitantes?

A Internet é uma grande fonte, tanto para dar como para receber informações. Você precisa decidir se apenas quer compartilhar informações com aqueles que vêm seu site ou se também quer obter informações deles, como por exemplo, opiniões sobre seu site ou informações dos clientes.

#### 5. Quanto tempo e dinheiro você pode gastar mantendo seu site?

Quanto mais complexo é o site, mais tempo e dinheiro ele consome em sua manutenção.

### O que é HTML?

**HTML** é a abreviação das palavras: **HyperText Markup Language**, que pode ser traduzido como Linguagem de Marcação de Hipertexto. **Não** é uma linguagem de programação. É uma linguagem de descrição/marcação de página.

**Hipertexto** é um método de organização que permite que textos, imagens, sons e ações fiquem interligados.

### O que você pode fazer com o HTML?

A função do HTML é criar páginas na Internet. Você pode criar textos nas páginas, fazer formatação, criar listas, criar tabelas, criar formulários, trabalhar com imagens, etc..

O HTML cria páginas estáticas, sem animação.

## O que é necessário para trabalhar com HTML?

Você irá precisar de:

1. Um editor de textos qualquer, ou você poderá usar o Composer do Mozilla.
2. E o browser (navegador) Mozilla.

Os arquivos devem ser salvos no formato texto puro (.txt) e com extensão .htm ou .html. Todos os arquivos de um projeto devem ser preferencialmente colocados dentro de uma mesma pasta (diretório).

## Sintaxe dos comandos HTML

Nos comandos são utilizadas duas marcas ou tags que são:

< > - Marca de abertura

</> - Marca de fechamento

<comando atributo="valor" atributo="valor">

Os atributos somente devem ser digitados na marca de abertura. As marcas de fechamento **não** aceitam atributos.

Exemplos de marcas:

<b> ... </b> - Coloca negrito no texto.

<center> ... </center> - Centraliza o texto na página.

<font> ... </font> - Permite definir o tipo, tamanho, cor, estilo da fonte.



## Como Exibir o Código HTML de uma Página da Internet

Toda página pode ter seu código fonte (HTML) exibido, bastando apenas seguir os passos abaixo.

1. Dar um duplo clique no ícone do Mozilla.
2. Acessar uma página qualquer.
3. Clicar no menu View e Page Source. Veja a figura abaixo.



## Estrutura Básica do HTML

A estrutura básica de um documento HTML é a seguinte:

```
<html>
<head>
<title> Título da Página </title>
</head>
<body>
...
</body>
</html>
```

## Elementos Básicos do HTML

<HTML> ... </HTML> - Indicam respectivamente o início e o fim de um documento HTML. Todos os outros elementos devem estar entre estas marcas.

**<HEAD> ... </HEAD>** - Delimitam a seção de cabeçalho do documento. Trata-se da primeira seção do documento.

**<TITLE> ... </TITLE>** - Indicam o título do documento, que será exibido na barra de título do navegador. Estas marcas devem constar da seção de cabeçalho.

**<BODY> ... </BODY>** - Representam o corpo do documento. Entre estas marcas estará contida a maior parte do conteúdo a ser exibido com textos e imagens. Este elemento pode conter cinco atributos opcionais, que são:

Atributo **BACKGROUND**: Especifica uma imagem como fundo da página.

Exemplo: `<body background="fundo.gif">`.

Atributo **BGCOLOR**: Configura a cor de fundo da página.

Exemplo: `<body bgcolor="white">`

Atributo **TEXT**: Configura a cor do texto da página.

Exemplo: `<body text="black">`

Atributos **LINK**, **ALINK**, **VLINK**: Configuram as cores dos links da página. **alink** configura a cor do link ativo, isto é, quando é clicado. **VLINK** configura a cor do link já visitado.

Exemplo: `<body link="blue" vlink="purple" alink="red">`

Este exemplo configura uma página com links azuis, links ativos em vermelho e links visitados em roxo.

Como o HTML **não** é uma linguagem de programação, você nunca será avisado de erros que tenha cometido na digitação ou na edição do seu documento. O simples esquecimento de uma barra

pode gerar efeitos colaterais inesperados na visualização de uma página, e estes efeitos serão o único sinal de que algo está errado.

## Títulos e Subtítulos

O destaque para títulos e subtítulos é importante, porque mostra ao leitor uma visão geral sobre o que trata o texto em questão.

Você pode utilizar até **seis níveis** de títulos, que são numerados de 1 (o maior) a 6 (o menor). São tags do tipo início-fim. Exemplos:

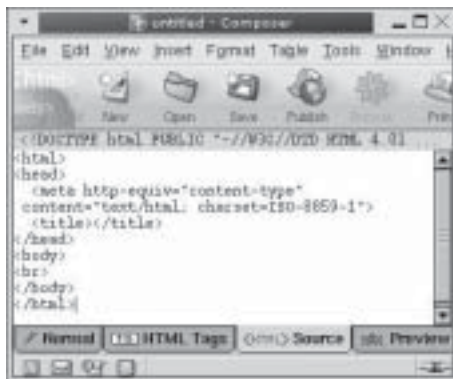
```
<h1> </h1>  
<h2> </h2>  
<h3> </h3>
```

### EXERCÍCIO

Abrir o Composer do Mozilla.

Roteiro:

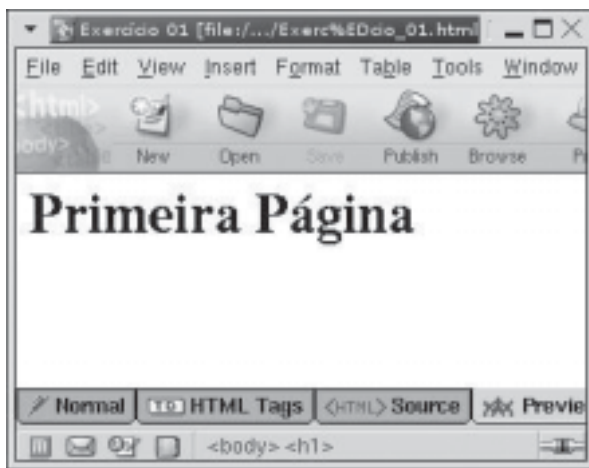
1. Abrir o Mozilla
2. Clicar no botão Composer, que fica localizado no canto inferior esquerdo da tela. É o terceiro botão da esquerda para direita.
3. Clicar na guia <HTML> Source. Veja a figura abaixo:



4. Digitar os comandos abaixo.

```
<html>
<head>
<title> Exercício 01</title>
</head>
<body bgcolor="white">
<h1> Primeira Página </h1>
</body>
</html>
```

Agora, vamos salvar no formato texto, com o nome `exercicio01.htm` e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Acentuação

Você não deve acentuar diretamente os documentos HTML, porque não é possível saber que tabela de caracteres o usuário dispõe, e nem saber a partir de qual equipamento a página será acessada. Como cada computador lida com os caracteres de uma forma diferente, os textos não devem ser acentuados.

O mecanismo para se solucionar esse impasse é a tabela de caracteres padrão desenvolvida pela ISO International Standards Organization. É uma organização que desenvolve e administra padrões. Veja os exemplos abaixo:

PARA CONSEGUIR	VOCÊ DEVE DIGITAR
Á	&Aacute;
Â	&Agrave;
Ã	&Atilde;
Ä	&Acirc;
Ç	&Ccedil;

### Formatação de Textos

Todos os comandos que alteram o estilo do texto são do tipo liga-desliga, ou seja, precisam ser especificados em pares, marcando o início e o fim do trecho que sofrerá a formatação.

Comando **<font face=... size=... color=...> ... </font>** - Define o tipo, o tamanho e a cor da fonte. Não é necessário usar todos os atributos juntos.

O atributo FACE define o tipo de letra (fonte).

O atributo SIZE define o tamanho da fonte. A relação dos tamanhos válidos são:

SIZE	
1	8 pt
2	10 pt
3	12 pt
4	14 pt
5	24 pt
6	36 pt
7	72 pt

O atributo **COLOR** especifica a cor do texto. Seu valor é especificado através de um nome predefinido de cores ou no formato hexadecimal. Veja abaixo, a tabela das principais cores:

NOME DA COR (INGLÊS)	NOME DA COR (PORTUGUÊS)	HEXADECIMAL
Beige	Bege	#F5F5D
Black	Preta	#000000
Blue	Azul	#0000FF
Cyan	Ciano	#00FFFF
Gold	Ouro	#FFD700
Green	Verde	#008000
Gray	Cinza	#808080
Yellow	Amarela	#FFFF00
Magenta	Magenta	#FF00FF
Maroon	Marrom	#800000
Olive	Verde oliva	#808000
Orange	Laranja	#FFA500
Red	Vermelha	#FF0000
Pink	Rosa	#FFC0CB
Purple	Púrpura	#800080
White	Branca	#FFFFFF

Os principais Comandos de Estilo de Texto são:

COMANDO	SINTAXE	FUNÇÃO
Negrito	<b>&lt;b&gt; texto &lt;/b&gt;</b>	Aplica negrito.
Itálico	<i>&lt;i&gt; texto &lt;/i&gt;</i>	Aplica itálico.
Sublinhado	<u>&lt;u&gt; texto &lt;/u&gt;</u>	Aplica sublinhado.
Big	<b>&lt;big&gt; texto &lt;/big&gt;</b>	Aumenta o tamanho da fonte e aplica negrito.
Tachado	<b>&lt;s&gt; texto &lt;/s&gt;</b>	Aplica tachado.
Small	<b>&lt;small&gt; texto &lt;/small&gt;</b>	Reduz e altera a fonte.
Sobrescrito	<b>&lt;sup&gt; texto &lt;/sup&gt;</b>	Aplica sobrescrito.
Subscrito	<b>&lt;sub&gt; texto &lt;/sub&gt;</b>	Aplica subscrito.
Pisca-pisca	<b>&lt;blink&gt; texto &lt;/blink&gt;</b>	Faz o texto piscar. Alguns navegadores não aceitam este comando.

Comando **<marquee width=... scroll=... direction=... scrollamount=...> ... </marquee>** - Faz o texto rolar horizontalmente pela tela.

O atributo WIDTH define a largura de rolagem do texto.

O atributo SCROLL define o tempo de rolagem.

O atributo DIRECTION define a direção de rolagem. Pode ser left ou right.

O atributo SCROLLAMOUNT define a velocidade de rolagem do texto.

Exemplo:

```
<marquee width="50%" scroll="infinite" direction="left"
scrollamount="10"> texto </marquee>
```

#### EXERCÍCIO

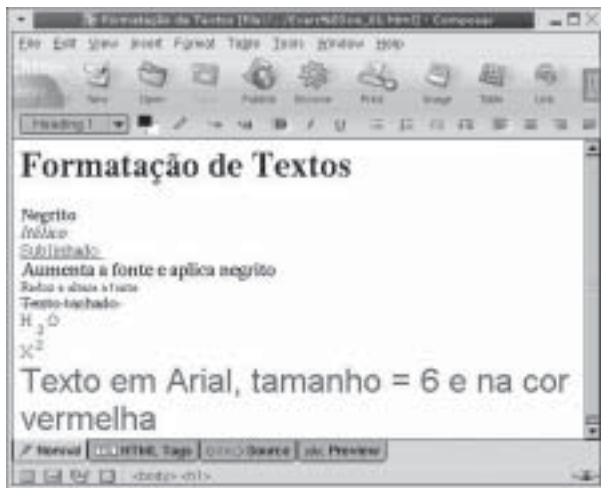
Abrir um arquivo novo no Composer e digitar os comandos abaixo.

```
<html>
<head>
<title> Formatação de Textos </title>
</head>
<body>
<h1> Formatação de Textos </h1>
<b> Negrito </b><br>
<i> Itálico </i><br>
<u> Sublinhado </u><br>
<big> Aumenta a fonte e aplica negrito </big><br>
<small> Reduz e altera a fonte </small><br>
<s> Texto tachado </s> <br>
H <sub> 2 </sub> O <br>
X<sup> 2 </sup> <br>
```

```
<font face="Arial" size="6" color="red"> Texto em Arial,  
tamanho = 6 e  
na cor vermelha </font>  
</body>  
</html>
```

Observação: O comando `<br>` força uma quebra de linha, sem deixar uma linha em branco.

Salvar no formato texto, com o nome `exercicio02.htm` e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Formatação de Parágrafos

Comando `<P ALIGN=...> ... </P>` - Delimita um parágrafo. É possível omitir o elemento de fim `</p>` sem danos ao resultado final. O efeito visual é uma linha em branco antes do início do próximo parágrafo. O atributo opcional **ALIGN** configura o alinhamento do parágrafo e pode conter os valores `left` (esquerda), `center` (centro) ou `right` (direita).



Comando **<CENTER> ... </CENTER>** - Centraliza texto e parágrafos.

Comando **<BLOCKQUOTE> ... </BLOCKQUOTE>** - Este comando destaca um bloco de texto citado de outra fonte através de uma endentação à direita. É utilizado na criação de citações longas que não devem ser aninhadas em parágrafos.

## Linhas Horizontais

As **Linhas Horizontais** são utilizadas para dar destaque a títulos ou para gerar a sensação de quebra entre um e outro item de informação. Trata-se de uma marca solitária, que pode ser utilizada com alguns atributos opcionais **<HR SIZE=... WIDTH=... ALIGN=...>** ou sem nenhum atributo **<HR>**.

O atributo **SIZE** configura a espessura da linha.

O atributo **WIDTH** configura a largura da linha, e pode ser especificado em pixels ou percentual.

O atributo **ALIGN** configura o alinhamento da linha. Pode ser: left, center e right.

## Hyperlinks

Os Hyperlinks são construções HTML que permitem o acesso a recursos na Internet com um simples clique. É possível programar um hyperlink para levar um usuário para o topo da página atual ou para um servidor em outro país.

Os hyperlinks podem ter a forma de hipertexto ou hipermissão. Na maioria dos navegadores, o hipertexto é exibido como um texto azul e sublinhado. A categoria hipermissão inclui hiperimagens. Estas são simples imagens gráficas incorporadas, tais como fotografias ou ícones, que aceitam vínculos.

Os destinos de um hyperlink são:

- a) Um indicador (um local na mesma página);
- b) Uma página interna;
- c) Um e-mail;
- d) Um site externo;

#### DEFINIR UM LINK DENTRO DE UMA PÁGINA

Um link dentro de uma página é um hyperlink que aponta para um outro local dentro da mesma página.

Um link dentro de uma página tem duas partes: o link onde o usuário deve clicar e o ponto na página para o qual o cursor deve saltar. A marca Âncora <A> permite links para outras páginas ou para outro local na mesma página.

No caso deste tipo de vínculo, é preciso definir uma âncora com o nome do local de destino. O formato é o seguinte: <A **NAME="#topo"**>, onde **topo** é o nome atribuído ao local de destino. O nome do local de destino deve iniciar com uma letra e não pode começar com número e nem com caractere especial.

Exemplo:

```
<a name="topo"> ... </a>
```

...

```
<a href="#topo"> Voltar para o Início </a>
```

Onde **#topo** é o indicador do local de destino. O símbolo **cerquilha (#)** instrui ao navegador a posição do indicador nomeado. **Voltar para o Início** é a frase que os navegadores exibem como hyperlink.

## DEFINIR UM LINK PARA UMA PÁGINA INTERNA

Um Web site consiste de pelo menos algumas páginas separadas. Para tornar estas páginas acessíveis e convenientes para os visitantes, é preciso definir os links entre as páginas. Costuma-se colocar estes links no início ou no final de cada página.

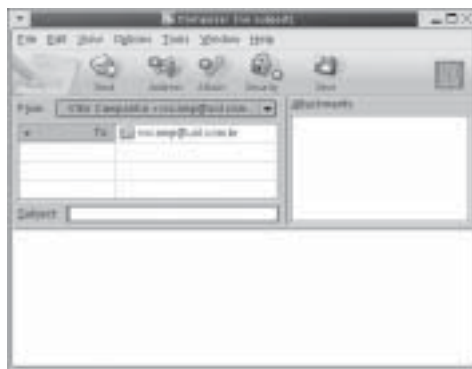
Exemplo:

```
<a href="página2.html"> Link para a Página 2 </a>
```

Neste exemplo é criado um link para a página de nome página2.html.

## DEFINIR UM LINK "MAILTO"

Um link Mailto permite que os visitantes enviem mensagens para o endereço de e-mail definido, simplesmente clicando no hyperlink. Quando o usuário clica no link mailto, o aplicativo de correio eletrônico padrão abre uma janela de mensagem endereçada para o mailto de destino. Veja a figura abaixo.



A seção de rodapé é um bom local para se colocar um link mailto.

Observação: Para este link funcionar corretamente, é necessário que os programas de correio estejam configurados.

Exemplo:

```
<a href="mailto:aluno1@telecentros.sp.gov.br"> Telecentros </a>
```

DEFINIR UM LINK PARA OUTRO SITE

Os hyperlinks externos fazem vínculos com outros sites na Internet. São eles que tornam a Internet possível. Deve-se incluir uma URL completa quando você definir um hyperlink externo.

Exemplo:

```
<a href="http://www.telecentros.sp.gov.br"> Telecentros </a>
```

DEFINIR UM LINK EM UMA IMAGEM

Um link em uma imagem é também chamado de vínculo de hipermídia, que é um gráfico que funciona com um hyperlink. Um uso para vínculos em imagens é criar botões e adicioná-los à barra de navegação.

Exemplo:

```
<a href="pagina2.html">  </a>
```

## Imagens

Além de texto, é possível colocar imagens em páginas Web. Os formatos gráficos padrões da Internet são: **JPEG** e **GIF**. Cada um deles tem vantagens e desvantagens. A escolha entre qual deles você usará vai depender da imagem desejada.

### a) GIF - Graphics Interchange Format

O formato GIF usa uma técnica de compressão no mesmo estilo de programas como o Winzip. Os arquivos GIF podem ter um máximo de 256 cores, e para a grande maioria dos usuários é mais do que suficiente, já que a grande maioria das pessoas tem suas placas de vídeo configuradas com este número de cores.

## b) JPEG - Joint Photographic Experts Group

O formato JPEG foi criado para que imagens "true-color" (com 16 milhões de cores) pudessem ser armazenadas em arquivos pequenos. O formato JPEG consegue isso usando um sistema onde a qualidade da imagem é reduzida de forma que o arquivo fique menor. Só que, na maioria das vezes, a perda de qualidade, não é notada na tela de um monitor, daí o seu grande sucesso.

Ao contrário do GIF, que tem uma compressão padrão (que não pode ser alterada), o JPEG tem uma compressão variável. Porém, quanto mais você comprime, maior será a perda de qualidade.

### Principais Diferenças entre as Imagens GIF e JPEG

CARACTERÍSTICA	GIF	JPEG
Animação	Sim	Não
Tamanho do arquivo	Maior	Menor
Resolução	Menos cores (256)	Mais cores (16 milhões)
Transparência	Sim	Não
Velocidade de download	Mais rápido	Mais lento
Exibição	Monta distorcida	Monta por faixas

### Qual Formato Escolher?

O que vai determinar o formato a ser escolhido é o número de cores de sua imagem. Normalmente, você deve usar o GIF para desenhos ou logotipos, e usar JPEG para fotos ou imagens com muita variação de cores.

Para que uma imagem seja apresentada em um documento HTML, é necessário fazer uma referência ao nome do arquivo da imagem, através da marca

**<IMG SRC="figura.gif">.**

Caso a imagem esteja em um diretório diferente do documento HTML, o nome do diretório deve ser incluído na referência.

Lembre-se que você pode também usar uma imagem como link. Neste caso, combine os elementos de link e imagem.

Exemplo:

```
<a href="http://www.telecentros.sp.gov.br">  
 </a>.
```

O atributo **BORDER** serve para definir uma borda para a imagem.

O atributo **ALT** server para definir uma legenda para a imagem com no máximo 255 caracteres.

## **GIFs Animados**

Um recurso muito bom e bastante utilizado para criar animações na sua página é utilizar as Gifs animadas. **GIF** é o formato de imagem mais utilizado na Internet. Esse padrão tem duas grandes vantagens: utilizar fundos transparentes e criar animações.

## **Definir uma Imagem no Segundo Plano**

Se você quiser ir além de um segundo plano apenas colorido, o HTML lhe permite atribuir uma imagem, para ser exibida no segundo plano de sua página Web.

Exemplo:

```
<body background="figura.gif">
```

O comando `<! ... —>` - Permite inserir comentários no seu programa.

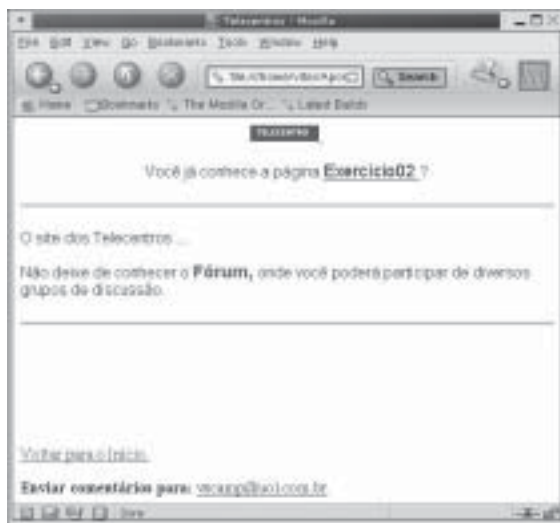
## EXERCÍCIO

Abrir um arquivo novo e digitar os comandos abaixo.

```
<html>
<head>
  <title>Telecentros</title>
</head>
<body>
  <a name="topo"> </a>
  <p align="center">
    <!--Link para outro site, usando uma figura -->
    <a href="http://www.telecentros.sp.gov.br">
       </a>
    <br> <br>
    <font face="arial" size="4">
      Você já conhece a página
      <!--Link para uma página -->
      <a href="exercicio02.html"> Exercício02 </a> </p>
      <hr size="2">
      <p> O site dos Telecentros ... </font> </p>
      <p> Não deixe de conhecer o <b>
        <font size="4" color="red"> Fórum, </font> </b> onde você
        poderá participar de diversos grupos de discussão.
      </font> </p>
      <hr size="2">
      <br> <br> <br> <br> <br> <br> <br>
      <!--Link para dentro da mesma página -->
      <a href="#topo"> Voltar para o Início </a>
      <br> <br>
      <b> Enviar comentários para:</b>
      <!--Link para correio eletrônico -->
```

```
<a href="mailto:fulano@beltrano.com.br">fulano@beltrano.com.br</a>
</body>
</html>
```

Salvar no formato texto, com o nome exercicio03.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Listas

Na linguagem HTML existem elementos específicos para a criação de listas, que podem ser listas ordenadas (**OL**), listas sem ordenação (**UL**), ou listas de definições (**DL**).

Um detalhe interessante é que podem ser criadas listas aninhadas, ou seja, listas dentro de listas. A seguir, veremos as características de cada uma delas.



## LISTAS ORDENADAS

**<OL TYPE=... START=... > </OL>** - OL significa **Ordered Lists**.

A estrutura das listas ordenadas é bastante simples. Entre os elementos de início e fim, os itens da lista são definidos pelos elementos **<LI> </LI>**.

O atributo opcional **TYPE** define como será o tipo de numeração de cada linha. Os tipos disponíveis são: "A" (A, B, ..., Z), "a" (a, b, ..., z), "I" (i, ii, ..., v), e "1" (1, 2, ..., 5). Se omitido, é utilizado o tipo padrão (1, 2, 3, 4).

O atributo opcional **START** define a partir de que elemento a numeração deve se iniciar.

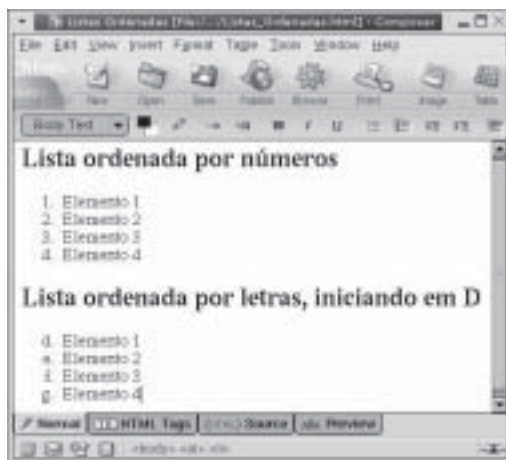
## EXERCÍCIO

Abrir um arquivo novo e digitar os comandos abaixo.

```
<html>
<head>
<title> Listas Ordenadas </title>
</head>
<body>
<h2> Lista ordenada por números </h2>
<ol>
<li> Elemento 1 </li>
<li> Elemento 2 </li>
<li> Elemento 3 </li>
<li> Elemento 4 </li>
</ol>
<h2> Lista ordenada por letras, iniciando em D </h2>
<ol type="a" start=4>
```

```
</li> Elemento 1 </li>
</li> Elemento 2 </li>
</li> Elemento 3 </li>
</li> Elemento 4 </li>
</ol>
</body>
</html>
```

Salvar no formato texto, com o nome exercicio04.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## LISTAS NÃO ORDENADAS

**<UL TYPE=...> ... </UL>** - **UL** significa **U**nordered **L**ists.

A estrutura das listas sem ordenação é a mesma das listas ordenadas, sendo que, na apresentação, os itens serão precedidos por um marcador (bullet). No caso de listas sem ordenação aninhadas (listas dentro de listas), o paginador utilizará uma marcador (bullet) diferente para os itens de cada lista.

O atributo **TYPE** pode ser:

**square:** É um quadrado preenchido.

**circle:** É um círculo vazado.

**disk:** É um círculo preenchido.

## EXERCÍCIO

Abrir um arquivo novo e digitar os comandos abaixo.

```
<html>
<head>
<title> Listas Não-Ordenadas </title>
</head>
<body>
<h2> Lista Não-Ordenada </h2>
<ul>
<li> Elemento 1 </li>
<li> Elemento 2 </li>
<li> Elemento 3 </li>
<li> Elemento 4 </li>
</ul>
<h2> Duas listas Não-Ordenadas aninhadas </h2>
<ul type=square>
<li> Elemento 1 </li>
<li> Elemento 2
<ul>
<li> Elemento 2.1 </li>
<li> Elemento 2.2 </li>
<li> Elemento 2.3 </li>
</ul>
</li>
<li> Elemento 3 </li>
<li> Elemento 4 </li>
</ul>
</body>
</html>
```

Salvar no formato texto, com o nome `exercicio05.htm` e executar o programa.

Como já foi visto acima, `<LI> ... </LI>` é o elemento que define um item de uma lista ordenada ou sem ordenação. O seu conteúdo pode ser texto, outras listas, imagens, links, etc.

#### LISTA DE DEFINIÇÃO

`<DL> ... </DL>` - São marcas que englobam uma lista de definições, ideais para a criação de glossários e coisas do gênero. A estrutura desta lista é diferente das outras, pois existem dois elementos - o termo a ser definido (**DT**), e a definição propriamente dita (**DD**). Na exibição cada termo aparece em uma linha, e a respectiva definição na linha seguinte, deslocada para a direita.

#### EXERCÍCIO

Abrir um arquivo novo e digitar os comandos abaixo.

```
<html>
<head>
<title> Listas de Definição </title>
</head>
<body>
<h2> Listas de Definição </h2>
<dl>
<dt> HTML </dt>
<dd> HyperText Markup Language </dd>
<dt> OL </dt>
<dd> Listas Ordenadas </dd>
<dt> UL </dt>
<dd> Listas Sem Ordenação </dd>
<dt> LI </dt>
<dd> Elemento da Lista </dd>
```

```
</dl>  
</body>  
</html>
```

Salvar no formato texto, com o nome exercicio06.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



Ao fazer suas listas, lembre-se sempre que você pode combinar tipos diferentes para chegar ao resultado esperado.

## Tabelas

Assim como as listas, no HTML existem elementos específicos para a criação e formatação de tabelas. O recurso de tabelas é muito interessante e muito usado nas páginas Web. As tabelas são formadas por **linhas** e **colunas**, e na interseção delas estão as **células**.

Na linguagem HTML, você pode inserir nas células tudo que normalmente faz parte do corpo de um documento, como textos, links, imagens, listas e até outras tabelas.

## ELEMENTOS DE UMA TABELA

As marcas que englobam a definição de uma tabela são: **<TABLE> ... </TABLE>**. Todas as outras marcas referentes à tabelas - linhas e células - somente serão consideradas se incluídas entre estas marcas.

Comando **<CAPTION> ... </CAPTION>** - Trata-se de um elemento opcional que define o título da tabela, e deve constar entre as marcas que definem a tabela, mas separado das marcas que definem linhas e colunas. Sem parâmetros, o título é apresentado acima da tabela e centralizado.

Comando **<TR> ... </TR>** - É a abreviação das palavras Table Row ou Linha de Tabela. Este é o elemento utilizado na definição de linhas das tabelas. As tabelas são definidas em linhas, sendo as linhas compostas de células. O número de linhas de uma tabela corresponde ao número de **<TR> </TR>** existentes.

Comando **<TD> ... </TD>** - É a abreviação das palavras Table Data ou Dado de Tabela. Estas marcas delimitam as células que compõem as linhas, e assim sendo devem estar inseridas entre marcas de linhas.

O alinhamento padrão de uma célula é à esquerda horizontalmente e centralizado verticalmente, e caso o número de células varie de uma linha para outra, as linhas com menos células são completadas à direita com células em branco.

Comando **<TH> ... </TH>** - É a abreviação das palavras Table Header ou Cabeçalho de Tabela. É o elemento que define as células de cabeçalho. As células de cabeçalho têm as características idênticas às células de dados definidas por **<TD>**, a não ser pelo alinhamento padrão horizontal, que é centralizado, e pela utilização de fonte em negrito.

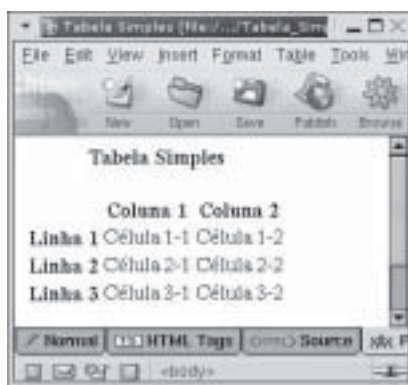
## EXERCÍCIO

Abrir um arquivo novo e criar a tabela da figura abaixo.

```
<html>
<head>
<title> Tabela Simples </title>
</head>
<body>
<br>
<table>
  <caption> <b> Tabela Simples </b> </caption> <br>
  <tbody>
    <tr>
      <td> <br>
    </td>
    <th> Coluna 1 </th>
    <th> Coluna 2 </th>
  </tr>
  <tr>
    <th> Linha 1 </th>
    <td> Célula 1-1 </td>
    <td> Célula 1-2 </td>
  </tr>
  <tr>
    <th> Linha 2 </th>
    <td> Célula 2-1 </td>
    <td> Célula 2-2 </td>
  </tr>
  <tr>
    <th> Linha 3 </th>
    <td> Célula 3-1 </td>
    <td> Célula 3-2 </td>
  </tr>
</tbody>
```

```
</table>  
<br>  
<br>  
<br>  
</body>  
</html>
```

Salvar no formato texto, com o nome exercicio07.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Parâmetros

Estes elementos básicos possuem alguns parâmetros que permitem um maior controle sobre alguns detalhes da apresentação da tabela.

### Parâmetros do Elemento <TABLE>

#### BORDER

A apresentação padrão de uma tabela é sem bordas. A presença do parâmetro **BORDER** indica justamente que deve ser desenhada uma borda em torno de cada célula da tabela. Pode ser usado para indicar a espessura da borda em pixels.



Exemplo: `<table border=2>`

#### WIDTH

Especifica a largura da tabela, que pode ser definida em pixels ou em percentual referente à largura da janela. Caso não seja especificado, o próprio browser se encarrega de determinar a largura mais adequada, baseado nos textos inseridos nas células.

Exemplo: `<table width=75%>`

#### CELLSPACING

Define o espaço entre as células, ou seja, a largura das linhas de grade (as bordas que envolvem as células). É especificado em pixels.

Exemplo: `<table cellpadding=3>`

#### CELLPADDING

Determina em pixels, o espaço entre o conteúdo e as bordas da célula.

Exemplo: `<table cellpadding=6>`

#### ALIGN

Configura o alinhamento horizontal da tabela em relação aos outros elementos da página. Pode conter valores LEFT (esquerda), CENTER (centro) ou RIGHT (direita). Este parâmetro não funciona em alguns browsers.

## BGCOLOR

Define a cor de fundo da tabela.

## EXERCÍCIO

Acrescentar no Exercício 07 a linha abaixo:

```
<body>  
...  
<table border="1" width="50%" cellpadding="4"  
bgcolor="yellow">
```

Salvar no formato texto, com o nome exercicio04.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Parâmetros do Elemento <TR>

### ALIGN

Configura o **alinhamento horizontal** dos elementos contidos nas células de uma linha. Pode conter os valores LEFT (esquerda), CENTER (centro) ou RIGHT (direita). Se omitido, o alinhamento é à esquerda para as células de dados (<TD>), e centralizado para células de cabeçalho (<TH>).

## VALIGN

Define o **alinhamento vertical** dos elementos contidos nas células de uma linha. Pode conter os valores TOP (topo), MIDDLE (meio) ou BOTTOM (fundo). Se omitido, o alinhamento é ao meio.

## BGCOLOR

Define a cor da linha.

## Parâmetros dos Elementos <TD> e <TH>

### ALIGN

Configura o alinhamento horizontal dos elementos contidos na célula. Pode conter os valores LEFT (esquerda), CENTER (centro) ou RIGHT (direita). Se omitido, o alinhamento é à esquerda para as células de dados (<TD>), e centralizado para células de cabeçalho (<TH>).

### VALIGN

Define o alinhamento vertical dos elementos contidos na célula. Pode conter os valores TOP (topo), MIDDLE (meio) ou BOTTOM (fundo). Se omitido, o alinhamento é ao meio.

### BGCOLOR

Define a cor da linha.

### NOWRAP

Quando este parâmetro encontra-se associado a uma célula, o browser entende que o texto dentro daquela célula não pode ser dividido em mais de uma linha. Este parâmetro deve ser utilizado com cuidado, para evitar colunas demasiadamente largas.

## COLSPAN

Especifica o número de colunas de uma tabela a ser ocupado por uma célula. Deve ser utilizado para expandir uma célula **horizontalmente**. Se atribuirmos COLSPAN=2 a uma célula, ela ocupará o seu espaço e o espaço de mais de uma célula para a direita.

## ROWSPAN

Define o número de linhas a ser ocupado por uma célula. Deve ser utilizado para expandir uma célula **verticalmente** (para baixo). Ao atribuir rowspan=2 para uma célula, diminua em 1 o número de células da linha de baixo.

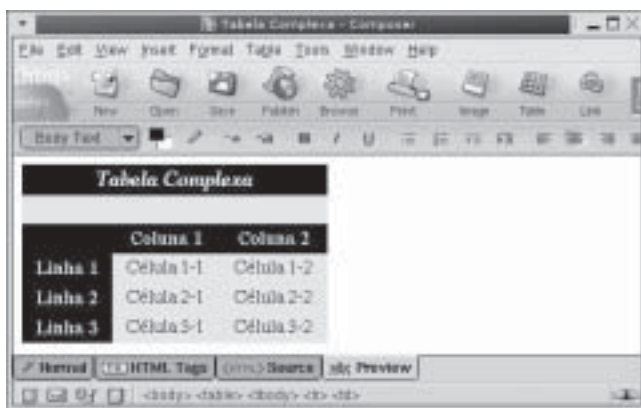
## EXERCÍCIO

Abrir um arquivo novo e digitar os comandos abaixo, para criar uma tabela complexa.

```
<html>
<head> <title> Tabela Complexa </title> </head>
<body>
<table width=50%    cellpadding=4
bgcolor=yellow>
  <tr bgcolor=black align=center>
    <td colspan=3> <font size=3 color=yellow> <i> <b> Tabela
Complexa </b>
    </i> </font> </td>
  </tr>
  <tr> <td colspan=3> </td> </tr>
  <tr bgcolor=black>
    <td> </td>
    <th> <font color=yellow> Coluna 1<font face=yellow> </th>
    <th> <font color=yellow> Coluna 2 <font face=yellow></th>
  </tr>
```

```
<tr align=center>
<th bgcolor=black> <font color=yellow> Linha 1 </font> </th>
<td> Célula 1-1 </td>
<td> Célula 1-2 </td>
</tr>
<tr align=center>
<th bgcolor=black> <font color=yellow> Linha 2 </font> </th>
<td> Célula 2-1 </td>
<td> Célula 2-2 </td>
</tr>
<tr align=center>
<th bgcolor=black> <font color=yellow> Linha 3 </font> </th>
<td> Célula 3-1 </td>
<td> Célula 3-2 </td>
</tr>
</table>
</body>
</html>
```

Salvar no formato texto, com o nome exercicio09.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Formulários

Com certeza você já deve ter preenchido algum formulário em suas viagens pela Internet. Seja para cadastros, pesquisas ou mesmo envio de comentários, os formulários aumentam o poder de interação da Web e são uma forma diferenciada de receber dados dos visitantes de sua página.

Atualização ou consultas a base de dados, envio de dados por e-mail, ou simplesmente a construção de uma nova página (gerada a partir dos novos dados) são algumas das aplicações mais comuns.

### ELEMENTOS HTML PARA FORMULÁRIOS

1. **<form** action=... method=... target=... > ... **</form>**

Estes são os elementos que delimitam um formulário numa página.

O parâmetro ACTION (ação) deve conter o endereço do programa que vai receber os dados no formulário (escrito em PHP por exemplo).

O parâmetro METHOD (método) define como as informações coletadas no formulário serão enviadas para a URL indicada no atributo ACTION. Ele deve ter os valores GET ou POST. O valor mais comum é POST, que envia todas as informações separadamente. O GET envia todos os dados do formulário em uma sequência de caracteres concatenados.

O parâmetro TARGET (alvo) é opcional e só é necessário quando você está usando frames, e deseja que a resposta ao formulário seja exibida em um frame diferente do que está o formulário.

2. **<input type=... name=... value=... size=... maxlenght=... checked>**

INPUT significa entrada de dados, logo este é um dos elementos que determina como será a entrada de dados nos campos de um formulário.

O INPUT é o elemento que define o formato da entrada dos dados no formulário. Use-o para montar caixas de texto, botões e até caixas de verificação de senhas.

O parâmetro TYPE (tipo) é muito importante pois define o tipo do elemento: caixa de texto, botão de escolha, ou botão simples. Para ficar mais claro, mostraremos um de cada vez, explicando quais parâmetros devem ser usados.

O parâmetro NAME deve conter o nome, ou identificador, do campo e o parâmetro VALUE pode conter um valor predefinido para o campo.

#### TIPOS DO ELEMENTO INPUT

##### 2.1. **<input type=**text **name=... value=... size=... maxlenght=...>**

O valor TEXT (texto) no parâmetro TYPE indica que o campo será de texto, ou seja, uma campo onde você digita os dados.

O parâmetro VALUE (valor), neste caso, pode ser usado se você quiser definir uma valor prévio para o campo, de tal forma que, quando a página for carregada, este valor já venha preenchido.

O parâmetro SIZE (tamanho) configura o tamanho do campo e é baseado no número de caracteres. Se você quiser um campo com tamanho de 40 caracteres, deverá digitar SIZE=40. Note que este valor não limita o campo em 40 caracteres, ele somente define o tamanho com que será mostrado na página.

O que define o número máximo de caracteres que podem ser digitados é o parâmetro **MAXLENGTH** (comprimento máximo), que é opcional.

2.2. `<input type=password name=... value=... size=... maxlength=...>`

Com o valor **PASSWORD** (senha) no parâmetro **TYPE**, tudo funciona da mesma forma que o valor **TEXT**, exceto que todas as letras digitadas aparecem como um asterisco (tal como quando você digita sua senha para se conectar à Internet).

2.3. `<input type=radio name=... value=... checked>`

O valor **RADIO** no parâmetro **TYPE** define botões de escolha. Eles são usados para questões onde somente uma opção pode ser selecionada. Em nosso exemplo, este tipo de elemento serve para construir as opções para a pergunta "Você gostou da minha página?".

O parâmetro **NAME**, neste caso, deve ser igual para todos os campos deste tipo, pois estará identificando a questão formulada.

O parâmetro **VALUE** deve conter o valor do campo, que será passado ao programa interpretador do formulário.

O parâmetro **CHECKED** deve ser usado quando você deseja que uma das opções esteja selecionada como padrão.

2.4. `<input type=checkbox name=... value=... checked>`

O valor **CHECKBOX** no parâmetro **TYPE** define botões de verificação. A tradução do termo **checkbox** é meio difícil, mas, ao contrário dos botões de escolha (do tipo **RADIO**), ele deve ser usado quando uma ou mais opções são válidas. No nosso exemplo,



usamos este tipo na pergunta “Você gostaria que eu respondesse a este seu comentário?”.

O parâmetro NAME, neste caso, deve ser diferente para cada campo.

O parâmetro VALUE deve conter o valor do campo, que será passado ao programa interpretador do formulário.

O parâmetro CHECKED pode ser usado se você desejar que a opção esteja selecionada como padrão.

#### 2.5. <input type=**reset** value=...>

O valor RESET no parâmetro TYPE define um botão que limpa todos os campos, colocando os mesmos valores de quando a página foi carregada.

No parâmetro VALUE, pode-se definir o que estará escrito no botão. Caso nenhum valor seja definido, aparecerá escrito somente “Reset”. Em nosso exemplo usamos este elemento no botão “Limpar todos os campos”.

#### 2.6. <input type=**submit** name=... value=...>

O valor SUBMIT no parâmetro TYPE define um botão que aciona o envio das informações preenchidas no formulário ao programa interpretador (aquele definido no parâmetro ACTION do elemento FORM).

O parâmetro VALUE define o que estará escrito no botão. Caso nenhum valor seja definido, aparecerá escrito “Submit”. Para definir o botão “Enviar os comentários” de nosso exemplo, usamos este tipo de elemento.

### 2.7. <input type=**image** name=... src=... alt=...>

Alternativamente ao botão tipo SUBMIT, pode-se usar uma imagem. Para isso, existe o tipo IMAGE. Neste caso, passam a existir os parâmetros SRC e ALT normalmente usados no elemento que define uma imagem.

O parâmetro SRC define o endereço ou nome do arquivo da imagem.

O parâmetro ALT, opcional, define o texto que será mostrado, caso a imagem não seja carregada.

### 2.8. <input type=**hidden** name=... value=...>

O valor HIDDEN (oculto) no parâmetro TYPE define dados que devem ser passados ao programa interpretador, mas que não devem aparecer para quem está vendo a sua página

Neste caso, o parâmetro NAME identifica o dado e o parâmetro VALUE define o valor que deve ser passado.

### 3. <**select** name=... size=... > <option value=...> <option value=...> ... </select>

O elemento SELECT possibilita definir uma lista de opções para o visitante escolher.

O parâmetro NAME define o nome desta lista, e SIZE define quantos elementos serão exibidos na tela simultaneamente. Se for omitido, somente uma opção aparece por vez.

Cada opção da lista é definida através de um elemento `OPTION` e o parâmetro `VALUE` deste elemento é que determinará o valor de cada opção.

Em nosso exemplo, usamos este elemento para criar a lista de opções para a resposta da pergunta "Qual é a página de que você mais gostou?".

#### 4. `<textarea name=... rows=... cols=...> ... </textarea>`

O elemento `TEXTAREA` (área de texto) permite definir um campo de texto com várias linhas. Ele é útil para colher comentários e depoimentos dos visitantes.

O parâmetro `ROWS` define o número de linhas da caixa de texto, e o parâmetro `COLS` define quantos caracteres (colunas) cada linha possui.

O parâmetro `NAME` define o nome da caixa de texto.

#### EXERCÍCIO

Abrir um arquivo novo e digitar os comandos abaixo, para criar um formulário.

```
<html>
<head>
  <title>Formulário 01</title>
</head>
<body>
  <h4> Cadastro de Clientes </h4>
  <hr>
  <form          method="post"          action="mailto:
meu_nome@provedor.com.br">
    <input type="hidden" name="AnyFormMode" value="mail">
    <input      type="hidden"      name="AnyFormDisplay"
```

```
value="standard">
    <input      type="hidden"      name="AnyFormTo"
value="meu_nome@provedor.com.br">
    <input type="hidden" name="AnyFormSubject" value="Dados
dos Clientes">
    <b> Nome: </b> <input type="text" name="nome" size="30"
maxlength="5">
    <br> <br>
    <b> Digite o seu E-Mail: </b> <input type="text"
name="anyformfrom" size="30">
    <b> Senha do E-Mail: </b> <input type="password"
name="senha" size="5" maxlength="5">
    <br> <br>
    <b>Sexo:</b>
    <br>
    <input type="radio" name="sexo"> Masculino <br>
    <input type="radio" name="sexo"> Feminino <b>
    <br> <br>
    Produtos Comprados:</b>
    <br>
    <input type="checkbox" name="computador"> Computador
<br>
    <input type="checkbox" name="impressora"> Impressora <br>
    <input type="checkbox" name="monitor"> Monitor <br>
    <input type="checkbox" name="teclado"> Teclado <br>
    <br>
    <b> Estado: </b>
    <select name="listagem" selected="">
    <option size="3"> AM </option>
    <option> MG </option>
    <option> PE </option>
    <option> RJ </option>
    <option> SP </option>
    </select>
    <br> <br>
    <b> Histórico do Cliente: </b><br>
```

```
<textarea name="observacao" cols="30" rows="3"> </textarea>
<br> <br>
<input type="file" name="Procurar" value="Procurar">
<input type="submit" value="Enviar Informações">
<input type="reset" value="Limpar"> <br>
</form>
</body>
</html>
```

Salvar no formato texto, com o nome exercicio10.htm e executar o programa. Veja na figura abaixo como deverá ficar a sua tela.



## Hospedando seu Site

Depois de passar por todas as etapas apresentadas neste manual, nada melhor do que conhecer como hospedar sua home page na Internet.

Você tem duas opções. Ou escolhe um serviço de hospedagem **gratuito** ou um serviço de hospedagem **pago**. Os dois têm vantagens, desvantagens e limitações.

## **Sites de Hospedagem Gratuita**

Se o seu site é pessoal, então a escolha da hospedagem gratuita pode ser uma boa opção. Para isso, você deve escolher um serviço de hospedagem gratuita como o Geocities, ou outro qualquer de sua preferência.

A desvantagem do serviço grátis é que o seu site pode acabar tendo um endereço gigantesco e difícil de decorar, e terá janelas abrindo exibindo anúncios inseridos pelo serviço de hospedagem.

## **Sites de Hospedagem Paga**

Os serviços de hospedagem paga oferecem muito mais recursos do que qualquer empresa de hospedagem grátis pode oferecer. A principal vantagem é o registro de domínio, isto é, você pode ter um site com o nome do tipo `www.seunome.com.br`, coisa que nenhum serviço de hospedagem grátis oferece.

## **Hospedando seu Site no Geocities**

Para hospedar seu site no Geocities basta acessar o endereço `http://br.geocities.yahoo.com/`, e seguir as instruções em português. Os passos são simples, e basta segui-los à risca para ter seu site publicado.



PARTE II  
**Java Script**



## Java Script

JavaScript é uma linguagem que permite injetar lógica em páginas escritas em HTML (HiperText Mark-up Language). Você terá aqui uma breve visão de como funciona o JavaScript, além de sua utilização prática.

Os parágrafos de lógica do JavaScript podem estar “soltos” ou atrelados á ocorrência de eventos. Os parágrafos soltos são executados na sequência em que aparecem na página (documento) e os atrelados a eventos são executados apenas quando o evento ocorre.

Para inserir parágrafos de programação dentro do HTML é necessário identificar o início e o fim dos comandos JavaScript, da seguinte forma:

```
<SCRIPT>  
Set de instruções  
</SCRIPT>
```

Este procedimento pode ser adotado em qualquer local da página. Entretanto, para melhor visualização e facilidade de manutenção, recomenda-se que toda a lógica seja escrita no início do documento, através da criação de funções a serem invocadas quando se fizer necessário (normalmente atreladas a eventos).

Se a lógica é escrita a partir de um determinado evento, não é necessário o uso dos comandos `<SCRIPT>` e `</SCRIPT>`.

Os comandos JavaScript são sensíveis ao tipo de letra (maiúsculas e minúsculas) em sua sintaxe.

## Operadores Lógicos

São operadores a serem utilizados em comandos condicionais, tais como: IF , FOR e WHILE.

Os comandos condicionais serão vistos mais a frente.

==	Igual
!=	Diferente
>	Maior
>=	Maior ou Igual
<	Menor
<=	Menor ou Igual
&&	E
	Ou

## Operadores Matemáticos

São operadores a serem utilizados em cálculos, referências de indexadores e manuseio de strings.

Estes operadores são largamente utilizados.

+	adição de valor e concatenação de strings
-	subtração de valores
*	multiplicação de valores
/	divisão de valores
%	obtem o resto de uma divisão:

Ex: 150 % 13 retornará 7

7 % 3 retornará 1

+=	concatena /adiciona ao string/valor já existente.
----	---

Ou seja:

$x += y$  é o mesmo que  $x = x + y$

Da mesma forma podem ser utilizados:  $-=$  ,  $*=$  ,  $/=$  ou  $\%=$

Um contador pode ser simplificado utilizando-se :  $X++$  ou  $X--$  o que equivale às expressões:

$X = X + 1$  ou  $X = X - 1$  respectivamente.

Para inverter sinal:  $X = -X$  negativo para positivo ou positivo para negativo.

## Controles Especiais

`\b` - backspace

`\f` - form feed

`\n` - new line characters

`\r` - carriage return

`\t` - tab characters

`//` - Linha de comentário

`/*...*/` - Delimitadores para inserir um texto com mais de uma linha como comentário.

Os delimitadores naturais para uma string são `"` ou `'` . Caso seja necessário a utilização destes caracteres como parte da string, utilize `\` precedendo `"` ou `'` .

Ex. `alert ("Cuidado com o uso de \" ou \' em uma string")`

## Comandos Condicionais

São comandos que condicionam a execução de uma certa tarefa à veracidade ou não de uma determinada condição, ou enquanto determinada condição for verdadeira.

São eles:

COMANDO IF:

```
if (condição)
{ ação para condição satisfeita }
[ else
{ ação para condição não satisfeita } ]
```

Ex:

```
if (Idade < 18)
{Categoria = "Menor" }
else
{Categoria = "Maior"}
```

COMANDO FOR:

```
for ( [inicialização/criação de variável de controle ;]
[condição ;]
[incremento da variável de controle] )
{ ação }
```

Ex:

```
for (x = 0 ; x <= 10 ; x++)
{alert ("X igual a " + x) }
```

COMANDO WHILE:

Executa uma ação enquanto determinada condição for verdadeira.

```
while (condição)
{ ação }
```

Ex:

```
var contador = 10  
while (contador > 1)  
{ contador— }
```

OBS: Nos comandos FOR e WHILE a diretiva “break” pode ser utilizada para interromper a condição principal e sair do loop. Da mesma forma, a diretiva “continue” interrompe uma ação (se determinada condição ocorrer) mas volta para o loop.

Diretivas/condições entre [ ] significam que são opcionais.

## Eventos

São fatos que ocorrem durante a execução do sistema, a partir dos quais o programador pode definir ações a serem realizadas pelo programa. Abaixo apresentamos a lista dos eventos possíveis, indicando os momentos em que os mesmos podem ocorrer, bem como, os objetos passíveis de sua ocorrência.

onload - Ocorre na carga do documento. Ou seja, só ocorre no BODY do documento.

onunload - Ocorre na descarga (saída) do documento. Também só ocorre no BODY.

onchange - Ocorre quando o objeto perde o focus e houve mudança de conteúdo.

válido para os objetos Text, Select e Textarea.

onblur - Ocorre quando o objeto perde o focus, independente de ter havido mudança.

válido para os objetos Text, Select e Textarea.

onfocus - Ocorre quando o objeto recebe o focus.

válido para os objetos Text, Select e Textarea.

onclick - Ocorre quando o objeto recebe um Click do Mouse.

válido para os objetos Buton, Checkbox, Radio, Link, Reset e Submit.

onmouseover - Ocorre quando o ponteiro do mouse passa

por sobre o objeto.

válido apenas para Link.

onselect - Ocorre quando o objeto é selecionado.

válido para os objetos Text e Textarea.

onsubmit - Ocorre quando um botão tipo Submit recebe um click do mouse.

válido apenas para o Form.

## **Criando Variáveis**

A variável é criada automaticamente, pela simples associação de valores à mesma.

Ex. NovaVariavel = "Jose"

Foi criada a variável de nome NovaVariavel que, passou a conter a string Jose.

## **Escrevendo no Documento**

O JavaScript permite que o programador escreva linhas dentro de uma página (documento), através do método write. As linhas escritas desta forma, podem conter textos, expressões JavaScript e comandos Html. As linhas escritas através deste método aparecerão no ponto da tela onde o comando for inserido.

Ex:

```
<script>
valor = 30
document.write ("Minha primeira linha<br>")
document.write ("Nesta linha aparecerá o resultado de : " +
(10 * 10 + valor))
</script>
```

A idéia do exemplo acima é escrever duas linhas. Para evitar que as duas linhas sejam escritas na mesma linha é utilizada a tag

Html <br> para dar uma quebra de linha.

## Mensagens

Existem três formas de comunicação com o usuário através de mensagens.

APENAS OBSERVAÇÃO:

alert ( mensagem )

Ex.

alert ("Certifique-se de que as informações estão corretas")

MENSAGEM QUE RETORNA CONFIRMAÇÃO DE OK OU CANCELAR:

confirm (mensagem)

Ex.

if (confirm ("Algo está errado...devo continuar?"))

{ alert("Continuando") }

else

{ alert("Parando") }

RECEBE MENSAGEM VIA CAIXA DE TEXTO INPUT:

Receptor = prompt ("Minha mensagem", "Meu texto")

Onde:

Receptor é o campo que vai receber a informação digitada pelo usuário.

Minha mensagem é a mensagem que vai aparecer como Label da caixa de input.

Meu texto é um texto, opcional, que aparecerá na linha de digitação do usuário.

Ex.

```
Entrada = prompt("Informe uma expressão matemática", "")
Resultado = eval(Entrada)
document.write("O resultado é " + Resultado)
```

## Criando Funções

Uma função é um set de instruções, que só devem ser executadas quando a função for acionada.

A sintaxe geral é a seguinte:

```
function NomeFunção (Parâmetros)
{ Ação }
```

Suponha uma função que tenha como objetivo informar se uma pessoa é maior ou menor de idade, recebendo como parâmetro a sua idade:

```
function Idade (Anos) {
  if (Anos > 17)
  { alert ("Maior de Idade") }
  else
  { alert ("menor de Idade") }
}
```

Para acionar esta função, suponha uma caixa de texto, em um formulário, na qual seja informada a idade e, a cada informação, a função seja acionada.

```
<form>
<input type=text size=2 maxlength=2 name="Tempo"
onchange="Idade(Tempo.value)">
</form>
```

Observe que o parâmetro passado (quando ocorre o evento "onchange") foi o conteúdo da caixa de texto "Tempo" (propriedade "value") e que, na função, chamamos de "Anos". Ou seja, não existe



co-relação entre o nome da variável passada e a variável de recepção na função. Apenas o conteúdo é passado.

## **Interagindo com o usuário**

A interação com o usuário se dá através de objetos para entrada de dados (textos), marcação de opções (radio, checkbox e combo), botões e link's para outras páginas.

Veja abaixo a lista de eventos relacionados às tags HTML:

### **OBJETO INPUT TEXT**

Os eventos associados a este objeto são: onchange, onblur, onfocus e onselect.

Ex:

```
<form name="TText">  
  <p>Entrada de Texto <input type=text size=20 maxlength=30  
name="CxTexto" value="" onchange="alert ('Voce digitou ' +  
CxTexto.value)">  
  </p>  
</form>
```

### **OBJETO INPUT PASSWORD**

Os eventos associados a este objeto são: onchange, onblur, onfocus e onselect.

Ex:

```
<form name="TPassword">  
  <p>Entrada de Senha<input type=password size=10  
maxlength=10 name="Senha" value="">  
  </p>  
</form>
```

## OBJETO INPUT CHECKBOX

O único evento associado a este objeto é onclick.

## OBJETO INPUT RADIO

O único evento associado a este objeto é onclick.

Ex:

```
<p>Radio</p>
<p> <input type=radio name="Rad" value="1"
onclick="document.bgColor='green'"> Fundo Verde
<input type=radio name="Rad" value="2"
onclick="document.bgColor='blueviolet'"> Fundo Violeta
<input type=radio name="Rad" value="3"
onclick="document.bgColor='#FFFF00'"> Fundo Amarelo
</p>
```

## OBJETO INPUT BUTTON

Este objeto tem por finalidade criar um botão ao qual se possa atrelar operações lógicas, a serem executadas quando o mesmo receber um click.

O único evento associado a este objeto é onclick.

Ex.

```
<p>
<form method="POST" name="TstButton">
  Digite um Texto <input type=text size=30 maxlength=30
name="Teste" value="">
</p><p>
  Click no Botao <input type=button name="Bteste" value="Botao
de teste"
  onclick="alert ('Voce digitou: ' + TstButton.Teste.value)">
</p>
</form>
```

### OBJETO INPUT SUBMIT

Este objeto é um botão que tem por finalidade submeter (enviar) o conteúdo dos objetos do formulário ao servidor.

O único evento associado a este objeto é onclick.

### OBJETO TEXTAREA

Os eventos associados a este objeto são: onchange, onblur, onfocus e onselect.

### OBJETO SELECT

Para utilização deste objeto é importante o conhecimento de outras propriedades associadas:

Objeto.length : Retorna a quantidade de opções existentes na lista

Objeto.selectedIndex : Retorna o "index" do objeto selecionado (primeiro = 0)  
Objeto.options[index].text : retorna o texto externo associado a cada opção  
Objeto.options[index].value : retorna o texto interno (value) associado a cada opção  
Objeto.options[index].selected : retorna verdadeiro ou falso

Os eventos associados a este objeto são: onchange, onblur e onfocus.

## **Abrindo novas janelas**

Mostraremos agora como podem ser abertas novas janelas sobre uma janela contendo o documento principal.

A sintaxe geral deste método é a seguinte:

Variavel = window.open ("Url", "Nome da janela", "Opções")

Onde:

Variavel - Nome que será atribuído como propriedade da janela.

Url - Endereço Internet onde a janela será aberta. Normalmente voce estará utilizando a sua própria Url, neste caso, preencha com "".

Nome da Janela - É o nome que aparecerá no top da janela (Título)

Opções - São as opções que definem as características da janela, quais sejam:

- toolbar - Cria uma barra de ferramentas tipo "Back", "Forward", etc.

- location - Abre a barra de location do browse

- directories - Abre a barra de ferramentas tipo "What's New", "Handbook", etc.

- status - Abre uma barra de status no rodapé da janela

- scrollbars - Abre barras de rolamento vertical e horizontal

- menubar - Cria uma barra de menu tipo "File", "Edit", etc.

- resizable - Permite ao usuário redimensionar a janela

- width - Especifica a largura da janela, em pixels

- height - Especifica a altura da janela, em pixels

Todas as opções (exceto width e height) são booleanas e podem ser setadas de duas formas. Exemplo: "toolbar" ou "toolbar=1") são a mesma coisa. Se nada for especificado, entende-se que todas as opções estão ligadas; Caso seja especificada qualquer opção, será entendido que estão ligadas apenas as opções informadas.

As opções devem ser informadas separadas por vírgula, sem espaço entre elas.

## **Abrindo um Documento**

Para abrir um documento dentro da janela, deve ser utilizado o seguinte método:

Variavel.document.open()

Onde "Variavel" é o nome da variável associada ao método window.open

#### ESCREVENDO NO DOCUMENTO

Para escrever a tela no documento, deve ser utilizado o seguinte método:

Variavel.document.write ("Comandos html, Comandos JavaScript, Textos, etc.")

#### FECHANDO A JANELA

Para fechar a janela, utilize o seguinte método:

Variavel.document.write ("window.close()")

#### FECHANDO O DOCUMENTO

Para fechar o documento, utilize o seguinte método:

Variavel.document.close ()

A seguir, apresentamos um exemplo no qual estamos abrindo um documento onde o usuário ao clicar no botão "abre" é então aberta uma outra página.

```
<script>
function abrindo() {
  window.open ("", "OutraJanela",
'scrollbars=yes,resizable=yes,width=650,height=550')
  document.write ("<html><head><title>Essa é outra janela")
  document.write ("</title></head><body bgcolor='#FF00FF'>")
  document.write ("<br>ESSA É UMA OUTRA
PÁGINA<br><br>")
  document.write ("")
  document.write ("<br>Java Script<br><br>")
```

```
document.write ("</body></html>")
document.close() }
</script>
<body bgcolor="#FFFF00">
<p></p>
<p>Ao clicar no botão uma nova página irá abrir!</p>
<form method="POST" name="Form1">
<p></p>
<input type="button" name="abre" value="Nova Janela"
onclick="abrindo()">
</p>
</form>
</body>
```



## Anotações



## **Anotações**

## **Anotações**

## **Anotações**

## Anotações

## **Anotações**

## Anotações

## **Anotações**

## Anotações



## **Anotações**

## Anotações



