



**CENTRO DE TREINAMENTO
EM INFORMÁTICA**

APOSTILA DE HTML



**FUNDAÇÃO
DES. PAULO FEITOZA**

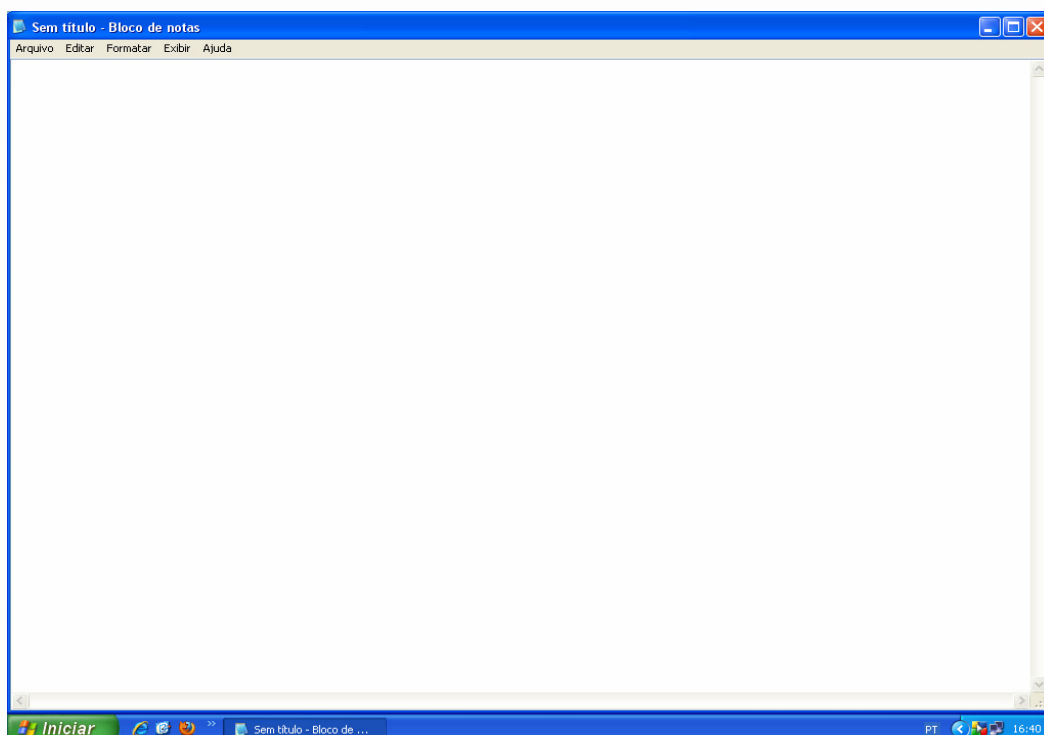
1ª AULA

- ☞ Assuntos a serem Abordados:
 - 📄 Notepad (Bloco de Notas) e Notepad++
 - 📄 O que é HTML?
 - 📄 Marcações HTML (tags)
 - 📄 Estrutura de uma página HTML
 - 📄 O que são Cabeçalhos?
 - 📄 Parágrafos e Quebra de Linha
 - 📄 Elementos e Atributos de Páginas
 - 📄 Alinhamentos de parágrafo
 - 📄 Criando uma Divisão
 - 📄 Texto Pré-Formatado
 - 📄 Verificação de Aprendizagem (Ver o Caderno de Exercícios)

1. Notepad(loco de Notas) e Notepad++

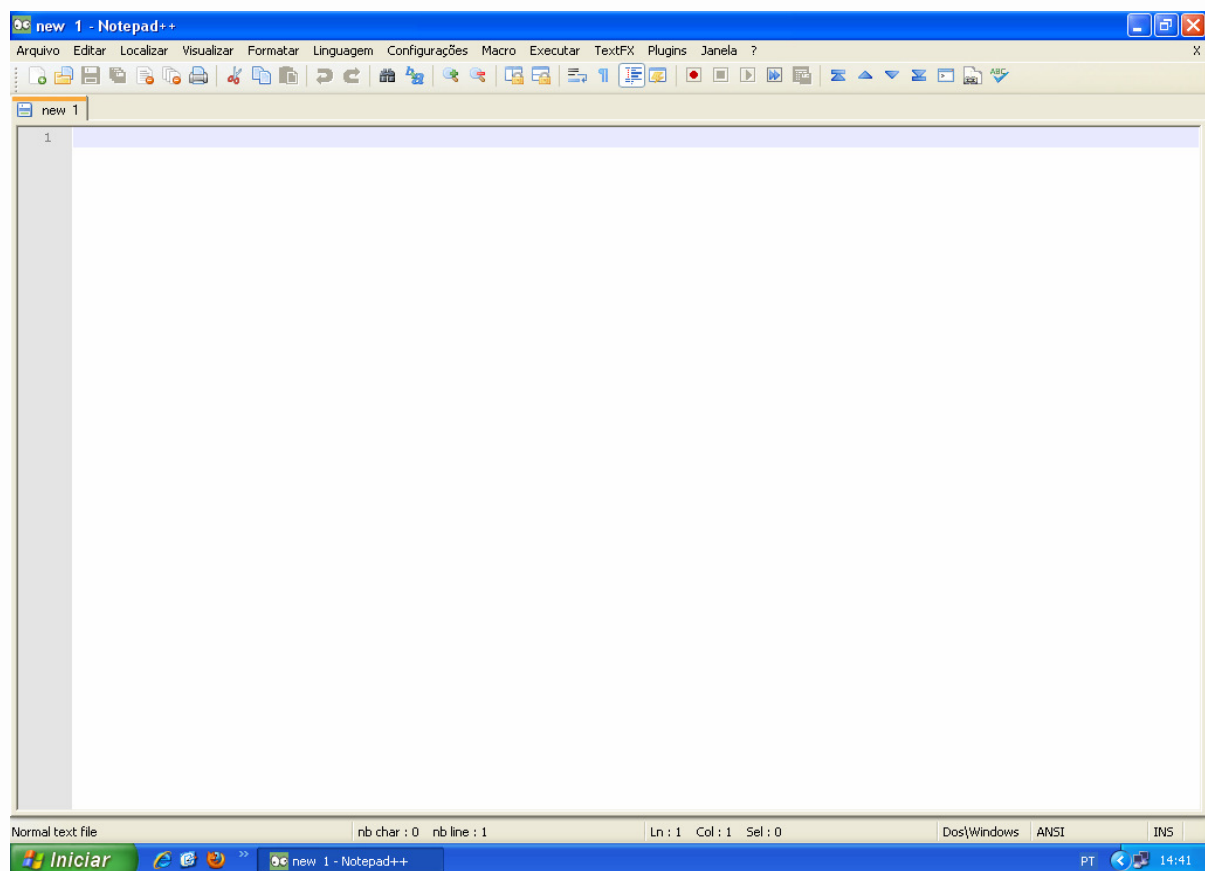
Notepad(Bloco de Notas)

O **Notepad** (ou **Bloco de notas** em português) é um [editor de textos](#) básico que pode ser utilizado para criar documentos simples. O uso mais comum do Bloco de notas é exibir ou editar [arquivos de texto \(.txt\)](#), mas muitos usuários o consideram uma ferramenta simples para criar [páginas da Web](#). Como o Bloco de notas oferece suporte apenas a uma formatação muito básica, não é possível salvar acidentalmente uma formatação especial em documentos que devem permanecer como texto puro. Isso é especialmente útil ao criar documentos [HTML](#) para uma página da [Web](#), uma vez que os caracteres especiais ou outra formatação não podem aparecer na página da Web publicada, pois podem até causar erros.



Notepad++

O **Notepad++** é um [editor de texto](#) que suporta várias linguagens de programação rodando sob o sistema [Microsoft Windows](#). O objetivo do Notepad++ é oferecer um esguio e eficiente binário com uma [interface gráfica](#) totalmente modificável. Além disto, usuários podem definir suas próprias linguagens usando um "sistema de definição de linguagem" integrado, que faz do Notepad++ extensível, para ter [colorização de sintaxe](#) e compactação de trechos de código. O Notepad++ não é apenas um editor de código, mas também uma alternativa ao [bloco de notas](#) graças a sua velocidade.



2. O QUE É HTML?

HTML: Hyper Text Markup Language (Uma linguagem de Marcação de Texto)

A HTML é uma linguagem de marcação. Criar um documento em uma linguagem de marcação significa que você começa com o texto da sua página e inclui tags especiais no início e no final de determinadas palavras ou parágrafos.

Características Gerais do HTML

- ✓ Documentos HTML são arquivos escritos em ASCII - texto.
- ✓ Podem ser criados em qualquer editor de texto (wordpad, edit, notepad).
- ✓ Existem editores específicos para várias plataformas.

- ✓ Existem conversores de vários formatos, por exemplo, doc para html.
- ✓ A unidade mínima de informação é a “página”.

Há diferenças entre os diversos clientes Web, de forma que nem todas as marcações e seus correspondentes recursos são suportados por todos eles. Quando um cliente não entende uma determinada marcação, ele simplesmente a ignora. Ao criar um documento, é importante testá-lo com vários clientes.

3. MARCAÇÕES HTML (Tags)

Marcações HTML – conhecidas como tags – são os sinais de **menor que e maior que** ('**<**' e '**>**'), e tudo que estiver entre eles. As tags são partes importantes em documentos HTML, elas são as responsáveis por toda a formatação do texto. Com elas você pode dividir o documento em área de cabeçalho e corpo, colocar títulos, criar tabelas, divisões, formulários e ainda formatar trechos específicos do texto. Entre os sinais **< >** são especificados os comandos propriamente ditos. No caso de tags que necessitam envolver um texto, sua finalização deve ser feita usando-se a barra de divisão **/** , indicando que a tag está finalizando a marcação de um texto. Há exceções a esse funcionamento em pares das marcações. Por exemplo, a que indica um final de parágrafo: **<p>** . Não necessita de uma correspondente: **</p>**. A marcação que indica quebra de linha - **
** - também não precisa de uma correspondente, e outras tais como **<hr>** e ****.

Exemplo de tag : <nome da tag> <i>texto</i> </nome da tag>
--

Assim como outras linguagens, HTML possui uma estrutura básica para seus programas. Para que um browser interprete corretamente o programa, ele deve possuir alguns comandos básicos que sempre deverão estar presentes. Alguns browsers até dispensam seu uso, porém é melhor assumir tais comandos como parte fundamental do programa.

INFORMAÇÕES ÚTEIS

- Uma consideração a ser feita em HTML sobre as tags, é que elas não são sensíveis à caixa, ou seja, a notação **<title>** é equivalente a **<TITLE>**, embora as convenções atuais incentivem o uso de todas as tags em letras minúsculas. Então, para seguir os mais novos padrões web você deve usar letras minúsculas.
- Uma outra questão que pode surgir é sobre a extensão dos arquivos. Este fato é uma herança dos tempos antigos (no que diz respeito à Internet) do **MS-DOS** e do **Windows 16 bits**, em que os nomes dos arquivos tinham no máximo 8 caracteres e as suas extensões não podiam ter mais de 3 caracteres, e por isso era comum ver arquivos com nomes como **'index.htm'**. Essas deficiências, que no passado obrigaram a usar a extensão **.htm** em vez de **.html** já foram eliminadas, logo é totalmente aceitável usar **'index.html'** ou **'index.htm'**.

4. ESTRUTURA DE UMA PÁGINA HTML

Uma página HTML é dividida em duas partes: a **cabeça** e o **corpo**.

Na **cabeça (ou cabeçalho)** são definidos os atributos principais do documento, como o título e as palavras-chave. O **corpo** contém a parte visível do documento, sendo aquela que você verá processada em seu navegador.

Estrutura Básica de uma Página HTML

```
<html>
<head>
<title> Título da Página </title>
</head>
<body>
Esta é minha primeira página em HTML.
</body>
</html>
```

A primeira tag em seu documento HTML é **<html>**. Esta tag define o início de um documento HTML e indica ao navegador que todo conteúdo posterior deve ser tratado como uma série de códigos HTML. A última tag em seu documento deverá ser **</html>**. Esta tag indica ao navegador que é o fim de seu documento HTML.

A tags **<head> ... </head>** representam a informação do cabeçalho. Nenhuma informação contida no cabeçalho é exibida na janela do navegador. Estes comandos para cabeçalho são opcionais, isto é, um programa HTML pode funcionar sem eles. Mas é conveniente usá-los, pois o título da página é acrescentado através deles.

A tags **<title> ... </title>** representam o título de seu documento, ou seja, o texto que estiver contido entre estas tags será o título e será exibido na legenda do navegador, na parte de cima do browser.

O texto entre as tags **<body> ... </body>** são as informações que serão exibidas na página. Estas tags definem o corpo de uma página HTML.

5. O QUE SÃO CABEÇALHOS?

Os cabeçalhos (também chamados de **Headings**) servem para criar títulos diferenciar as seções da sua página. Cabeçalhos são exibidos em letras maiores e em negrito. O primeiro cabeçalho em cada documento deve estar marcado como **<H1>**. Eles possuem seis valores diferentes, sendo que a de valor **1** é a que possui a maior fonte e a de valor **6** possui a menor fonte..

Por exemplo, usando o seguinte código:

```
<h1> Título 1 </h1>
<h2> Título 2 </h2>
<h3> Título 3 </h3>
<h4> Título 4 </h4>
<h5> Título 5 </h5>
<h6> Título 6 </h6>
```

No código acima, o HTML automaticamente solta uma linha em branco entre um Título e outro.

Tome cuidado ao definir o tamanho de um cabeçalho, você não está definindo o tamanho da letra (fonte 10, fonte 14). Você apenas define que ele aparecerá com maior tamanho e destaque que o resto do texto.

6. PARÁGRAFOS E QUEBRA DE LINHA

PARÁGRAFOS

Para começar um novo parágrafo, ou seja, avançar uma linha em branco e iniciar o texto na segunda linha após o final do parágrafo anterior deve ser usado o comando **<p>**. Esse comando deverá aparecer em par: **<p> </p>**.

QUEBRA DE LINHA

A tag **
** é utilizada quando você quer terminar uma linha, mas não quer começar um novo parágrafo. Com este comando você faz com que ocorra uma quebra de linha, onde você posicionar a tag. Note que para o elemento **
** não existe o comando **</br>**, isto é, a quebra de linha não age numa região de texto delimitada, mas sim num ponto do texto.

INFORMAÇÕES ÚTEIS

Sempre que você quiser inserir linhas em branco, utilize a tag **
. Existem pessoas que utilizam parágrafos vazios para obter o mesmo resultado, mas isso está errado. A tag **<p> deve ser usada apenas para definir parágrafos, e o elemento **
** não deve ser usado, por exemplo, para criar listas, pois existem tags concebidas especificamente para isso. Sempre que você precisar obter uma formatação especial, você deve usar o elemento que foi criado para esse efeito.

7. ELEMENTOS E ATRIBUTOS DE PÁGINAS

Elementos HTML são definidos usando **tags de abertura**, **conteúdo** e **tag de fechamento**. Tudo o que se encontrar entre as **tags de abertura** e de **fechamento** fazem parte do **conteúdo do elemento**. Uma página HTML é composta basicamente de títulos, textos, parágrafos, imagens e links, responsáveis pela chamada de outras páginas para a tela. Todos esses elementos são posicionados na página por meio de comandos específicos de linguagem.

✓ Título

O título de uma página web indica qual o assunto abordado e irá aparecer na barra de título do browser. Você poderá ter apenas um título. Este título é utilizado por programas de lista (**hotlist**) do seu navegador e também por outros programas que catalogam páginas da Internet. Para atribuir um título à página você deverá utilizar a tag: **<title> </title>**. Esta tag sempre será incluída no cabeçalho (entre as tags **<head> </head>**).

✓ Imagem

São figuras, desenhos e fotos usados para ilustrar a página.

✓ Texto

É a informação mais comum dentro da página. Pode ser formatado através de vários comandos.

Atributos HTML, definem características adicionais aos elementos, informam ao navegador a respeito das propriedades que os elementos podem assumir. Tais atributos devem ser colocados sempre na **tag de abertura** e nunca na **tag de fechamento**, logo após o nome do elemento, precedido de um espaço e é composto de um nome de atributo, um sinal de igual (=) e um valor de atributo, cercado por aspas duplas (") ou simples (').

No HTML, os valores dos atributos podem ser definidos da forma: **atributo=valor**

Assim como existem muitas tags, também existem muitos atributos. Alguns atributos são empregados em tags específicas, enquanto outros servem para várias tags e vice-versa. Algumas tags podem conter somente um tipo de atributo, enquanto outras podem conter vários tipos.

Para facilitar o entendimento de como utilizar atributos, veremos nas seções seguintes, atributos que serão aplicados a vários elementos.

8. ALINHAMENTOS DE PARÁGRAFO

Você já deve ter visto que em certos documentos o texto aparece ora alinhado à direita, ora à esquerda, no centro ou então ocupando uniformemente o espaço da página (texto justificado). Isso pode ser obtido facilmente em HTML. Veja a tabela abaixo:

<p align="left ">	Alinha o texto à esquerda
<p align="right ">	Alinha o texto à direita
<p align="center ">	Alinha o texto centralizado
<p align="justify ">	Alinha o texto justificado

Note que entre os delimitadores < e > não encontra-se apenas o elemento <p>. Além dele, existe o texto **align="alinhamento"**. Dizemos que **align** é um atributo do elemento **p** e alinhamento é o valor desse atributo. No caso de **<p align="justify">**, o valor do atributo **align** (que significa alinhamento em inglês) é **justify** (justificado).

9. CRIANDO UMA DIVISÃO

Existem momentos em que queremos que vários parágrafos possuam um mesmo valor de atributo - centralizado, por exemplo. Ao invés de escrevermos **align="justify"** a cada abertura de novo parágrafo, podemos usar o elemento **div**, que cria uma "divisão" no documento na qual alguns atributos são preservados. Suas vantagens sobre o **align** é que:

- ✓ Precisar ser utilizada apenas uma vez, ao contrário do atributo **align**, que tem de ser incluído em diversas tags.
- ✓ A tag **<div>** pode ser usada para alinhar qualquer elemento (cabeçalho, parágrafos, citações, imagens, tabelas, etc.). O atributo **align** encontra-se disponível apenas em um número limitado de tags.

Você deverá colocar a tag **<div>** e acrescentar o atributo **align** na tag de abertura. O atributo **align** poderá ter os valores: **left**, **right**, **justify** e **center**.

Vejamos o exemplo:

```
<div align="center ">  
<p>  
Texto 1  
Texto 2  
Texto 3  
</p>  
<p>  
Texto 4  
Texto 5  
Texto 6  
</p>  
</div>
```

10. TEXTO PRÉ-FORMATADO

O texto de um arquivo em HTML é formatado através das tags. Mas você possui a opção da tag **<pre>** **</pre>** para estabelecer uma formatação através de um editor de texto e que esta formatação permaneça na sua página html.

O texto pré-formatado é excelente para apresentar código-fonte com seus espaçamentos adequados. Este recurso pode ser utilizado para criar tabelas mas não é recomendável.

Uma observação importante é que, ao usar tags de estilo, não tags de elemento, neste espaço elas irão funcionar normalmente.

SINTAXE:

```
<PRE>  
    TEXTO  
        TEXTO  
            TEXTO  
</PRE>
```

11. VERIFICAÇÃO DE APRENDIZAGEM (VER O CADERNO DE EXERCÍCIOS)

2ª AULA

🔗 Assuntos a serem Abordados:

- 📄 Tags de Formatação
- 📄 Formatação de Fonte
- 📄 Entidades, Acentuação e Caracteres Especiais
- 📄 Tabelas de Caracteres Especiais
- 📄 Linhas Horizontais
- 📄 Efeitos de Movimento
- 📄 **Verificação de Aprendizagem (Ver o Caderno de Exercícios)**

1. TAGS DE FORMATAÇÃO

Assim como em um editor de texto, em HTML você poderá utilizar vários elementos para se formatar um texto, como por exemplo, escrever em **negrito**, *itálico*, sublinhado, etc.

TAG	SINTAXE	FUNÇÃO
NEGRITO	texto	Aplica o estilo negrito
ITÁLICO	<i>texto</i>	Aplica o estilo <i>itálico</i>
SUBLINHADO	<u>texto</u>	Aplica o estilo <u>sublinhado</u> (em alguns browsers esta tag não funciona)

Os principais comandos de estilo:

TAG	SINTAXE	FUNÇÃO
STRONG	texto	Similar ao negrito
TYPewriter	<tt>texto</tt>	Deixa o texto com espaçamento regular
BIG	<big>texto</big>	Aumenta a fonte e aplica Negrito
SMALL	<small>texto</small>	Reduz e altera a fonte
SOBRESCRITO	^{texto}	Eleva o texto e diminui seu corpo
SUBESCRITO	_{texto}	Rebaixa o texto e diminui seu corpo

2. FORMATAÇÃO DE FONTE

Para fazer alterações no **tamanho**, **tipo** e **cor** da fonte de seu texto você utilizará a tag . Sua sintaxe é a seguinte:

texto

Size - especifica o tamanho da fonte a ser utilizada para o texto. Os valores permitidos vão de 1, o menor, a 7, o maior. Sendo 3 o valor padrão.

Face - especifica o tipo de fonte a ser escolhida para o texto.

Color - especifica a cor do texto através de um nome predefinido de cores. No entanto, podemos obter qualquer cor que desejarmos combinando proporções corretas de três cores bases: **Vermelho (Red)** , **Verde (Green)** e **Azul (Blue)**. Seu valor será especificado no formato hexadecimal.








Desta forma, é possível exprimir cores usando três números hexadecimais que definem as quantidades de vermelho, verde e azul que entram na composição de uma determinada cor. O valor mais baixo de uma determinada cor é 0 (#00 em código hexadecimal) e o valor mais alto é 255 (#FF em código hexadecimal). Assim, a cor preta tem 0 de vermelho, 0 de verde e 0 de azul, escrevendo na forma #000000. Já o branco possui 255 de vermelho, 255 de verde e 255 de azul, sendo seu código #FFFFFF. O amarelo forte possui 255 de vermelho, 255 de verde e 0 de azul, sendo seu código #FFFF00. A tabela abaixo mostra os resultados de diversas combinações de cores:

TABELA DE CÓDIGOS DAS CORES:

Nome da Cor	RGB Hexadecimal	Cor
Amarelo	#FFFF00	
Amarelo Esverdeado	#99CC32	
Aquamarine	#70DB93	
Aquamarine Médio	#32CD99	
Azul	#0000FF	
Azul Ardósia	#007FFF	
Azul Ardósia Escuro	#6B238E	
Azul Brilhante	#C0D9D9	
Azul Celeste	#3299CC	
Azul Claro	#5F9F9F	
Azul Corn Flower	#42426F	
Azul Escuro	#00009C	
Azul Marinho	#23238E	
Azul Médio	#3232CD	
Azul Neon	#4D4DFF	
Azul Rich	#5959AB	
Azul Violeta	#9F5F9F	
Baker's Chocolate	#5C3317	
Branco	#FFFFFF	
Bright Ouro	#D9D919	

Bronze	#DB9370	
Bronze Claro	#EBC79E	
Bronze Escuro	#8C7853	
Caqui	#9F9F5F	
Cinza	#C0C0C0	
Cinza Brilhante	#A8A8A8	
Cinza Claro	#CDCDCD	
Cobre	#D98719	
Cobre Claro	#B5A642	
Cobre Escuro	#B87333	
Coral	#FF7F00	
Cyan	#00FFFF	
Escarlata	#8C1717	
Firebrick	#8E2323	
Goldenrod Médio	#EAEAAE	
Goldenrod	#DBDB70	
Laranja	#FF7F00	
Laranja Mandarin	#E47833	
Light Steel Blue	#8F8FBD	
Light Wood	#E9C2A6	
Magenta	#FF00FF	
Marrom	#8E236B	
Marrom	#A62A2A	
Marrom Escuro	#5C4033	
Medium Slate Blue	#7F00FF	
Orquídea Escuro	#9932CD	
Orquídea Médio	#9370DB	
Ouro	#CD7F32	
Ouro Velho	#CFB53B	
Plum	#EAADEA	

Preto	#000000	
Púrpura Escuro	#871F78	
Quartz	#D9D9F3	
Rosa	#FF6EC7	
Rosa Temperado	#FF1CAE	
Salmão	#6F4242	
Semi-Sweet Chocolate	#6B4226	
Sienna	#8E6B23	
Silver	#E6E8FA	
Steel Azul	#236B8E	
Summer Sky	#38B0DE	
Tan Escuro	#97694F	
Thistle	#D8BFD8	
Turquesa	#ADEAEA	
Turquesa Escuro	#7093DB	
Turquesa Médio	#70DBDB	
Verde	#00FF00	
Verde Amarelado	#93DB70	
Verde Cobre	#527F76	
Verde Cobre Escuro	#4A766E	
Verde Escuro	#2F4F2F	
Verde Floresta	#238E23	
Verde Floresta Médio	#6B8E23	
Verde Hunter	#215E21	
Verde Limão	#32CD32	
Verde Oceano	#426F42	
Verde Oceano claro	#238E68	
Verde Oliva Escuro	#4F4F2F	
Verde Primavera	#7FFF00	
Verde Primavera	#00FF7F	

Vermelho	#FF0000	
Vermelho Indiano	#4E2F2F	
Vermelho Violeta	#CC3299	
Violeta	#4F2F4F	
Violeta Vermelho Médio	#DB7093	
Wheat	#D8D8BF	
Wood Médio	#A68064	

3. ENTIDADES, ACENTUAÇÃO E CARACTERES ESPECIAIS

Alguns caracteres em HTML são chamados de **caracteres reservados** que quando aparecem num documento HTML são interpretados pelo navegador.

Para acessar um caractere, você precisa usar a seguinte convenção: **&nome-especial;**

Onde o **nome-especial** é o nome ou um número associado ao símbolo gráfico que você deseja mostrar no seu documento.

Há caracteres que não podem ser obtidos por meio do teclado. Eles podem ser inseridos em sua página através da especificação de um código especial que o browser interpreta e substitui por um caractere específico. Esses códigos podem ser obtidos a partir de uma entidade HTML.

A norma ISO utiliza um código composto pelo caractere **&** seguido do símbolo **#** (cerquilha) e, então, de uma combinação numérica com **ponto-e-vírgula**. Já a entidade HTML usa o caractere **&** seguido de uma palavra que identifica o símbolo com **ponto-e-vírgula**.

Por exemplo, e você utilizar a palavra **café** com a acentuação do seu teclado, com certeza na visualização do seu browser ficará perfeita, mas pode ser que em outros computadores com a configuração diferente não apareça da mesma forma.

O ideal é colocá-la assim: **café** ou **caé**

Ex: **Diário** = diário

Natação = Natação

Sequência = sequência

E se desejasse colocar a palavra entre aspas: **"café"**

4. TABELAS DE CARACTERES ESPECIAIS

Símbolos de Marcação

Código	Símbolo	Código Numérico	Símbolo
"	"	"	"
&	&	&	&
<	<	<	<
>	>	>	>

Símbolo de Espaço

Código	Símbolo	Código Numérico	Símbolo
 		 	

Currency Symbols

Código	Símbolo	Código Numérico	Símbolo
¢	¢	¢	¢
£	£	£	£
¤	¤	¤	¤
¥	¥	¥	¥

Símbolos de Pontuações

Código	Símbolo	Código Numérico	Símbolo
¡	!	¡	!
«	«	«	«
»	»	»	»
¿	¿	¿	¿
–	–	–	–
—	—	—	—
‘	‘	‘	‘
’	’	’	’
‚	,’	‚	,’
“	“	“	“
”	”	”	”
„	„	„	„
‹	‹	‹	‹
›	›	›	›

INFORMAÇÕES ÚTEIS

Caso você queira escrever uma página HTML em outro idioma terá que usar bastante esses caracteres especiais. Quando você utiliza a acentuação comum do teclado nas suas páginas podem até parecer que tudo irá correr tudo bem. Mas alguém que visualizar suas páginas em um computador que não tenha as mesmas configurações de vídeo e teclado, alguns caracteres podem não sair da forma desejada.

5. LINHAS HORIZONTAIS

A tag `<hr>` insere uma linha horizontal. Ela não possui tag de fechamento e é utilizada para auxiliar na distinção de blocos de conteúdo.

Essa linha possui vários atributos, oferecendo resultados diversos. Os atributos irão definir as propriedades que a linha irá conter, como largura, cor, posição e comprimento.

`<hr size=7>` insere uma linha de largura 7 (pixels).

`<hr color="red">` insere uma linha com cor vermelha.

`<hr align="left">` insere uma linha alinhada à esquerda.

`<hr width="50%">` insere uma linha de 50% de comprimento.

Você poderá utilizar os atributos separados ou juntos em uma linha horizontal.

`<hr size="7" color="red" align="left" width="50%">` insere uma de largura 7, com cor vermelha, alinhada à esquerda, de 50% de comprimento.

INFORMAÇÕES ÚTEIS

Para você utilizar os atributos de alinhamento em uma linha horizontal, você deve atentar para o comprimento da linha, ou seja, o espaço disponível que ela está ocupando.

Você poderá posicionar uma linha horizontal à direita, à esquerda e no centro, porém, se desejar posicionar na parte inferior da janela não será possível utilizar atributos de alinhamento, para isso, você poderá fazer uso da tag `
`.

6. EFEITOS DE MOVIMENTO

É possível obter o efeito de animação de texto, através da tag `<marquee>`.

Sintaxe: `<marquee behavior="efeito">Texto</marquee>`

Atributos:

scroll - o letreiro continuará rolando continuamente em um mesmo sentido.

`<marquee behavior="scroll">Texto</marquee>`

slide - o letreiro parte de um ponto da tela e vai até o outro, e pára quando chega ao final do seu percurso.

`<marquee behavior="slide">Texto</marquee>`

alternate - o letreiro deslizará até o outro lado da tela, e quando chegar lá, ele voltará pela tela novamente indo e voltando.

`<marquee behavior="alternate">Texto</marquee>`

Também é possível definir as direções que o texto irá ter em seu percurso através do efeito **direction**.

Sintaxe: `<marquee direction="efeito">Texto</marquee>`

Atributos:

up - o letreiro irá para cima.

`<marquee direction="up">Texto</marquee>`

down - o letreiro irá para baixo.

`<marquee direction="down">Texto</marquee>`

left - o letreiro irá para a esquerda.


`<marquee direction="left">Texto</marquee>`






right - o letreiro irá para a direita.

`<marquee direction="right">Texto</marquee>`

7. VERIFICAÇÃO DE APRENDIZAGEM (VER O CADERNO DE EXERCÍCIOS)

3ª AULA

 Assuntos a serem Abordados:

-  **Listas**
-  **Imagens**
-  **Colocando uma imagem como plano de fundo**
-  **Colocando uma cor no fundo da página**
-  **Verificação de Aprendizagem (Ver o Caderno de Exercícios)**

1. LISTAS

A linguagem HTML contém elementos que permitem criar dois tipos de listas: as **não ordenadas** e as **ordenadas**.

1.2 - Listas Não Ordenadas

Uma lista não ordenada contém vários itens marcados todos com o mesmo símbolo (normalmente um círculo pequeno ou um quadrado pequeno).

Para criar uma lista não ordenada, utilizamos o elemento `` (**unordered list**). Dentro desse elemento, os vários itens são criados com o elemento `` (**list item**).

O exemplo seguinte mostra uma lista simples:

```
<ul>
  <li> Laranja
  <li> Uva
</ul>
```

Este é o aspecto de como vai ficar em seu navegador:

- Laranja
- Uva

Dentro de uma lista não ordenada podemos colocar parágrafos, quebras de linha, imagens, outras listas, etc.

Você poderá personalizar sua lista não ordenada com o atributo **Type**. O atributo **Type** pode assumir três valores (**circle**, **disk** e **square**) para definir o tipo de marcação a ser usado na lista.

Veja o exemplo:

```
<ul type="square">
  <li> Laranja
  <li> Uva
</ul>
```

Agora veja como ficará em seu navegador:

- Laranja
- Uva

1.3 - Listas Ordenadas

Uma lista ordenada contém vários itens numerados e é criada com o elemento `` (**ordered list**). Os itens da lista ordenada definem-se com o elemento `` (**list item**).

```
<ol>
  <li> Abacate
  <li> Abacaxi
</ol>
```

Este é o aspecto de como vai ficar em seu navegador:

1. Abacate
2. Abacaxi

Dentro de uma lista ordenada podemos colocar parágrafos, quebras de linha, imagens, outras listas, etc.

Você poderá personalizar sua lista ordenada com o atributo **Type** e **Start**. O atributo **Type** pode assumir cinco valores (“ 1 ” , “ a ” , “ A ” , “ i ” , “ I ”) para definir o tipo de numeração a ser usado na lista ordenada.

"1" - Especifica que os algarismos arábicos padrão (1, 2, 3, 4, ...) serão usados para numerar a lista.

"a" - Especifica que as letras minúsculas (a, b, c, d, ...) serão usadas para numerar a lista.

"A" - Especifica que as letras maiúsculas (A, B, C, D, ...) serão usadas para numerar a lista.

"i" - Especifica que os algarismos romanos minúsculos (i, ii, iii, iv, ...) serão usados para numerar a lista.

"I" - Especifica que os algarismos romanos maiúsculos (I, II, III, IV, ...) devem ser usados para numerar a lista.

Agora um exemplo com o atributo Type:

```
<p>Ingredientes para bolo</p>
```

```
<ol type="a">
```

```
<li>Farinha de trigo
```

```
<li>Açúcar
```

```
<li>Manteiga
```

```
<li>Ovos
```

```
<li>Leite
```

```
<li>Fermento
```

```
</ol>
```

Agora veja como ficará em seu navegador:

Ingredientes para bolo

- a. Farinha de trigo
- b. Açúcar
- c. Manteiga
- d. Ovos
- e. Leite
- f. Fermento

Em uma lista ordenada o ponto inicial é **1**. Usando o atributo **Start** você poderá determinar o número ou a letra que inicia sua lista. Como no exemplo abaixo:

```
<p>Ingredientes para bolo</p>
```

```
<ol type=1 start=3>
```

```
<li>Farinha de trigo
```

```
<li>Açúcar
```

```
<li>Manteiga
```

```
<li>Ovos
```

```
<li>Leite
```

```
<li>Fermento
```

```
</ol>
```

Agora veja como ficará em seu navegador:

Ingredientes para bolo

3. Farinha de trigo
4. Açúcar
5. Manteiga
6. Ovos
7. Leite
8. Fermento

Usando o atributo **Value** em um item da lista ordenada poderemos alterar os valores da lista a partir de qualquer ponto.

Exemplo:

```
<ul>  
<li>  
<li value=10>  
<li>  
</ul>
```

2. IMAGENS

Além de manipular texto, o HTML também serve para exibir imagens de uma maneira muito simples, usando para isso o elemento **img**. As imagens dentro de uma página devem estar preferencialmente no formato **".gif"**. Este é o formato (ou extensão) mais universalmente aceito pelos visualizadores de **WWW (browsers)**. O Internet Explorer consegue decodificar outros formatos como o **".jpg"** e **".rgb"**, mas esta capacidade não é comum a todos os programas visualizadores. Por isso fique atento quando for aplicar certos tipos de formatos (extensões).

Para inserir uma imagem em uma página web utilizamos a tag **** e seus atributos.

**Sintaxe: Width="valor em pixel" | Largura da imagem |
Height="valor em pixel"	Altura da imagem
Align(center, left, right)	Alinhamento da imagem.
Alt="texto alternativo"	Texto que aparecerá ao passar o mouse sobre a imagem
Border="valor em pixel"	Especificação da largura da borda da imagem.
Vspace="valor em pixel"	Indica o espaço que deve ser deixado acima e abaixo da imagem.
Hspace="valor em pixel"	Indica o espaço que deve ser deixado nas laterais da imagem.

O atributo **src - source, (fonte em inglês)** dá o nome do arquivo da imagem e é o único atributo obrigatório para o elemento **img**. Quando seu navegador abre um arquivo html e encontra o

elemento **img**, ele usará o respectivo atributo **src** como o endereço onde está o arquivo de imagem a ser exibido. Por isso você deve organizar as imagens juntamente com os arquivos a serem salvos em um mesmo local.

3. COLOCANDO UMA IMAGEM COMO PLANO DE FUNDO

O atributo **background** estabelece que o padrão de fundo da página será uma imagem. Este atributo é inserido dentro da tag **<body>**. Sendo que o valor desse atributo indica o local onde se encontra a imagem. Se as dimensões da imagem forem inferiores às dimensões da página, o navegador repetirá a imagem (como num chão de mosaicos ou em uma parede de azulejos), por forma a ocupar todo o fundo da página.

O exemplo abaixo mostra bem como se faz isso:

```
<body background="imagem.jpg">  
<body background="c:\site\imagem.jpg">
```

O valor do atributo **background** é uma URL (endereço) que define o local onde se encontra a imagem. Na primeira linha mais acima, demos uma URL relativa e na segunda demos uma URL absoluta.

4. COLOCANDO UMA COR NO FUNDO DA PÁGINA

```
<body bgcolor="cor">
```

O atributo **bgcolor** nos permite escolher a cor de fundo da página, utilizando o código das cores ou até mesmo o nome de cada cor.

5. VERIFICAÇÃO DE APRENDIZAGEM (VER O CADERNO DE EXERCÍCIOS)

4ª AULA

📁 Assuntos a serem Abordados:

- 📄 Links
- 📄 Âncoras
- 📄 Atributos de **<body>**
- 📄 Tabelas – tag **<table>**
- 📄 Atributos da tag **<table>**
- 📄 Verificação de Aprendizagem (Ver o Caderno de Exercícios)

1. LINKS

A principal atração da Internet é a criação de documentos com o conceito de hipertexto, ou seja, um documento que se vincula a outros documentos por meio de ligações especiais chamadas links

ou hiperlinks. Com esse conceito, você pode criar documentos que façam referências e permitam ao usuário acessar tais referências não importando se elas estão em outra página Web, no seu micro ou em algum servidor de rede.

O uso de hipertexto também facilita a criação de documentos extensos cujas seções ou tópicos podem ser rapidamente acessados por meio dos links. Imagine que você crie uma página sobre determinado assunto em vários tópicos abordados. Assim como em um livro você cria todo o texto do documento e depois cria, no início, um sumário indicando os tópicos do documento. Se quiser acessar diretamente o texto de um tópico, basta dar um clique sobre o item do tópico no sumário e ele será imediatamente exibido.

Um link é reconhecido em uma página por estar em cor normalmente diferente do resto do texto e, ao passar com o mouse sobre esse texto o cursor muda para uma mão apontando para o link.

Isso é feito de modo muito simples com o elemento **a**, como segue:

texto que será o link

Exemplo: **Minha casa**

1.2. Usando uma imagem como link

Exemplo: ****

Como você deve ter percebido, o atributo **href (HiperText Reference)** indica o endereço da página que você quer "linkar" (ligar) ao seu documento.

2. ÂNCORAS

Uma âncora é um ponto de referência ou endereço que será acessado por um link, é usada dentro do documento para marcar o início de uma seção do documento.

Suponha que seu texto seja muito grande, o que tornaria trabalhoso para quem estiver vendo a página conseguir se movimentar entre um tópico e outro.

Então faça o seguinte: Nomeie um pedaço da sua página através do tag:

 Texto

Lembre –se que **"nome"** é fictício, você escolhe o nome.

Depois disso escreva o pedaço que você quer associar a esse Nome. Agora crie um link para chegar até esse pedaço, usando o tag:

Clique Aqui

Pronto! Ao clicar na mensagem "**Clique Aqui**" de sua página, o usuário irá até o pedaço que você nomeou.

ATENÇÃO: O sinal de cerquilha (#) é necessário para a âncora, pois avisa ao browser para procurar o link no documento atual.

Uma âncora é um ponto de referência ou endereço que será acessado por um link.

3. ATRIBUTOS DE <BODY>

Através de atributos de <body>, podemos definir cores para os textos, links e para o fundo das páginas, bem como uma imagem de fundo):

```
<body bgcolor="#ffffff" text="#rrggbb" link="#rrggbb" alink="#rrggbb" vlink="#rrggbb" >
```

onde:

- ✓ **bgcolor** : Cor de fundo (padrão: cinza ou branco)
- ✓ **text** : Cor dos textos da página (padrão: preto)
- ✓ **link** : A cor dos links (padrão: azul)
- ✓ **alink** : A cor dos links, **quando acionados** (padrão: vermelho)
- ✓ **vlink** : Cor dos links, **depois de visitados** (padrão: azul escuro ou roxo)

Seus valores podem ser dados em valores hexadecimais equivalentes a cores no padrão RGB (Red, Green, Blue) ou mesmo dando os nomes das cores. Existem tabelas de cores com esses valores, mas grande parte dos editores já oferece uma interface bem amigável através da qual escolhemos as cores desejadas, sem nos preocuparmos com códigos de cores complexos.

4. TABELAS – tag <table>

Tabelas auxiliam na visualização de dados ou na divisão de sua página em setores. Entenda por uma tabela HTML como sendo semelhante a uma tabela que você desenharia numa folha de papel ou num outro programa de computador. O elemento principal de uma tabela é o **table** e aqui segue um exemplo simples de tabela:

```
<table>  
<tr>  
<td>Primeira</td>  
<td>Segunda</td>  
</tr>  
<tr>  
<td>Terceira</td>  
<td>Quarta</td>  
</tr>  
</table>
```

Cujo resultado é:

Primeira Segunda
Terceira Quarta

Em HTML, as tabelas são divididas em linhas e células. Na tabela acima, a primeira linha contém as células **Primeira** e **Segunda**, enquanto que a segunda linha contém as células **Terceira** e **Quarta**.

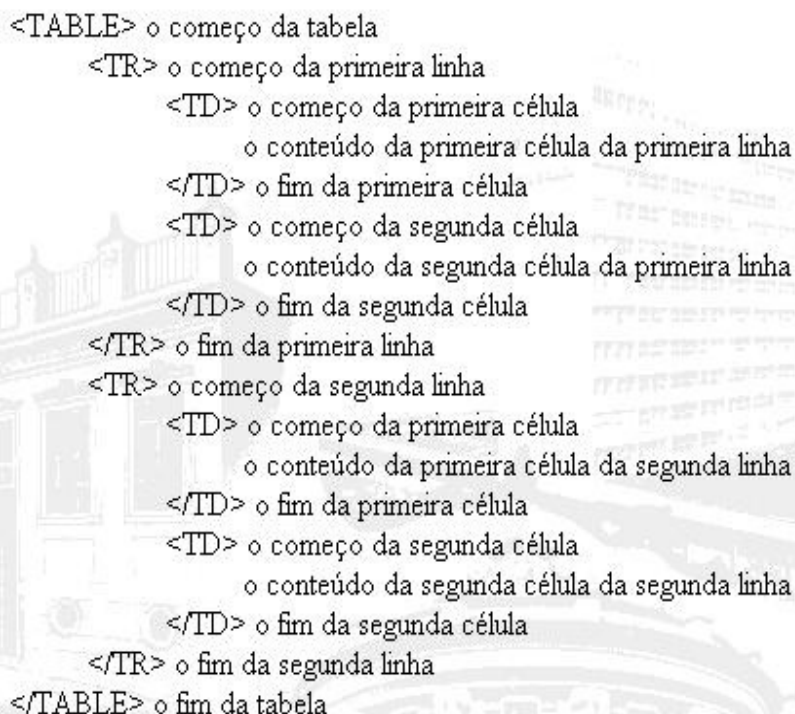
Dentro de um bloco `<table>`, os principais elementos que você utilizará são o `<tr>`(linha) , e o `<td>`(célula) . Dessa forma, podemos definir as linhas e as células da tabela. É obrigatório que um bloco `<td>` esteja definido dentro de um bloco `<tr>`, ou seja, que as células estejam dentro de linhas.

No exemplo anterior, criamos uma linha com `<tr>` e em seguida, dentro de um bloco `<td>`, escrevemos o texto da célula (Primeira). Abrimos outro bloco `<td>` nesse mesmo bloco `<tr>`, onde colocamos o texto da segunda célula (Segunda). Depois, fechamos a linha com o `</tr>` para em seguida abrir uma nova linha, onde criamos as células contendo os textos Terceira e Quarta.

Na figura abaixo, você poderá ver com detalhes o significado de cada elemento que compõe a estrutura básica de uma tabela

• Tabelas

Criação de Tabelas



```
<TABLE> o começo da tabela
  <TR> o começo da primeira linha
    <TD> o começo da primeira célula
      o conteúdo da primeira célula da primeira linha
    </TD> o fim da primeira célula
    <TD> o começo da segunda célula
      o conteúdo da segunda célula da primeira linha
    </TD> o fim da segunda célula
  </TR> o fim da primeira linha
  <TR> o começo da segunda linha
    <TD> o começo da primeira célula
      o conteúdo da primeira célula da segunda linha
    </TD> o fim da primeira célula
    <TD> o começo da segunda célula
      o conteúdo da segunda célula da segunda linha
    </TD> o fim da segunda célula
  </TR> o fim da segunda linha
</TABLE> o fim da tabela
```

Segue abaixo o conjunto de marcações essenciais para desenhar tabelas em HTML. Alguns atributos podem ser definidos para cada uma dessas marcações. Falaremos desses atributos mais adiante.

`<table></table>` - Toda tabela deve ser iniciada com a marcação `<table>` e encerrada com `</table>`.

- ✓ **ATENÇÃO:** Dessa forma a tabela vai aparecer sem bordas, mas dividindo linhas e células.
- ✓ Para que a tabela apareça com bordas simples, defina dessa forma `<table border></table>`.
- ✓ Antes e depois de uma tabela, acontece sempre uma quebra de linha.

`<tr></tr>` - Cada linha de uma tabela deve sempre aparecer entre as marcações `<tr>` e `</tr>`.

`<td></td>` - Esta é a marcação que define cada célula de uma tabela. As células de uma tabela devem sempre aparecer entre as marcações de linhas (`<tr>` e `</tr>`). Como padrão, o texto nas células é alinhado a esquerda.

`<th></th>` - Desta forma são definidos os títulos de uma tabela. Estes podem ser posicionados em qualquer célula. A diferença entre a marcação de célula e título de célula é que o título aparece em negrito.

5. ATRIBUTOS DA TAG `<TABLE>`

Vamos ver então o exemplo de uma tabela utilizando essas marcações básicas:

```
<table border>
<tr>
<td>Itens/Mês</td>
<th>Janeiro</th><th>Fevereiro</th><th>Março</th>
</tr>
<tr>
<th>Usuarios</th><td>80</td><td>93</td><td>120</td>
</tr>
<tr>
<th>Linhas</th><td>3</td><td>3</td><td>5</td>
</tr>
</table>
```

Uma tabela simples:

```
<table border=4>
<tr><th>coluna 1</th><th>coluna 2</th></tr>
<tr><td>linha1, coluna 1</td><td> linha 1, coluna 2</td></tr>
<tr><td>linha 2, coluna 1</td><td>linha 2, coluna 2</td></tr>
</table>
```

5.1. Aplicando bordas na tabela

O atributo **border** no elemento **table** serve para dar borda à tabela, sendo possível definir ainda, o valor da largura da borda da tabela. Sem o atributo **border**, o elemento **table** não ficaria muito distinto onde começa e termina cada uma das células:


```
<table border="1">
<tr>
<td>Primeira</td>
<td>Segunda</td>
</tr>
<tr>
<td>Terceira</td>
<td>Quarta</td>
</tr>
</table>
```

O exemplo abaixo mostra duas tabelas nas quais são especificados os valores 10 e 5 para a borda, e outra na qual nenhuma borda foi especificada. Você pode criar uma tabela sem bordas para montar um menu em que as opções não fiquem unicamente na vertical, como ocorre quando se usa o comando para a criação de listas.

```
<table border=10>
<tr>
<td>Linha 1 Coluna 1 </td>
<td>Linha 1 Coluna 2 </td>
<td>Linha 1 Coluna 3 </td>
</tr>
```

Linha 1 Coluna 1	Linha 1 Coluna 2	Linha 1 Coluna 3
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3
Linha 3 Coluna 1	Linha 3 Coluna 2	Linha 3 Coluna 3

```
<table border=5>
<tr>
<td>Linha 1 Coluna 1 </td>
<td>Linha 1 Coluna 2 </td>
<td>Linha 1 Coluna 3 </td>
</tr>
```

Linha 1 Coluna 1	Linha 1 Coluna 2	Linha 1 Coluna 3
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3
Linha 3 Coluna 1	Linha 3 Coluna 2	Linha 3 Coluna 3

```
<table>
<tr>
<td>Linha 1 Coluna 1 </td>
<td>Linha 1 Coluna 2 </td>
<td>Linha 1 Coluna 3 </td>
</tr>
```

Linha 1 Coluna 1 Linha 1 Coluna 2 Linha 1 Coluna 3
 Linha 2 Coluna 1 Linha 2 Coluna 2 Linha 2 Coluna 3
 Linha 3 Coluna 1 Linha 3 Coluna 2 Linha 3 Coluna 3

5.2. Largura de células e tabelas

Para escolher a largura de tabelas e células, utiliza-se o já conhecido atributo **width**, cujo argumento é idêntico para o caso de imagens:

```
<table border="1" width="200">
<tr>
<td width="30%">Primeira</td>
<td width="70%">Segunda</td>
```

```
</tr>
<tr>
<td>Terceira</td>
<td>Quarta</td>
</tr>
</table>
```

Primeira	Segunda
Terceira	Quarta

Note que quando o **width** é utilizado com o elemento **table**, a largura refere-se à largura da página, enquanto que quando usado dentro de um **td** a largura é referente ao tamanho da tabela. Isso tudo desde que o argumento seja dado em porcentagem e não em pixels.

5.3. Espaçamento entre células

Para controlarmos a distância entre duas células, utilizamos o atributo **cellspacing**, cujo argumento é dado em pixels:

```
<table cellspacing="20" border="1">
<tr>
<td>Primeira</td>
<td>Segunda</td>
</tr>
<tr>
<td>Terceira</td>
<td>Quarta</td>
</tr>
</table>
```

Primeira	Segunda
Terceira	Quarta

Para controlar o espaçamento vertical, utilizamos o **cellpadding**:

```
<table cell spacing="20" cellpadding="35" border="1">
<tr>
<td>Primeira</td>
<td>Segunda</td>
</tr>
<tr>
<td>Terceira</td>
<td>Quarta</td>
</tr>
</table>
```

Primeira	Segunda
Terceira	Quarta

5.4. Ampliando uma célula para mais de uma coluna

```

apostila - Bloco de Notas
Arquivo  Editar  Pesquisar  Ajuda

<table border>
<tr>
<td>Linha 1 Coluna 1 </td>
<td align=center colspan=2> Linha 1 Coluna 2-3 </td>
<td>Linha 1 Coluna 4 </td>
</tr>

```

O exemplo acima mostra o uso da opção Colspan para aumentar o tamanho de uma célula e fazer com que ela ocupe colunas adjacentes. Note que o conteúdo desta célula está centralizado pelo comando **align=center**. Lembre-se de que esse **não** é um recurso exclusivo do Internet Explorer.

Veja o resultado:

Linha 1 Coluna 1	Linha 1 Coluna 2-3		Linha 1 Coluna 4
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3	Linha 2 Coluna 4
Linha 3 Coluna 1	Linha 3 Coluna 2	Linha 3 Coluna 3	Linha 2 Coluna 4

5.5. Expandindo uma célula para mais de uma linha.

A opção Rowspan serve para aumentar o tamanho de uma célula e fazer com que ela ocupe colunas adjacentes. Note que o conteúdo desta célula está centralizado pelo comando **valign=top**.

Veja o efeito desses comandos:

```

<table border>
<tr>
<td>Linha 1 Coluna 1 </td>
<td>Linha 1 Coluna 2 </td>
<td>Linha 1 Coluna 3 </td>
<td>Linha 1 Coluna 4 </td>
</tr>

<tr>
<td>Linha 2 Coluna 1 </td>
<td valign=top rowspan=2>
  Linha 2 Coluna 2 </td>
<td>Linha 2 Coluna 3 </td>
<td>Linha 2 Coluna 4 </td>
</tr>

<tr>
<td>Linha 3 Coluna 1 </td>
<td>Linha 3 Coluna 3 </td>
<td>Linha 2 Coluna 4 </td>
</tr>

<tr>
<td>Linha 4 Coluna 1 </td>
<td>Linha 4 Coluna 2 </td>
<td>Linha 4 Coluna 3 </td>
<td>Linha 4 Coluna 4 </td>
</tr>
</table>

```

Resultado do Código:

Linha 1 Coluna 1	Linha 1 Coluna 2	Linha 1 Coluna 3	Linha 1 Coluna 4
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3	Linha 2 Coluna 4
Linha 3 Coluna 1		Linha 3 Coluna 3	Linha 2 Coluna 4
Linha 4 Coluna 1	Linha 4 Coluna 2	Linha 4 Coluna 3	Linha 4 Coluna 4

5.6. Títulos de Colunas

Com o par de comandos **<th>** (Title Heading), você pode especificar títulos para as colunas. Esses comandos funcionam da mesma forma que os **<td>**, só que deixam o conteúdo da célula centralizado e em negrito. Veja o próximo exemplo.

Resultado do Código:

```
<table border>
<tr>
<th>Título1</th> <th>Título2</th>
| <th>Título3</th>
</tr>
<tr>
<td>Linha 1 Coluna 1 </td>
<td>Linha 1 Coluna 2 </td>
<td>Linha 1 Coluna 3 </td>
</tr>

<tr>
<td>Linha 2 Coluna 1 </td>
<td>Linha 2 Coluna 2 </td>
<td>Linha 2 Coluna 3 </td>
</tr>

<tr>
<td>Linha 3 Coluna 1 </td>
<td>Linha 3 Coluna 2 </td>
<td>Linha 3 Coluna 3 </td>
</tr>
</table>
```

Título1	Título2	Título3
Linha 1 Coluna 1	Linha 1 Coluna 2	Linha 1 Coluna 3
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3
Linha 3 Coluna 1	Linha 3 Coluna 2	Linha 3 Coluna 3

Formatação de Tabelas

<TABLE ***>

- BORDER="tamanho das margens exteriores"
- BORDER COLOR="[número hexadecimal da cor](#) das margens"
- CELLPADDING="espaço entre as margens e o conteúdo das células"
- CELLSPACING=tamanho das margens interiores"
- WIDTH="largura da tabela em pixels ou em % da tela"
- ALIGN="alinhamento horizontal do conteúdo da tabela (left/center/right)"
- VALIGN="alinhamento vertical do conteúdo da tabela (top/center/bottom)"
- BGCOLOR="[número hexadecimal da cor](#) do fundo da tabela"

<TR ***>

- ALIGN="alinhamento horizontal do conteúdo da linha (left/center/right)"
- VALIGN="alinhamento vertical do conteúdo da linha (top/center/bottom)"
- BGCOLOR="[número hexadecimal da cor](#) do fundo da linha"

<TD ***>

- WIDTH="largura da célula em pixels ou em % da tabela"
- ROWSPAN="altura da célula em linhas"
- COLSPAN="largura da célula em colunas"
- ALIGN="alinhamento horizontal do conteúdo da célula (left/center/right)"
- VALIGN="alinhamento vertical do conteúdo da célula (top/center/bottom)"
- BGCOLOR="o [número hexadecimal da cor](#) que quer usar para o fundo da célula"

6. Verificação de Aprendizagem (Ver o Caderno de Exercícios)

5ª AULA

🖨 Assuntos a serem Abordados:

- 🖨 Formulários – tag `<form>`
- 🖨 `<frames>` / `<frameset>`
- 🖨 Endereços
- 🖨 Verificação de Aprendizagem (Ver o Caderno de Exercícios)

1. FORMULÁRIOS – TAG `<FORM>`

Os formulários servem para recolher dados introduzidos pelos visitantes e enviá-los para você, através de seu servidor. Um formulário é uma seção da página HTML que contém elementos que permitem ao visitante inserir dados (elementos `<textarea>` e vários tipos de elementos `<input>`, `<option>` e `<select>`).

Estes elementos permitem inserir dados numéricos, textos pequenos, textos longos, selecionar elementos em uma lista com várias escolhas, responder facilmente com respostas do tipo "sim" ou "não", selecionar rapidamente uma opção em um pequeno grupo, etc.

1.2. Criar Formulários

Os formulários são criados com o elemento `<form>`. Dentro desse elemento principal colocamos diversos elementos para a inserção dos dados.

```
<form>
<input>
.
.
.
<input>
</form>
```

O elemento `<form>` por si só não faz com que o navegador desenhe nada na página nem permite inserir dados. Ele contém elementos que recolhem os dados e possui atributos que "dizem" ao navegador como e para onde devem ser enviados os dados inseridos pelo visitante.

Veremos a sintaxe desta tag:

```
< form name="form1" method="post" action="">
```

- ✓ **name** – é o nome que será dado ao formulário, por padrão é definido "form1", mas o usuário pode colocar outro nome.

- ✓ **method** – é o método que será utilizado para que os dados do formulário sejam enviados para o cliente da página. Existem dois métodos para este envio, o “**post**” e o “**get**”. Veremos melhor a sua utilização no curso de **ASP**.
- ✓ **action** – é o arquivo que será executado pelo formulário para que os dados cheguem ao cliente com segurança.

Após colocarmos a **<form>**, teremos que criar as caixas de texto onde serão digitadas as informações. Para isso utilizaremos a tag **<input>** veja a sintaxe:

<input type="text" name="textfield">

onde:

- ✓ **type** – é o tipo que será a caixa de texto, normalmente o tipo é “**text**”, mas há outros tipos “**password**” que é utilizado em caixas que guardam informações sobre senhas.
- ✓ **name** – é o nome que será dado à caixa de texto que guardará as informações.

1.3. Action e Botões (Reset e Submit)

Quando o visitante clica sobre o botão “**Submeter**” (ou “**Submit**”), as opções marcadas e o texto que foram inseridos no formulário são enviados para você (ou para seu servidor). Conheça dois dos principais botões e suas funções.

- ✓ **submit** - usado para enviar os dados do formulário.
- ✓ **reset** - utilizado para limpar os dados do mesmo.

O atributo **action** do elemento **<form>** contém o endereço (**URL**) do recurso da Web (site, download, email, etc.) que está encarregado de realizar este processamento. É para lá que o conteúdo do formulário será enviado.

```
<form name="input" action="exemplos /action.html " method="get ">
Nome de usuário:
<input type="text" name="usuario">
<input type="submit" value="submeter">
</form>
```

O exemplo acima ficará sendo exibido da seguinte forma em seu navegador:

Nome de usuário:

1.4. Input

O elemento que encontramos com maior frequência em formulários é o elemento **<input>**.

O exemplo abaixo mostra um formulário simples com dois elementos **input**:

```
<html>
<head>
<title>Cadastro Pessoal</title>
</head>
<body>
<form name="cadastro" method="post" action=" ">
<b><i>Cadastro Pessoal </i></b>
<br>
Nome:
<input type="text" name="Nome">
<br>
Telefone:
<input type="text" name="Nome">
<br>
<br>
    <input type="submit" name="Enviar" value="Enviar">
    <input type="reset" name="Limpar" value="Limpar">
</form>
</body>
</html>
```

ATENÇÃO: É um input para cada caixa de texto.

O formulário acima ficará sendo exibido desta forma em seu navegador:

Cadastro Pessoal

Nome:

Telefone:

O elemento **<input>** pode assumir diversas formas com finalidades diferentes.

1.5. Radio Buttons

Os **"Radio Buttons"** são utilizados para se criar um grupo pequeno de opções em que apenas podemos selecionar uma de cada vez.

```
<form action="processamento.asp">
<input type="radio" name="sexo" value="masculino"> Masculino
<br>
<input type="radio" name="sexo" value="feminino"> Feminino
</form>
```


O exemplo acima ficará sendo exibido desta maneira em seu browser:

- ☒ Masculino
☐ Feminino

Repare que só podemos selecionar uma das opções acima.

1.6. Checkboxes

As caixas de validação "**checkboxes**" devem ser usadas sempre que queremos que o visitante aceite (ou não) os itens dentro de um pequeno grupo. É permitido validar mais de uma opção simultaneamente.

```
<form>  
<input type="checkbox" name="carro">Eu tenho um carro  
<br>  
<input type="checkbox" name="surf">Eu tenho uma prancha de surf  
</form>
```

O exemplo acima ficará sendo exibido desta forma em seu browser:

- ☒ Eu tenho um carro
☒ Eu tenho uma prancha de surf

Repare que podemos selecionar várias opções ao mesmo tempo.

2. <FRAMES> / <FRAMESET>

Os **frames** (**molduras**) são subjanelas definidas sobre a janela principal do browser. Estas subjanelas são criadas dividindo a janela em várias partes, cada parte é chamada de **frameset**. Os **frames** são geralmente usados para que um **frameset** permaneça enquanto outros mudem baseados nos links que o usuário seleciona.

2.1. <frameset></frameset>

Determina o início e o fim de um frame. Este elemento divide a janela do browser em linhas e colunas. Deve ser usado juntamente com os parâmetros **cols** (frame na vertical) ou **rows** (frame na horizontal).

2.2. Atributos Frame

src="página.htm"

Determina a página ou arquivo que será carregado na região correspondente.

name="nome"

Determina um nome ao frameset correspondente. O nome é de vital importância na utilização dos frames.

noresize

Deixa a linha do frameset estática.

marginheight="x"

Determina a largura em pixels da margem vertical.

marginwidth="x"

Determina a largura em pixels da margem horizontal.

scrolling="valor"

Determina se aparecerá barra de rolagem. Os valores podem ser yes (com barra de rolagem), no (sem barra de rolagem) ou auto (barra de rolagem só aparecerá quando o conteúdo da página ultrapassar o tamanho do frameset).

cols="x,x"

Determina o número de regiões verticais da página. Deve ser indicado juntamente com uma lista de valores. Cada valor especifica a largura de cada frameset e pode ser definido em pixels, percentual ou através de asterisco.

rows="x,x"

Determina o número de regiões horizontais da página. Deve ser indicado juntamente com uma lista de valores. Cada valor especifica a largura de cada frameset e pode ser definido em pixels, percentual ou através de asterisco.

border="x"

Determina a espessura da linha divisória que aparece entre os framesets. Sendo que x é o valor da borda em pixels.

bordercolor="cor"

Determina a cor da linha divisória que aparece entre os framesets.

Importante!

Para abrir uma página em **frameset** diferentes você deve especificar o nome do mesmo no link.

Exemplo: ``

Logo abaixo está o código básico de uma página com frame.

Este exemplo mostra uma página HTML com Frames (**molduras**).

1ª Página: pagina_banner.htm

2ª Página: pagina_menu.htm

3ª Página: pagina_princ.htm

Criar a Página Index.htm, que será a página dos Frames, onde será visualizado as páginas anteriormente criadas.

1ª Página: pagina_banner.htm

`<html>`

`<body>`

Página Banner
`
`

Oi Tudo bem !!!
`<p>`

`</body>`

`</html>`

2ª Página: pagina_menu.htm

`<html>`

```
<body>  
    Página Menu<br>
```

```
Oi Tudo OK !!!<p>
```

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

3ª Página: pagina_princ.htm

```
<html>  
<body>  
    Página Principal<br>
```

```
Oi Tudo Legal !!!<p>
```

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

Pode ser acrescentados cores, imagens e efeitos em todas as páginas! Use sua criatividade.

Página Index.htm

```
<html>  
<head>  
<title> Home Page </title>  
</head>  
<frameset rows="25%,25%,25%" bordercolor="yellow">  
<frame src="pagina_banner.htm" name="banner">  
<frame src="pagina_menu.htm" name="menu">  
<frame src="pagina_princ.htm" name="conteudo">  
</frameset>  
</html>
```

Esta página que foi criada, deve ser chamada de index.htm.

3. ENDEREÇOS

Esta tag é utilizada para identificar autorias de sua página. Geralmente são apresentadas na parte inferior de cada página. Mostra quem confeccionou aquela página, com quem o leitor deve entrar em contato caso queira ter alguma informação ou fazer alguma observação. Estes endereços são precedidos de um fio <hr>, que ainda veremos, e você poderá utilizar a tag
 para separar linhas.

Sintaxe: **<ADDRESS> TEXTO</ADDRESS>**

Esta é uma excelente forma de garantir que as pessoas entrem em contato com você caso necessitem e também uma forma de estar melhorando a sua HP (Home Page) através de sugestões.

6ª AULA

📁 Assuntos a serem Abordados:

- 📁 Cascading Style Sheets
- 📁 Porque o CSS
- 📁 Vantagens dos Estilo CSS
- 📁 Trabalhando com css.
- 📁 Adicionando fontes
- 📁 Trabalhando com o painel Design na aba Estilos CSS
- 📁 Formatando o plano de fundo da pagina.

Cascading Style Sheets(css).

Cascading Style Sheets (ou simplesmente **CSS**) é uma [linguagem de estilo](#) utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como [HTML](#) ou [XML](#). Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

Ao invés de colocar a formatação dentro do documento, o desenvolvedor cria um [link](#) (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um [portal](#). Quando quiser alterar a aparência do [portal](#) basta portanto modificar apenas um arquivo.

Porque o CSS

Reduz o tempo de trabalho e manutenção das paginas web sendo possível uma melhor administração das tags HTML, acelerando as possíveis alterações futuras, que em um site grande geraria(muito trabalho). No CSS é possível editar de forma mais abrangentes as estruturas de tags HTML, reduzindo a limitação da linguagem, personalizando mais ainda as paginas. Podendo controlar FONTES, CORES, IMAGEM DE FUNDO, LINHAS, ALTURA, LARGURA, MARGENS. Resumindo é um HTML advanced onde você tem mais liberdade para trabalhar

Vantagens dos Estilo CSS

Antes de mais nada, deve ser dito que quase tudo o que se pode fazer com style sheets pode ser feito também com os descritores de HTML convencionais (existem apenas umas poucas exceções, e uma enorme quantidade de coisas que só podem ser feitas com tediosas construções de HTML). Entretanto, há vantagens adicionais no uso do css.

1. Separação Lógica da Estruturação e da Formatação das Informações

Com o tempo o HTML evoluiu para uma combinação complexa e confusa de estruturar e formatar os descritores, o que torna a manutenção de um web site grande e complicado uma tarefa árdua e

difícil (se você usar a tag FONT, por exemplo, e quiser mudar uma fonte ou a sua cor num certo número de páginas, vai gastar horas para rastrear todas as tags). Os estilos CSS vão lhe permitir isolar os códigos de formatação aplicado a várias páginas, de maneira que as mudanças gerais de estilo podem ser feitas editando apenas um único arquivo.

2. Redução do Tamanho e de Tempo de Carregamento das Páginas

Quando um usuário quer acessar uma página web, o browser tem que analisar todo o código HTML da página para representá-la na tela. Se você colocar uma formatação complexa nas suas páginas, isso acrescenta um monte de tempo de análise para cada página. Ao colocar as formatações numa única folha de estilo referenciada por cada página, você reduz a quantidade de código das páginas e conseqüentemente isto reduz também a quantidade de dados que têm que ser transmitidos e analisados pelos browsers. O resultado são páginas que aparecem mais rapidamente e usuários mais felizes.

Trabalhando com CSS.

Uma maneira bem fácil de acrescentar estilização à página é adicionar algumas cores a ela. Começamos com uma folha de estilos incorporada ao arquivo HTML. Mais a frente, separaremos os arquivos HTML e CSS. Separar os arquivos é uma boa técnica, pois isto permite usar a mesma folha de estilos para vários arquivos HTML: você precisará escrever uma só folha de estilos. Por ora vamos colocar nossa CSS no mesmo arquivo HTML.

Vamos precisar inicialmente do elemento <style> no nosso arquivo HTML. A folha de estilos será colocada dentro deste elemento acrescente mais cinco linhas de código dentro da tag "head" do seu arquivo HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Minha primeira página CSS</title>

  <style type="text/css">  → Delimitador
    body {                → tag que recebera os atributos
      color: purple;      → Altera cor da fonte
      background-color: #d8da3d } → Plano de fundo da pagina.
    </style>              → Delimitador

</head>

<body>
```

A primeira linha informa que trata-se de uma folha de estilos escrita para CSS ("text/css"). A segunda linha diz que as regras de estilo a seguir, aplicam-se ao elemento "body". A terceira determina que os textos serão em cor purple (púrpura) e a linha a seguir coloca a cor #d8da3d (amarela esverdeada) para o fundo da página.

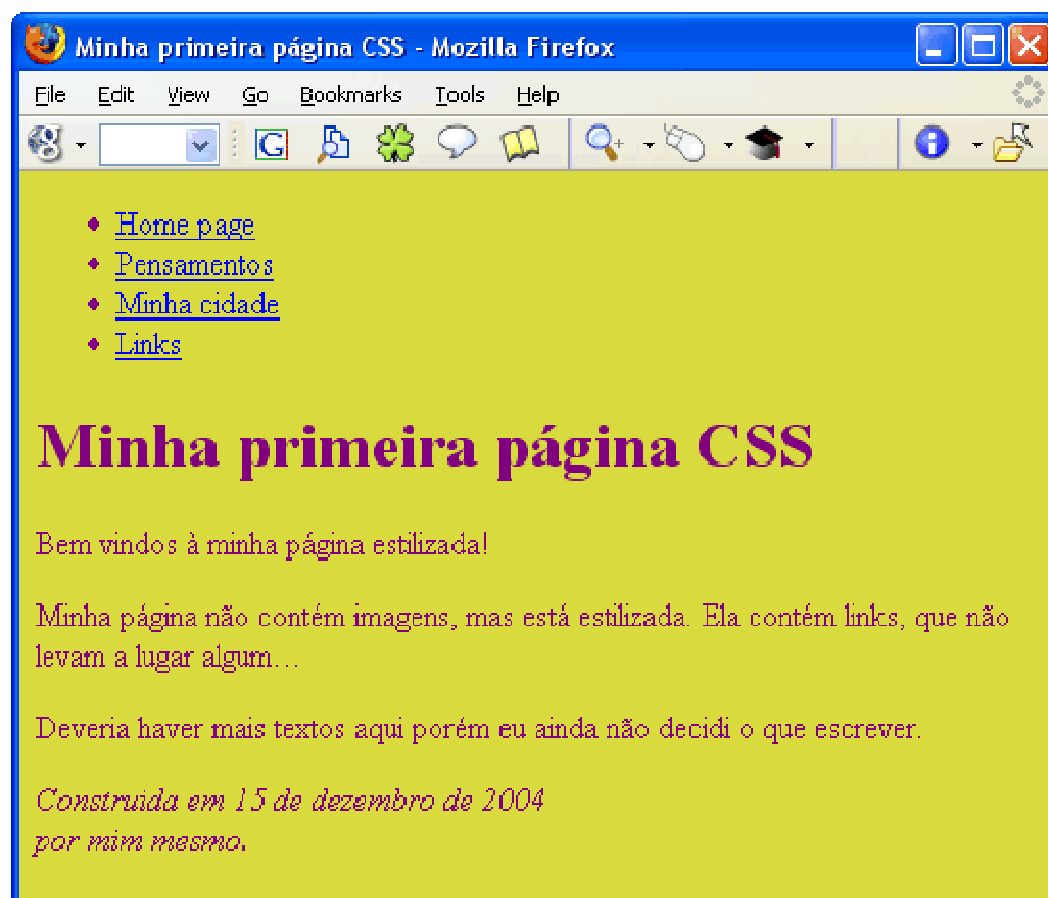
Em CSS as folhas de estilos são constituídas de *regras*. Cada regra compõe-se de três partes:

1. um *selector* (no exemplo: “body”), que informa ao browser qual é a parte do documento a qual se aplica a regra;
2. uma *propriedade* (no exemplo, 'color' e 'background-color', duas propriedades distintas), que especifica qual é o aspecto do layout que está sendo estilizado;
3. e um *valor* ('purple' e '#d8da3d'), que fornece o valor da propriedade.

No exemplo nota-se que as regras podem ser combinadas. Ali estão escritas duas propriedades. Poderíamos ter escrito duas regras separadamente sem combiná-las:

```
body { color: purple }  
body { background-color: #d8da3d }
```

A cor de fundo (background-color) do elemento body valerá para todo o documento. Nós não escrevemos nenhuma regra adicional colocando fundo de outra cor para os outros elementos da página (p, li, address...) então por 'default' eles não terão cor de fundo (ou seja: serão transparentes). A propriedade 'color' determina a cor dos textos para o elemento body e todos os elementos contidos em body herdarão esta cor, a menos que uma outra regra CSS determine uma cor diferente para qualquer elemento na página.

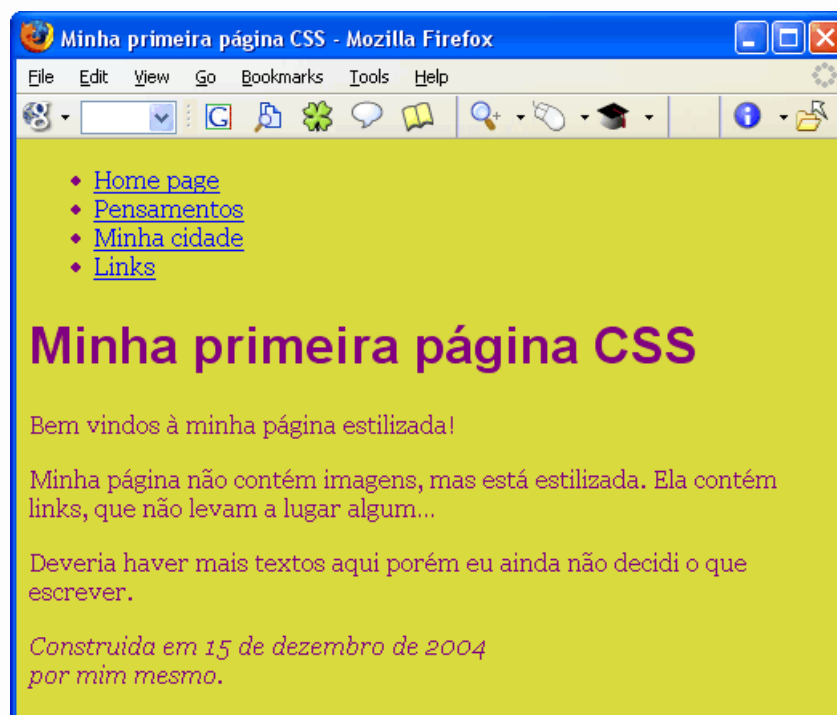


Adicionando fontes

Outra coisa bem fácil de se estilizar são as fontes para os elementos da página. Vamos escolher fonte “Georgia” para os textos, exceto para o cabeçalho h1 ao qual atribuiremos a fonte “Helvetica.”

Você nunca terá como saber quais são as fontes que os visitantes do seu site têm instaladas em suas máquinas, por isso é de boa técnica sempre especificarmos fontes alternativas nas folhas de estilos: se a fonte Georgia não estiver disponível as fontes Times New Roman ou Times irá substituí-la sem grandes prejuízos e ainda mais, se ambas também estiverem indisponíveis o browser do visitante usará uma fonte (with serif) com serifas. Se a fonte Helvetica não estiver instalada, as fontes Geneva, Arial e SunSans-Regular são similares e se nenhuma delas estiver disponível o browser usará uma fonte sem serifas.

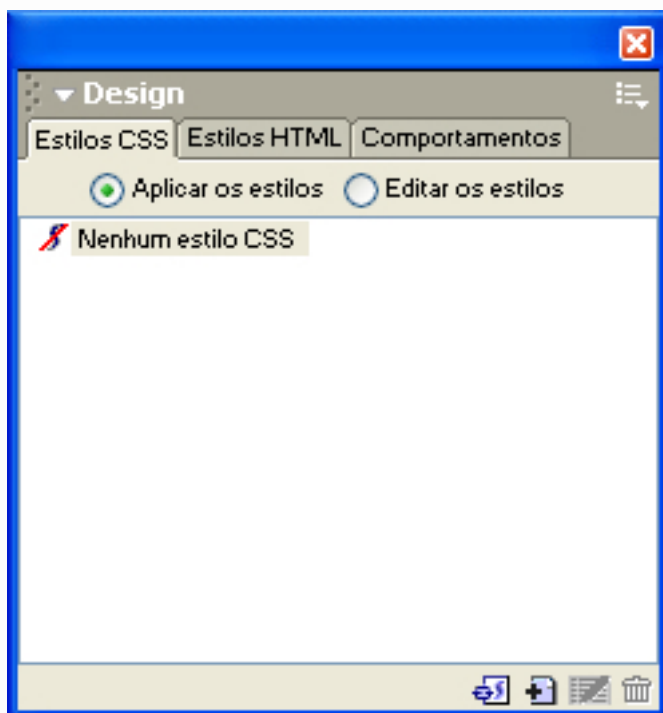
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>Minha primeira página CSS</title>
  <style type="text/css">
    body {
      font-family: Georgia, Times New Roman, Times, serif; → fonte
      color: purple; → cor da fonte
      background-color: #d8da3d }
    h1 { → Alterando a fonte no elemento
      font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }
  </style>
</head>
<body>
```





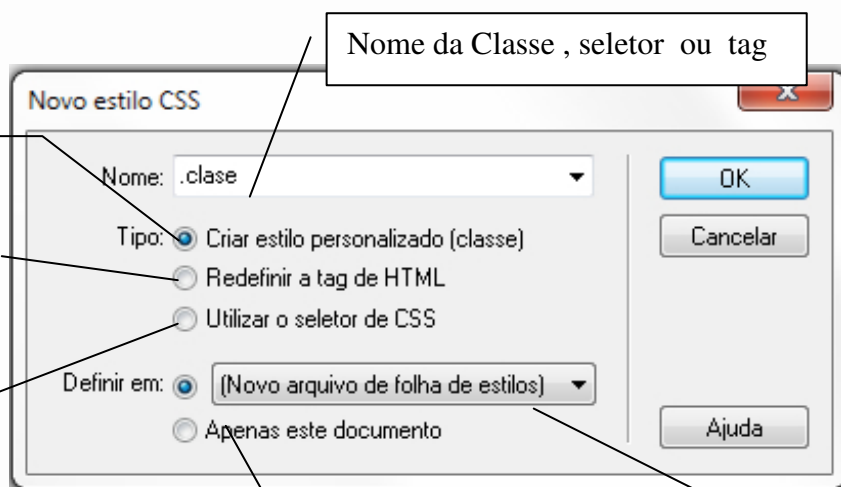
Agora cabeçalho(h1) e texto estão com fontes diferentes.

Trabalhando com o painel Design na aba Estilos CSS

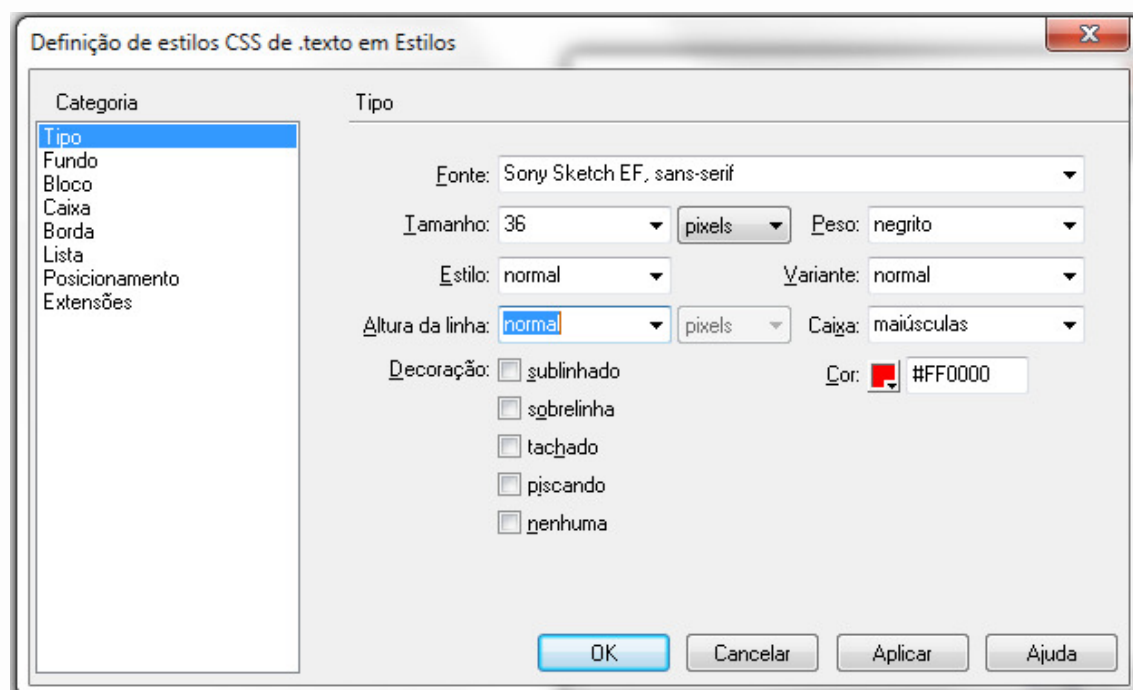
Este será o painel que usaremos para construir os estilos CSS de uma maneira mais dinâmica sem a utilização de códigos .



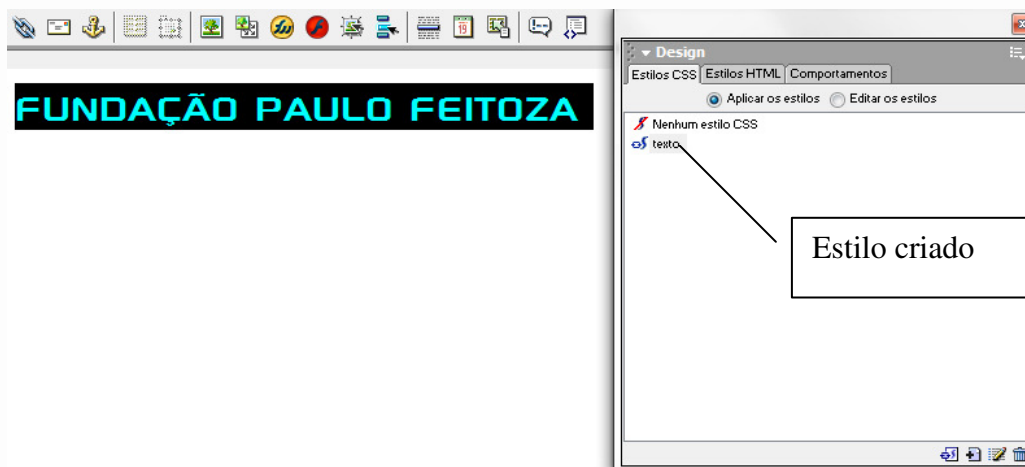
A aba estilo CSS é formada pelos botões anexar folha de estilos  , nova folha de estilo  . Ao clicar no botão novo estilo abrirá a seguinte janela.



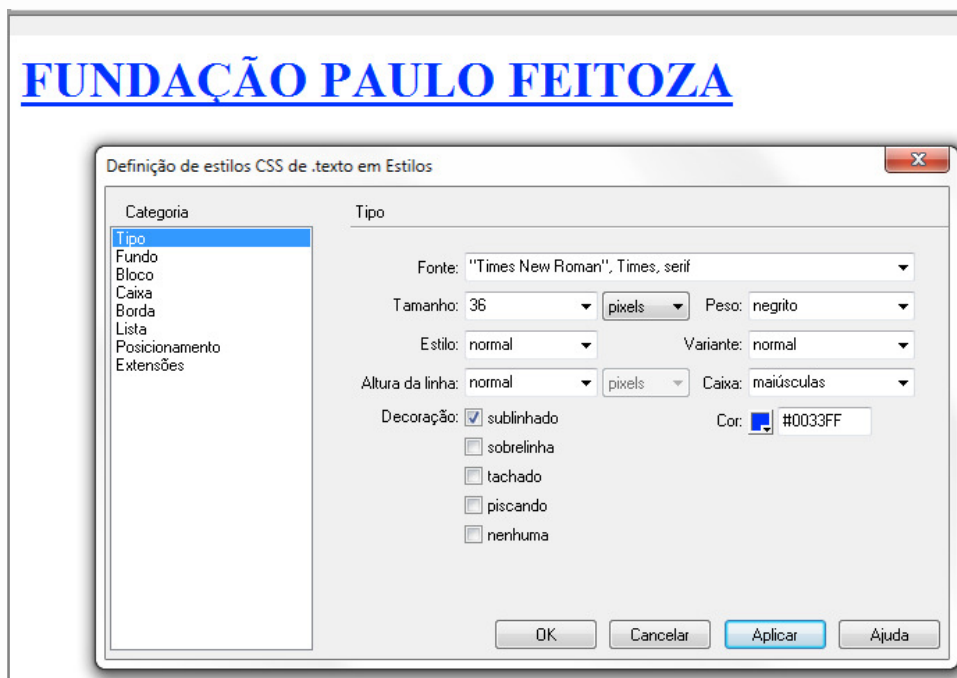
Depois de configurar seu novo estilo pressione o botão OK para salvar o estilo na pasta raiz de seu projeto. Após salvar o arquivo abrir uma janela para configurar seu primeiro estilo css.



Formate a classe a classe como desejado depois pressione OK. Para aplicar o estilo no objeto selecione-o e clique no nome do estilo criado.



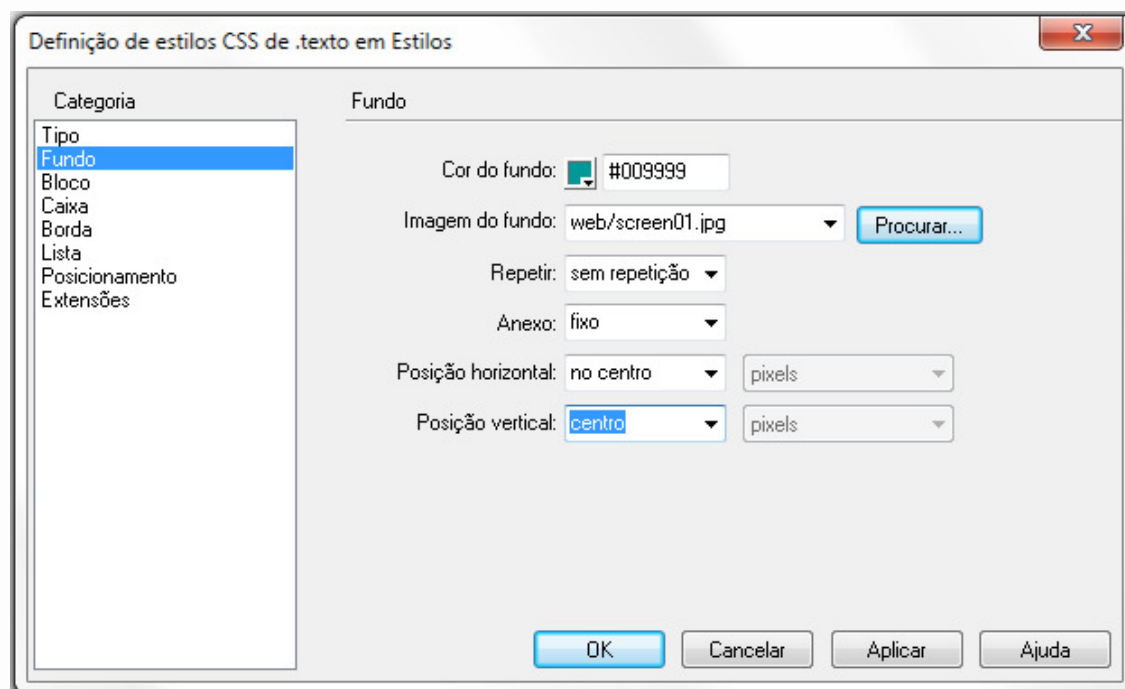
Ao clicarmos no estilo o texto absorveu as características predefinidas na janela do estilo. Podemos alterar as formatações predefinidas no estilo clicando com o botão direito ou dando um clique duplo em cima do estilo, realize as alterações desejadas depois OK.



Perceba que ao clicar em OK, automaticamente tudo que tinha a formatação antiga foi alterada, agilizando assim o processo de formatação do site.

Formatando o plano de fundo da pagina.

Através do css podemos também editar o plano de fundo do site , podendo assim ter um maior controle sobre plano de fundo. Crie um novo estilo, clique na categoria fundo , formate como e OK.



Criando Lista personalizada.

Com o css podemos também redefinir os marcadores das lista ordenadas e não ordenadas .Antes de criar um estilo abra o Fireworks e crie o seu marcador (altura : 20 e largura: 20).

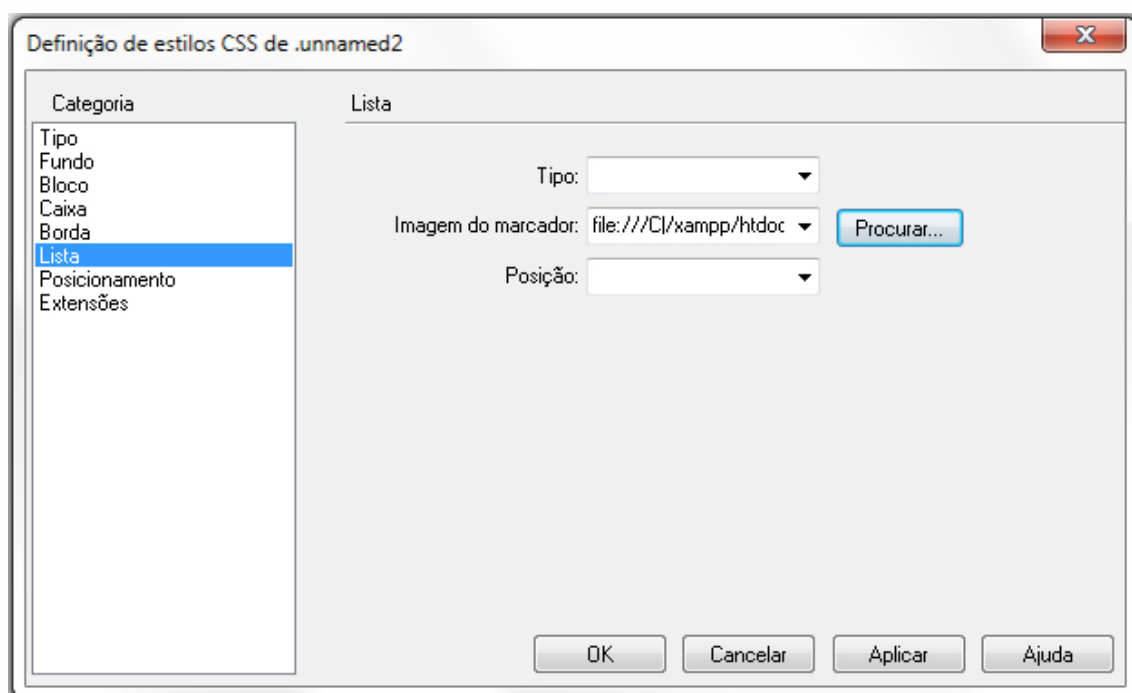


ORIGINAL



MARCADOR

Com o marcador já pronto volte para o Dreamweaver , crie uma novo estilo ,OK, clique na categoria lista escolha a imagem que deseja utilizar como marcador(" A imagem criada no fireworks") e OK.



Diferente dos outro estilos já vistos para adicionar o estilo ao texto deve-se primeiro aplicar os marcadores tradicionais que se localizam na barra de propriedades , e so depois aplicar o estilo CSS.



FUNDAÇÃO
DES. PAULO FEITOZA

APOSTILA HTML



5555555555555555

5555

5