

Prof. Lucas Rafael Filipak

HTML

Por Exemplos



Copyright © 2016 by **Lucas Rafael Filipak**

Todos os direitos reservados. Vedada a produção, distribuição, comercialização ou cessão sem autorização do autor. Os direitos desta obra não foram cedidos.

Impresso no Brasil
Printed in Brazil

Diagramação
Rayanne Lima

Capa
Andreza Souza

Revisão
Do Autor

Dados Internacionais de Catalogação na Publicação (CIP)
Ficha Catalográfica

T278

Título do livro/Autor;
Olinda: Livro Rápido, janeiro de 2016.

210 p.

Contém bibliografia ao final de cada capítulo
ISBN 978-85-406-0669-2

1. Responsabilidade tributária. 2. Confisco tributário. 3. Relação de trabalho e relação de consumo. 4. Licitação – Carta convite. I. Moita, Paulo Rochelle Andrade. II. Valotto, Carlos Antônio. III. Moita, Hilda Pereira Madeira. IV. Magalhães, Rafael Geovani da Silva. V. Albuquerque, Yves Porfírio Castro de. VI. Título.

34 CDU (1999)

Fabiana Belo - CRB-4/1463

Livro Rápido Editora – Elógica
Coordenadora editorial: *Maria Oliveira*
Rua Dr. João Tavares de Moura, 57/99 Peixinhos
Olinda – PE CEP: 53230-290
Fone: (81) 2121.5307/ (81) 2121.5313
livrorapido@webelgica.com
www.livrorapido.com

AGRADECIMENTOS

Agradecer, é algo que sempre fazemos ao sermos presenteados, neste momento, quero expressar, em palavras, o presente a mim dado durante anos. Agradeço, então, à Deus por todas as concretizações maravilhosas que aconteceram em minha vida; ao meu pai, exemplo, mestre e maior apostador no meu sucesso; à minha mãe, por sempre estar me motivando e incentivando; ao meu avô e à minha avó, por formarem a pessoa que hoje escreve, a criação e dedicação que serão sempre passivos do mesmo carinho e, sem esquecer-me, mesmo sem nomear, para não cair em deselegância, às pessoas que destinaram parte do seu tempo para analisar este livro.

SUMÁRIO

INTRODUÇÃO	9
Internet.....	9
Como funciona a Internet.....	9
Navegadores	11
HTML	12
Site ou Home Page?	12
Como compro um Domínio	13
Hospedagem.....	13
PRIMEIROS PASSOS	15
Fatores na escolha de um bom editor	16
Símbolos	20
Comentário	21
FORMATAÇÃO DE TEXTOS.....	23
Estilos pré-definidos.....	25
Outras TAGs – Formatação de texto.....	25
Texto Pré-Formatado	26
LISTAS	29
Listas não ordenadas	30
Listas ordenadas	31
Parágrafos	33
Alinhamentos	34
Traço ou Linha	35
TABELA.....	39
Criando Tabelas.....	40
Mesclando células	42
ALINHAMENTOS EM TABELAS.....	47
Alinhamentos Verticais	49

IMAGENS.....	55
Imagens como plano de Fundo.....	58
Qual o tipo de imagem usar.....	59
Ícones na aba do navegador – favicon	61
LINKS.....	63
Para que servem os links.....	64
Link para navegação na mesma página	66
Formatação de links.....	67
Imagem como link.....	67
Projeto HTML 1: Turismo em Curitiba.....	70
VÍDEO E ÁUDIO	71
Vídeo	72
Áudio	73
Som no fundo da página	74
FORMULÁRIOS	77
Formulários	79
Campo Texto	79
Campo Senha	80
Botão Radio (Radio Button)	81
Botão de Checagem (Checkbox)	82
Fieldset	83
Botão Submeter (Submit)	83
Botão Image	84
Botão Reset.....	85
Botão	85
Select.....	87
Área de Texto	89
Contadores de caracteres.....	90
Projeto HTML 2: Música.....	93
CSS	95
O que é CSS	95
Comentário.....	99

CSS Interno	100
CSS Externo	101
Agrupar seletores	102
Formatação de Textos	104
Espaçamento entre linhas	106
Formatação de Plano de Fundo	107
Formatação de Tabela	109
Espaço entre as bordas	111
Alinhamento Vertical	113
Arredondando cantos da tabela	114
Pseudo-Classe	117
Formatação de Link	118
Formatando Imagens	119
Posicionando Imagens (e outros elementos)	121
Formatando Formulários	125
Projeto CSS 1: Formulário de Login	129
Criando Layout com DIVs	129
Tipos de Posicionamentos	131
Relative X Absolute	133
Ordenação das DIVs	135
Exemplificando DIVs – relative e absolute	136
Position Fixed	139
Projeto CSS 2: Layout de Formulário	142

INTRODUÇÃO

Internet

As guerras não trazem apenas tragédias e desgraças. Na Guerra Fria, os Estados Unidos precisavam desenvolver um meio de comunicação eficaz. Surgiu então o Arphanet, que foi utilizado para interligar as bases militares americanas. Terminado a guerra, os militares acharam que seria inútil manter essa tecnologia somente em seu poder. Os cientistas foram os primeiros a serem contemplados com a nova tecnologia. Mais tarde as universidades americanas também tiveram acesso. Somente se tornou popular quando foi liberada para os pesquisadores domésticos. Aos poucos a internet foi ganhando espaço e nos dias de hoje é considerada o maior sistema de comunicação desenvolvida pelo homem.

Como funciona a Internet

A internet é uma enorme teia, composta por muitas redes de computadores. Tudo o que trafega pela internet é dividido em pacotes de dados. Os pacotes são enviados e ao chegarem ao seu destino são remontados. Caso algum pacote não chegue ao seu destino, o mesmo é reenviado. O protocolo que garante a comunicação entre os computadores é o TCP (do inglês Transmission Control Protocol). Na internet todos

os computadores são identificados por um número único, chamado de IP (Internet Protocol). O servidor responsável em distribuir os IPs para os computadores do mundo todo chama-se DHCP (do inglês Dynamic Host Configuration Protocol). Os IPs podem ser fixos ou dinâmicos. Cada vez que o usuário conecta a internet, o computador recebe um IP para que ele possa navegar na internet. Quando o computador é desconectado ele perde o IP, caracterizando assim o IP dinâmico. O IP fixo é um serviço contratado no provedor de acesso. Esse serviço garante que se o modem precisar ser reiniciado ou desligado, ao se reconectar na internet ele ficará com o mesmo IP.

Como todo site está hospedado em um provedor (o provedor possui um IP fixo), se for digitado o IP na barra de endereço o site correspondente aquele IP irá abrir no navegador. Para evitar que ficamos decorando os IPs dos nossos sites favoritos, existe um serviço conhecido como DNS (Domain Name Service) que traduz os IPs para os nomes de domínios. Exemplo:

Se na barra de endereço for digitado o IP 200.147.3.142 é o mesmo que digitarmos www.uol.com.br.

O FTP (File Transfer Protocol) é o protocolo responsável pelas transferências de dados entre computadores de uma rede TCP/IP. É utilizado na internet nos downloads (transfere um arquivo que se encontra em um servidor WEB para o seu computador) e nos uploads (envia um arquivo que se encontra em seu computador para um servidor WEB).

Os provedores são empresas que oferecem serviços de acesso à internet, tanto pra as residências quanto para as empresas.

Navegadores

No começo, a internet não passava de textos compartilhados e atraía apenas pessoas interessadas em estudá-la ou geeks (pessoas obcecadas por tecnologia, eletrônica, jogos eletrônicos ou de tabuleiro e outros). Um grupo de estudantes fundou a Netscape Corporation e começaram a trabalhar em um projeto que revolucionaria tudo o que já tinha sido visto. A Netscape criou o primeiro navegador, o Netscape Navigator que permitia a exibição de imagens e sons. A Microsoft, uma das maiores empresas de tecnologias, também estava trabalhando em um navegador, o Internet Explorer. Em meados de 1994 o Navigator era utilizado por 86% dos internautas. A Microsoft tinha uma vantagem, pois ela fornecia o Windows (sistema operacional). A cartada da Microsoft foi incluir o Internet Explorer no Windows, pois a grande maioria dos usuários de computadores utilizavam o Windows. Essa atitude fez com que em 1999 apenas 4% dos usuários utilizassem o Navigator. Bill Gates, o fundador e diretor da Microsoft respondeu um processo anti-trust (monopólio), foi condenado e quase viu a sua empresa ser dividida.

HTML

HTML (HyperText Markup Language - Linguagem de Marcação de Hipertexto) é considerada a linguagem base para todas as outras linguagens utilizadas para desenvolvimento WEB. Foi desenvolvida em 1989 por Tim Berners Lee. Ele criou uma linguagem baseada em marcações (etiquetas ou tags) que se popularizou quando o navegador Mosaic surgiu. Além de texto, ela pode exibir imagens, vídeos, animações e sons.

Para serem exibidas, as páginas HTMLs necessitam de um Browser (ou navegador). O protocolo responsável em transmitir e/ou exibir as páginas em HTML é o HTTP (protocolo de transferência de Hipertexto).

Quando começou a ser expandido o HTML, as empresas começaram uma guerra. Eram rivais e criavam códigos HTMLs que só funcionavam apenas em seus próprios navegadores. A W3C (Word Wide Web Consortium) é a empresa que padronizou as tags do HTML garante que a maioria das programações nos documentos HTMLs sejam visualizadas em diferentes navegadores.

Site ou Home Page?

O site é um conjunto de páginas que representa uma pessoa ou uma empresa na internet. Alguns autores que não gostam de utilizar palavras ou termos em inglês adotam o termo sitio (essa prática é muito comum em

Portugal). Todo site tem uma página principal, conhecida como home page.

Como compro um Domínio

Antes de comprarmos um domínio, precisamos saber o que é um domínio! O domínio é o nome digitado na barra de endereço para acessar o site. Como exemplo pegamos o site www.google.com.br onde o domínio é google.com.br.

O site oficial de venda de domínios brasileiros (os .com.br) é o registro.br. O primeiro passo após acessar o site é pesquisar se o domínio desejado encontra-se disponível para compra. A pesquisa pode ser feita por qualquer usuário da internet, mas para efetuar a compra é preciso fazer o cadastro no site do registro.br.

Na hora da compra, você vai escolher quanto tempo quer ficar com o domínio, pode ser 1 ano ou no máximo 10 anos. O valor anual do domínio é R\$ 30,00 reais. Quando o prazo for expirado, você pode renovar o domínio ou não. Para se adquirir um domínio primeiramente tem que ser contratada a hospedagem.

Hospedagem

Diferente do domínio que tem um site oficial para sua compra, as hospedagens são oferecidas por diversas empresas. O valor da hospedagem depende de vários

fatores, tais como: espaço, taxa de transferência, suporte a linguagens de programação e banco de dados.

Agora que já sabemos o que é e para que serve o HTML vamos colocar a mão na massa! Todo documento HTML é criado em um editor de texto. Como exemplo de editor temos o bloco de notas (que já vem instalado com o sistema operacional) e o NotePad++.

Atividades

1) Pesquisa:

Tema 1: Como surgiu a internet?

Tema 2: Como os dados trafegam pela internet?

Tema 3: Principais navegadores da atualidade e suas características.

PRIMEIROS PASSOS

Fatores na escolha de um bom editor

Existem vários softwares que podem ser utilizados para a programação de códigos HTML. Dois fatores são muito importantes na escolha: o recurso de auto-completar e o de tags coloridas.

Um bom editor para estudantes é o Notepad++. Além de ser grátis ele pode ser utilizado na versão portable, não precisando a sua instalação, podendo rodar diretamente do pendrive.

Uma ressalva do Notepad++ é que ele é destinado para pessoas que gostam de desenvolver seus trabalhos utilizando somente o código em si.

A linguagem HTML é desenvolvida através de tags pré-definidas onde cada tag tem a sua função. Para criar uma página de internet é preciso, basicamente, duas coisas: o documento tem que ser salvo com a extensão .html e deve iniciar com a tag <html>. Todo conteúdo que vai aparecer no site (textos, tabelas, imagens) tem que ficar dentro da tag <body> que é o corpo do site.

Exemplo:

```
<html> tag de abertura
  <head>
    <title> Título da Página </title>
  </head>
  <body> tag de abertura
    Conteúdo do site
```


`</body>` tag de fechamento

`</html>` tag de fechamento

A tag `<head>` é a cabeça do site, ou seja, são informações que são necessárias para o site. Essas informações não vão aparecer no site, mas são de extrema importância para o seu funcionamento. As chamadas para CSS externo, JavaScript, título da página são todas definidas dentro da tag `<head>`.

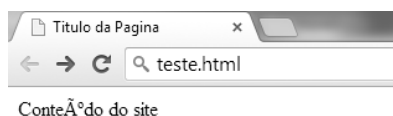
A tag `<title>` vem antes do início da tag `<body>` pois o seu conteúdo não está dentro da página. Poucas tags não possuem fechamento, mas para manter um padrão de desenvolvimento, utiliza-se a barra de fechamento na própria tag de abertura.

Exemplo:

`
` - pois não existe tag de abertura e fechamento para `
`

Importante dizer que as tags que são digitadas erradas ou que não existem não geram erros no seu site, simplesmente não aparece no navegador. Exemplo:

```
<html>
  <head>
    <title> Título da Página </title>
  </head>
  <body>
    Conteúdo do site
  </body>
</html>
```



Reparando no exemplo acima temos a palavra Conteúdo (com o ã no lugar do ú). Isso ocorre porque não foi definida uma linguagem para a página HTML. Toda a programação de uma página de internet é feita em inglês e alguns navegadores não reconhecem os caracteres especiais, como o ç e os caracteres acentuados (á, ú, ü, etc). A codificação utf-8 quer dizer que o site vai utilizar todos os caracteres.

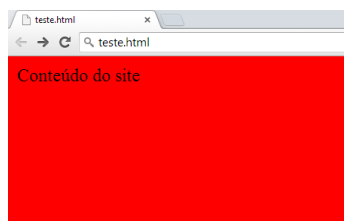
Exemplo:

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
<body>
```

Como já explicado a tag <body> é o corpo do site e todo conteúdo deve estar dentro dela. Para inserir uma cor de fundo no site utilizamos o atributo bgcolor dentro da tag <body>.

Exemplo:

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body bgcolor=red>
    Conteúdo do site
  </body>
</html>
```



As cores podem ser inseridas de duas maneiras. A primeira é utilizando o seu nome em inglês e a segunda é com o seu código hexadecimal. Abaixo uma pequena tabela de exemplo de cores com seus respectivos códigos hexadecimais.

"#FFFFFF"	"#FFCCCC"	"#FFCC99"	"#FFFFCC"
"#CCCCCC"	"#FF6666"	"#FFCC33"	"#FFFF99"
"#C0C0C0"	"#FF0000"	"#FF9900"	"#FFFF00"
"#999999"	"#CC0000"	"#FF6600"	"#FFCC00"
"#666666"	"#990000"	"#CC6600"	"#999900"
"#333333"	"#660000"	"#993300"	"#666600"
"#000000"	"#330000"	"#663300"	"#333300"

O atributo bgcolor deriva da palavra "background color". Para deixar o fundo do site em vermelho utiliza-se `<body bgcolor=red>` ou `<body bgcolor=#FF0000>` (`#FF0000` é o valor hexadecimal para o vermelho).

A formatação que está no editor não é a mesma visualizada na página HTML. Um exemplo disso é escrever um texto em duas linhas no editor, quando a página é aberta o texto estará na mesma linha.

Exemplo:

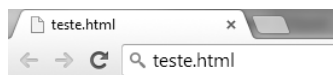
```
<html>
  <body>
    Primeira linha!
    Segunda linha!
  </body>
</html>
```



A tag responsável para fazer uma quebra de linha é a `
`, do inglês **break row**.

Exemplo:

```
<html>
  <body>
    Primeira linha! <br />
    Segunda linha!
  </body>
</html>
```



Primeira linha!
Segunda linha!

Aproveito a oportunidade para falar sobre os acentos e caracteres especiais. Estamos falando de internet, e a internet não aceita muito bem os caracteres especiais. Não estou falando do conteúdo, mas sim de nome de tags e nome de arquivos. Nunca utilizar acentos ou caracteres especiais em nome de arquivos.

Outro macete é que na organização dos arquivos na pasta do seu site, colocar um (underline) na frente do nome das pastas. Isso simplesmente para diferenciar as pastas dos arquivos. O Windows já faz essa separação, mas o Linux ordena os arquivos e pastas em ordem alfabética misturando os dois.

Símbolos

O HTML faz inúmeros símbolos, o que vamos entender aqui é o conceito dos símbolos, pois seria inviável mostrar todos aqui. Todos os símbolos seguem um padrão começando com o `&`, daí vem o código do

símbolo e fecha com o; tudo isso sem espaço. Exemplo:
€ (o código vai mostrar o € do Euro)

Outros Símbolos:

>	>	®	®
<	<	©	©
≤	≤	™	™
≥	≥	≠	≠

Comentário

Os comentários são pedaços de códigos que estão escritos no documento, mas que o navegador não vai executá-los. São utilizados basicamente para duas finalidades: comentar algo em relação ao código ou procurar erros separando pedaços para não serem executados.

<!-- esta parte aqui está dentro do código, mas não vai aparecer na tela -->

Em HTML temos apenas os comentários de bloco, onde tudo o que estiver entre a tag <!-- e a tag --> não vai ser executado.

Atividades

- 1) Criar uma página chamada index.html, escrever o seu nome e colocar um fundo amarelo.
- 2) Criar uma página HTML e colocar o dia da semana na primeira linha e o mês na linha de baixo. Colocar a data atual, mas dentro de um comentário.

FORMATAÇÃO DE TEXTOS

A tag responsável pela formatação do texto é a . O texto a ser formatado deve estar entre a tag de abertura e a tag de fechamento . Os atributos para são:

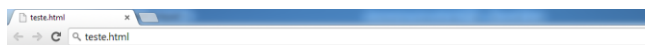
color – cor da fonte (nome da cor escrita em inglês ou seu correspondente em hexadecimal)

size – tamanho da fonte (de 1 até 7 em HTML)

face – tipo da fonte (arial, times new roman, courier)

Exemplo:

```
<html>
  <body>
    <font face=arial size=5 color=red> Esse texto vai ser escrito
em arial, tamanho 5 e cor vermelha </font>
  </body>
</html>
```



Esse texto vai ser escrito em arial, tamanho 5 e cor vermelha

Observação: Os três atributos não necessitam ser utilizados juntos e não tem uma ordem definida. Note que os atributos de uma tag não aparecem na tag de fechamento.

Estilos pré-definidos

O HTML possui tags específicas para definir os títulos e subtítulos da página. As tags utilizadas para isso são as <h1> até a <h6>. Essa tag segue um nível de hierarquia onde a <h1> tem um maior destaque, com uma fonte maior, do que a <h2> e assim sucessivamente. Exemplo:

<html>	
<body>	Estilo 1
<h1>Estilo 1</h1>	
<h2>Estilo 2</h2>	Estilo 2
<h3>Estilo 3</h3>	Estilo 3
<h4>Estilo 4</h4>	Estilo 4
<h5>Estilo 5</h5>	Estilo 5
<h6>Estilo 6</h6>	Estilo 6
</body>	
</html>	

É pela tag <h1> que o Google identifica qual o principal assunto da página.

Outras TAGs – Formatação de texto

Existem outras tags específicas utilizadas na formatação de texto.

 negrito – bold
<i> itálico – italic </i>
<u> sublinhado – underline </u>

<s> riscado – strike </s>
_{Subscrito}
^{Sobrescrito}

<body>
 texto em negrito

 <u><i>itálico e sublinhado</i></u>

 <s> texto riscado </s>

 _{Subscrito} normal
 ^{Sobrescrito}
</body>

texto em negrito
itálico e sublinhado
~~texto riscado~~
Subscrito normal Sobrescrito

Texto Pré-Formatado

Fugindo a regra de que o HTML não segue a formatação que está escrita no código, existe a tag <pre> onde possibilita a inserção de texto pré-formatados.

```
<pre>
Sou um texto
    Sou um texto
        e mantenho os espaços
            e mantenho os espaços
                formatados na edição.
                    formatados na edição.
formatados na edição.
</pre>
```

Note que no exemplo acima, o que é exibido na página HTML segue exatamente a formatação dos espaços no código. A tag <pre> forma um bloco com as informações entre a abertura <pre> e o fechamento </pre> e qualquer frase após a tag de fechamento é colocada em outra linha abaixo do texto formatado.

Por padrão, o texto pré-formatado usa uma fonte monospace, com tamanho de 13px e cor preta. Os textos pré-formatados costumam poupar tempo de formatação, mas não abuse.

Atividades

1) Relacione as tags de formatação de texto com sua utilização:

- | | | | |
|---|-------|-----|----------------------------------|
| 1 | | () | Escreve o texto mais para cima |
| 2 | <u> | () | Texto sublinhado |
| 3 | <s> | () | Escrever o texto mais para baixo |
| 4 | <sub> | () | Texto em negrito |
| 5 | <sup> | () | Texto riscado |

2) Escreva o teorema de Pitágoras: $a^2 = b^2 + c^2$

3) Escreva o texto igual ao exemplo abaixo:

Sou um **texto** e estou me sentindo muito importante porque ~~estou servindo de exemplo~~ em uma atividade de um livro. **Os alunos ficaram muito** felizes pois com certeza, com um ~~pouco~~ de paciência e dedicação vão conseguir ^{finalizar} essa atividade.

4) Escreva a palavra "Dimensionando" e formate-a das duas maneiras:

D I M E N C I O N A N D O

d i m e n c l o n a n d o

LISTAS

Listas não ordenadas

As listas não ordenadas utilizam a tag `` (do inglês **unordered lists**) e simplesmente usam símbolos para a sua formatação, podendo ter três formatações diferentes: circle, square e disc. Se o tipo de formatação não for informado, o código assume a disc como padrão. Exemplo:

Independente se a lista for ordenada ou não ordenada, os itens que vão compor as listas são os `` (do inglês **list items**).

```
<ul>
  <li> primeiro tópico </li>
  <li> segundo tópico </li>
  <li> terceiro tópico </li>
</ul>
```

- primeiro tópico
- segundo tópico
- terceiro tópico

Exemplo 2:

```
<ul type=“circle”>
  <li> tópico 1 </li>
  <li> tópico 2 </li>
</ul>
```

- tópico 1
- tópico 2

```
<ul type=“square”>
  <li> tópico 1 </li>
  <li> tópico 2 </li>
</ul>
```

- tópico 1
- tópico 2

```
<ul type=“disc”>
  <li> tópico 1 </li>
  <li> tópico 2 </li>
</ul>
```

- tópico 1
- tópico 2

Também pode ser feitas listas não ordenadas sem os caracteres de formatação na frente dos itens. Para isso utiliza-se none no atributo type. Exemplo:

```
<ul type=none>  
  <li> tópico 1 </li>      tópico 1  
  <li> tópico 2 </li>      tópico 2  
</ul>
```

Listas ordenadas

As listas ordenadas usam caracteres de ordem para listar seus tópicos. A tag utilizada para ordenar é a (do inglês **ordered lists**) e os seus tipos podem ser:

```
<ol type=1> lista numerada  
<ol type=I> lista usando caracteres romanos em  
maiusculo  
<ol type=i> lista usando caracteres romanos em  
minusculo  
<ol type=A> lista com letras maiúsculas  
<ol type=a> lista com letras minúsculas
```

Além do atributo type, as listas ordenadas ainda tem o atributo start, que nada mais é do que onde inicia a lista. Por padrão, o type=1 e o start=1.

Exemplo:

<code></code>	
<code> primeiro tópico </code>	1. primeiro tópico
<code> segundo tópico</code>	2. segundo tópico
<code> terceiro tópico</code>	3. terceiro tópico
<code></code>	
<code><ol type=I start=3></code>	III. primeiro tópico
<code> primeiro tópico </code>	IV. segundo tópico
<code> segundo tópico</code>	V. terceiro tópico
<code> terceiro tópico</code>	
<code></code>	
<code><ol type=I></code>	III. terceiro tópico
<code><li value=3> terceiro tópico </code>	V. quinto tópico
<code><li value=5> quinto tópico</code>	VII. sétimo tópico
<code><li value=7> sétimo tópico</code>	
<code></code>	

Atividades

1) Montar as seguintes listas:

- | | | | |
|----|---|----|---|
| a) | <ul style="list-style-type: none">i. Batata Fritaii. Batata Assadaiii. Camarão
viii. Frango Fritoix. Pão de Queijox. Lasanha | b) | <ul style="list-style-type: none">1. Introdução2. Desenvolvimento<ul style="list-style-type: none">◦ Objetivos◦ Fundamentação◦ Testes3. Conclusão |
|----|---|----|---|

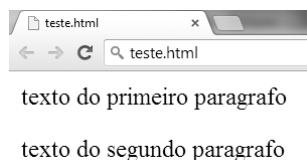
- c)
1. Reunião início
 - Trabalho
 - Procurar sobre o assunto
 - Montar a pesquisa
 - Formatar norma ABNT
 - Apresentação
 - Montar a apresentação
 - Treinar a apresentação
 2. Reunião término
 3. Apresentar

Parágrafos

O HTML também separa os textos em parágrafos. As diferenças básicas entre um parágrafo e uma quebra de linha são que um parágrafo pode ser alinhado e tem um espaçamento entre linhas maior antes e depois dos parágrafos.

Exemplo:

```
<html>
  <body>
    <p> texto do primeiro parágrafo </p>
    <p> texto do segundo parágrafo </p>
  </body>
</html>
```



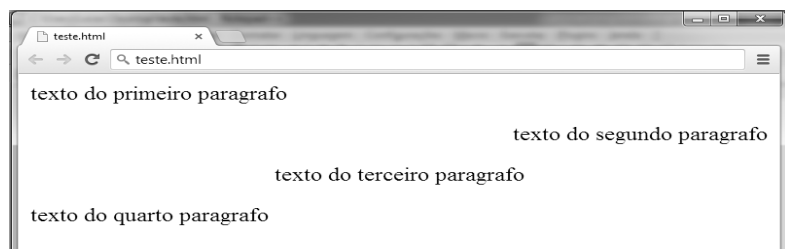
Alinhamentos

Pensando em alinhamentos, existem duas direções possíveis: alinhamento na horizontal e alinhamento na vertical. Nesse capítulo, estudaremos os alinhamentos na horizontal. Dentro dos horizontais temos quatro tipos: esquerda, direita, centro e justificado. Todos os elementos em HTML (tabelas, textos, imagens, etc) podem ser alinhados.

Por padrão todos os elementos vêm alinhados a esquerda, por isso esse alinhamento é bem pouco utilizado.

Exemplo:

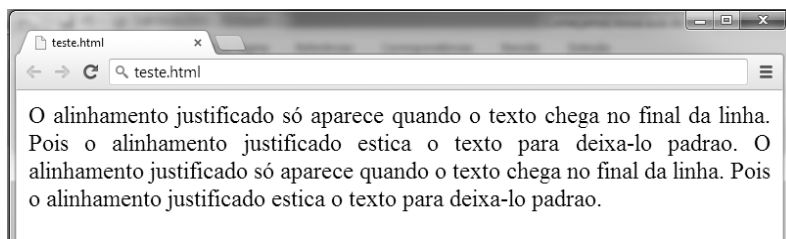
```
<html>
  <body>
    <p> texto do primeiro parágrafo </p>
    <p align=right> texto do segundo parágrafo </p>
    <p align=center> texto do terceiro parágrafo </p>
    <p align=justify> texto do quarto parágrafo </p>
  </body>
</html>
```



No exemplo temos quatro parágrafos. Note que o primeiro está alinhado à esquerda mesmo sem ter nada programado. O último parágrafo está com alinhamento justificado, mas só conseguimos ver se o texto chegar ao final da linha.

Exemplo:

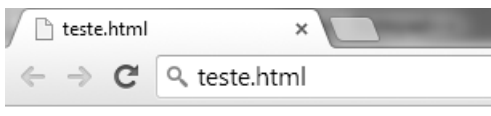
```
<html>
  <body>
    <p align=justify>
      O alinhamento justificado só aparece quando o
      texto chega ao final da linha. Pois o alinhamento
      justificado estica o texto para deixá-lo padrão. O
      alinhamento justificado só aparece quando o texto
      chega ao final da linha. Pois o alinhamento
      justificado estica o texto para deixá-lo padrão.
    </p>
  </body>
</html>
```



Traço ou Linha

A tag `<hr>` faz uma linha simples, que por padrão vai de um lado para o outro na página.

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



Para obtermos um efeito mais interessante, usamos alguns atributos para formatar essa tag.

size - espessura da linha

color - cor da linha

width - comprimento da linha (se não for informado ele fica com 100%)

align - alinhamento da linha (caso o comprimento for menor que 100%)

noshade - define que a linha não possui sombra e é sólida

Exemplo:

```
<hr size=10 color=red width=50% align=left />
```



Exemplo 2:

```
<hr size=10 width=30% />
<hr size=10 width=30%
noshade />
```



Com um pouco de criatividade pode-se fazer alguns efeitos legais.

```
<html>
  <body>
    <hr size=10 color=purple width=50% align=left>
    <hr size=10 color=purple width=40% align=left>
    <hr size=10 color=purple width=30% align=left>
    <hr size=10 color=purple width=20% align=left>
    <hr size=10 color=purple width=10% align=left>
  </body>
</html>
```



Atividades

1) Montar uma página com fundo preto e três parágrafos da seguinte forma:

- O primeiro com o tamanho da letra em 5, cor azul e sublinhado.
- O segundo com o tamanho da letra em 4, cor verde, negrito e centralizado.
- O terceiro com o tamanho de letra em 7, cor branca, sublinhado e itálico.

Observação: Cada parágrafo deve ser separado por um traço na cor branca.

2) Montar os desenhos abaixo, um em cada página HTML.

```

*****
*           *
*       *   *
*   ***   *
*       *   *
*           *
*           *
*       *   *
*   ***   *
*       *   *
*           *
*****

```

```

    @@@@
  @@@@@@
  @@@@@@
  @@@@@@ @
  @@@@@@ @@@
    |||  @@@
    |||  //
    |||  //
    |||  //
    |||  //
-----

```

TABELA

Criando Tabelas

A tabela é um recurso muito utilizado nos sites. Para criar uma tabela utilizamos a tag `<table>`. Uma tabela é formada por linhas e colunas (mais conhecidas como células). As linhas são definidas pela tag `<tr>` (do inglês *table row*) e as células definidas pela tag `<td>` (do inglês *table data*). Lembrando que a tabela deve estar dentro da tag `<body>`, pois aparecerá no site. Para melhor visualização, colocamos uma borda na tabela (`border=1`).

Exemplo:

A tabela abaixo tem uma linha e duas células.

```
<table border=1>
```

```
<tr>
```

```
<td> célula 1 </td>
```

```
<td> célula 2 </td>
```

```
</tr>
```

```
</table>
```

célula 1	célula 2
----------	----------

A tabela abaixo tem duas linhas e duas células em cada linha.

```
<table border=1>
```

```
<tr>
```

```
<td> célula 1 </td>
```

```
<td> célula 2 </td>
```

```
</tr>
```

```
<tr>
```

```
<td> célula 3 </td>
```

```
<td> célula 4 </td>
```

célula 1	célula 2
célula 3	célula 4


```
</tr>  
</table>
```

A tabela abaixo tem três linhas e uma célula em cada linha.

```
<table border=1>  
  <tr>  
    <td> célula 1 </td>  
  </tr>  
  <tr>  
    <td> célula 2 </td>  
  </tr>  
  <tr>  
    <td> célula 3 </td>  
  </tr>  
</table>
```

célula 1
célula 2
célula 3

Note que o tamanho das tabelas estão relacionadas com seus conteúdos. Para definirmos os tamanhos precisamos de dois atributos width (largura) e height (altura). Os valores para esses atributos podem ser definidos em pixels ou em porcentagem.

Exemplo:

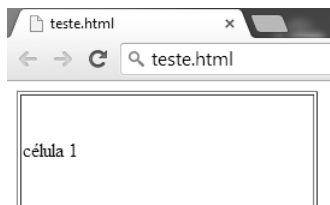
```
<table border=1 width=500 height=300>
```

No exemplo acima temos uma tabela que tem 500 pixels de largura e 300 pixels de altura. Quando utilizamos pixels o tamanho da tabela vai variar de acordo com a resolução que o computador que está visualizando a página tiver usando. Se a resolução for maior, menor vai ser a tabela.

```

<html>
  <body>
    <table border=1 width=250 height=100>
      <tr>
        <td> célula 1 </td>
      </tr>
    </table>
  </body>
</html>

```



Utilizando a porcentagem, o elemento (que pode ser uma tabela, uma imagem, entre outros) não tem vínculo com a resolução do vídeo no computador:
Exemplo:

```
<table border=1 width=60% height=20%>
```

No exemplo acima, a tabela vai ocupar 60% de largura da página e 20% de altura da página.

Mesclando células

Mesclar células nada mais é do que unir duas ou mais células. Quando utilizamos tabelas, podemos mesclar as células na horizontal (na mesma linha) e na vertical (na mesma coluna).

Para mesclar células na mesma coluna utiliza-se o atributo `rowspan="numero de células que quer mesclar"`. O atributo `colspan` é utilizado da mesma forma, mas para mesclar células na mesma linha.

Resumindo, o rowspan significa a expansão em linha, fazendo com que a célula ocupe o espaço de mais de uma célula na vertical. O colspan é a expansão das colunas, onde a célula vai ocupar o espaço de mais de uma célula, mas na horizontal.

No exemplo abaixo, a primeira célula está mesclada e ocupando o espaço de 2 células na horizontal:

```
<html>
  <boby>
    <table border=1 width=250 height=100>
      <tr>
        <td colspan=2> célula 1 </td>
      </tr>
      <tr>
        <td> célula 2 </td>
        <td> célula 3 </td>
      </tr>
    </table>
  </boby>
</html>
```



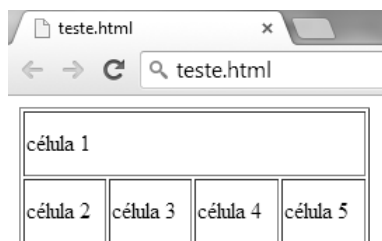
No exemplo abaixo, a primeira célula está ocupando o espaço de 4 células na horizontal:

```
<table border=1 width=250 height=100>
  <tr>
    <td colspan=4> célula 1 </td>
  </tr>
```

```

<tr>
  <td> célula 2 </td>
  <td> célula 3 </td>
  <td> célula 4 </td>
  <td> célula 5 </td>
</tr>
</table>

```



A screenshot of a web browser window titled 'teste.html'. The browser's address bar shows 'teste.html'. Below the browser window, a table is displayed. The table has two rows and four columns. The first row contains a single cell labeled 'célula 1'. The second row contains four cells labeled 'célula 2', 'célula 3', 'célula 4', and 'célula 5' from left to right.

célula 1			
célula 2	célula 3	célula 4	célula 5

No exemplo abaixo, a primeira célula está ocupando o espaço de 2 células na vertical:

```

<table border=1 width=250 height=100>
  <tr>
    <td rowspan=2> célula 1 </td>
    <td> célula 2 </td>
  </tr>
  <tr>
    <td> célula 3 </td>
  </tr>
</table>

```



A screenshot of a web browser window titled 'teste.html'. The browser's address bar shows 'teste.html'. Below the browser window, a table is displayed. The table has two rows and two columns. The first row contains a cell labeled 'célula 1' and a cell labeled 'célula 2'. The second row contains a cell labeled 'célula 3' and an empty cell. The 'célula 1' cell spans both rows.

célula 1	célula 2
	célula 3

Atividades

1) Montar uma tabela com 7 linhas com os dias da semana. Formatar cada célula de maneira diferente (Mudar cor de fundo e formatar a fonte).

2) Montar uma tabela com 2 linhas e 5 colunas com o fundo preto. Mesclar a primeira linha e escrever "Vogais" formatada com tamanho 6 e cor branca. Na próxima linha, colocar em cada coluna uma vogal. Formatar a segunda linha com o texto em cor cinza e tamanho 5.

3) Para exercitar os comandos colspan e rowspan, fazer as seguintes tabelas:

a)

b)

c)

4) Montar um horário do colégio, com os dias da semana e as aulas. Formatar da melhor forma possível.

Desafio

Para os iniciantes, as tabelas podem ser complicadas. Para testar seus conhecimentos sobre os conceitos de `<table>`, `<td>`, `<tr>`, `colspan` e `rowspan` segue um desafio:

ALINHAMENTOS EM TABELAS

Assim como os textos, os alinhamentos são utilizados em tabelas, imagens, vídeos, etc.

Exemplo:

```
<html>
  <body>
    <table border=1 width=300 height=100>
      <tr>
        <td align=center bgcolor=green> célula 1
        </td>
        <td align=right bgcolor=yellow> célula 2 </td>
      </tr>
    </table>
  </body>
</html>
```



No exemplo acima temos uma tabela com 300 pixels de largura e 100 pixels de altura. A primeira célula está com o fundo verde e o texto alinhado ao centro. A segunda célula está com o fundo amarelo e o texto alinhado a direita.

Alinhamentos Verticais

Os alinhamentos verticais usam o atributo valign (do inglês vertical align). Os três alinhamentos verticais possíveis são:

top: em cima

bottom: em baixo

middle: no centro



célula 1	célula 2	célula 3
----------	----------	----------

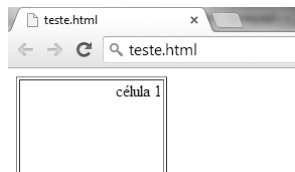
Exemplo:

```
<html>
  <body>
    <table border=1 width=300 height=100>
      <tr>
        <td valign=top> célula 1 </td>
        <td valign=middle> célula 2 </td>
        <td valign=bottom> célula 3 </td>
      </tr>
    </table>
  </body>
</html>
```

Pode-se utilizar os alinhamentos verticais e os horizontais na mesma tag. No exemplo abaixo o conteúdo está alinhado a direita na horizontal e em cima na vertical.

```
<html>
  <body>
    <table border=1 width=150 height=100>
      <tr>
        <td align=right valign=top> célula 1 </td>
      </tr>
    </table>
  </body>
</html>
```

```
</tr>
</table>
</body>
</html>
```



Todos os exemplos acima estão sendo aplicados no conteúdo da célula. Isso acontece porque os alinhamentos estão dentro da tag `<td>`. Para alinhar uma tabela ao centro é preciso colocar o atributo `align` dentro da tag `<table>`.

Exemplo:

```
<table align=center>
<tr>
  <td> Célula 1 </td>
</tr>
</table>
```

Olhando com mais atenção na tabela criada, ela parece estar com uma borda dupla. Na verdade ela tem uma borda interna e outra borda externa. O espaço entre as duas bordas é controlado pelo atributo `cellspacing`. Quanto maior for o valor de `cellspacing`, maior será a distância entre as duas bordas.

Exemplo 1: `<table border=1>` - esse é o padrão

Exemplo 2: `<table border=1 cellspacing=0>`

Exemplo 3: `<table border=1 cellspacing=10>`

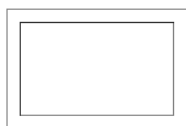
Exemplo 1:



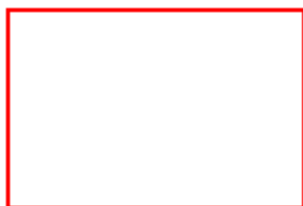
Exemplo 2:



Exemplo 3:



Para alterar a cor da borda na tabela usamos o atributo `bordercolor` dentro da tag `<table>`. No exemplo abaixo a tabela ficará com a borda vermelha.



```
<table border=1 width=150 height=100 cellspacing=0  
bordercolor=red>  
  <tr>  
    <td> </td>  
  </tr>  
</table>
```

As tabelas ainda podem ter legendas. A tag <caption> é responsável por isso e deve ser colocada depois da tag <table> e antes da primeira tag <tr>.

Exemplo:

```
<table border=1>
```

```
  <caption> legenda da tabela </caption>
```

```
  <tr>
```

```
    <td> célula 1 </td>
```

```
    <td> célula 1 </td>
```

```
  </tr>
```

```
</table>
```

legenda da tabela

célula 1	célula 1
----------	----------

Atividades

1) Criar uma tabela para ajudar o professor a controlar a nota dos alunos. Segue modelo:

Alunos	Notas		
	Teórico	Prático	Média
Joaquim	7.8	10.0	8.9
Pedro	4.5	8.0	6.3
Maria	9.8	6.7	8.3
João	8.9	5.6	7.3

2) Montar uma tabela de classificação do campeonato espanhol com as seguintes formatações:

Classificação Campeonato Espanhol				
Class	Equipe	Pontos	Jogos	Vitórias
1º	Barcelona	71	29	23
2º	Real Madrid	67	29	22
3º	Atlético de Madrid	62	29	19
4º	Valencia	61	29	18
5º	Sevilha	58	29	18
6º	Villarreal	50	29	14
7º	Málaga	45	29	13

3) Elaborar a seguinte tabela para organizar as tarefas do TCC.

Organização TCC	
Joaquim	- Pesquisa de campo - Tabulação dos Dados
Arthur	- Análise do Projeto - Criação dos Diagramas - Programação
Anastácia	- Design do Site - Documentação

IMAGENS

Inserindo Imagens

Já imaginou como seriam os sites sem imagens? Podemos colocar uma imagem que esta salva na pasta do site ou uma imagem qualquer da internet. O recomendado é salvar a imagem na pasta, por que se a imagem estiver em outro site ela pode ser removida, consequentemente o seu site também ficara sem a imagem.

Sintaxe ``



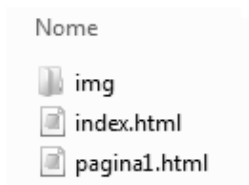
O atributo `src` vem do inglês **source** e quer dizer origem.

Utilizando o código acima a imagem é inserida no site no tamanho original (que foi salva). Para redimensionar utilizaremos dois atributos já conhecidos: `width` e `height`.

Exemplo:

```
<html>
  <body>
    <img src=html.jpg width=150 height=130>
  </body>
</html>
```


O comando acima está inserindo uma imagem com o nome html.jpg que está na mesma pasta da página e vai possuir 150 pixels de largura e 130 pixels de altura.



Para a estrutura de pastas do site ficar organizado, as imagens ficam em uma pasta separadas dos arquivos do site. Por padrão é criada uma pasta chamada img e dentro dela inserida todas as imagens. Lembre-se de alterar o caminho das imagens.

Exemplo:

```
<img src=img/html.jpg width=150 height=130>
```

Uma dica muito importante é que todas as imagens visualizadas no navegador são baixadas para uma pasta temporária no computador. Sabendo disso, evite carregar imagens muito grandes. Um grande erro é ter um arquivo de imagem muito grande e simplesmente redimensionar com width e height. A maneira correta é ajustar o tamanho do arquivo de imagem.

O atributo alt é utilizado para descrever a imagem caso a imagem não consiga ser exibida na página e o atributo title mostra uma informação da imagem cada vez que o cursor é posicionado em cima da imagem.

Exemplo:

```
<img src=html.jpg title="Aula de HTML">
```



```
<img src=html.jpg alt="Aula de HTML">
```



A tag alt também inclui as palavras informadas nos mecanismos de buscar utilizado por softwares de acessibilidade e usabilidade.

Imagens como plano de Fundo

Em HTML também conseguimos colocar uma imagem ou uma textura como plano de fundo. Esse recurso é utilizado no CSS, pois em HTML não há como alterar o tamanho da imagem nem o seu posicionamento. Com a imagem na mesma pasta, em HTML fica assim:

```
<body background="textura.jpg">
```

Qual o tipo de imagem usar

As extensões definem a taxa de compressão das imagens, afetando a qualidade e o seu tamanho. Todos os navegadores suportam pelo menos três extensões de imagens:

JPG – imagens com uma boa qualidade, mas tem a taxa de compactação ruim, significando arquivos maiores (fotos e texturas)

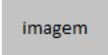
GIF – utiliza apenas 256 cores, recomendada para animações e ícones

PNG – veio para substituir o GIF, mas as vezes não consegue uma taxa de compactação boa, deixando a imagem muito grande

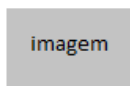
As extensões GIF e PNG permitem transparência.

Atividades

1) Montar as seguintes estruturas usando tabelas:

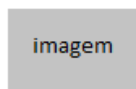
- a) Eu sou um texto e estou na esquerda da imagem.  Eu sou um texto e estou na direita da imagem.

- b) Eu sou um texto e estou em cima da imagem.



Eu sou um texto e estou em baixo da imagem.

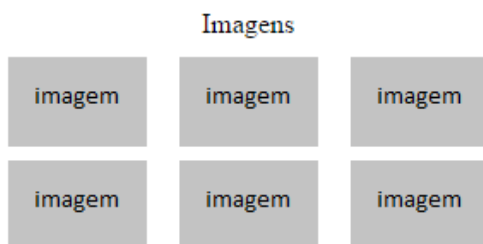
c)



Eu sou um texto e
estou em baixo da
imagem.

Eu sou um texto e
estou em baixo da
imagem.

2) Criar uma galeria com 6 imagens, 3 em cada linha.
Para isso utilizar uma tabela.



3) Criar um menu com 5 imagens. Não é preciso fazer os links nas imagens, apenas montar a estrutura. Aproveite para inserir uma textura na página.

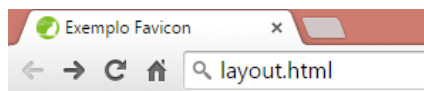
Ícones na aba do navegador – favicon

O Favicon são ícones (desenhos) que aparecem do lado do nome do site na aba do navegador. Esses ícones podem ser de dois tipos diferentes de arquivo. Pode ser PNG ou ICO. Há diversos sites que transformam sua imagem em um arquivo .ico (ícone).

O tamanho padrão dos favicons varia de acordo com o dispositivo que está sendo usado, pois pode ser utilizado em páginas de internet (tamanho padrão de 16px x 16px) ou tamanhos maiores quando utilizados em smartphones ou tablets.

O ícone não faz parte do corpo do site, então ele deve ser colocado antes da tag <body>. Exemplo:

```
<head>  
  <link rel="icon" href="teste.ico" />  
  <title> Exemplo Favicon </title>  
</head>
```



LINKS

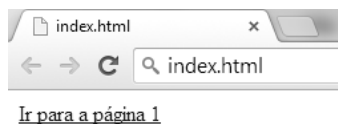
Para que servem os links

Os links são utilizados para fazer a navegação do site. Um bom exemplo de links são os menus. Um link pode ser interno ou externo. Um link interno abre uma página do mesmo site e um link externo abre uma página de outro site.

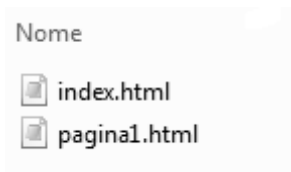
Sintaxe nome que vai aparecer na tela

Exemplo:

```
<html>
  <head>
    <meta charset=utf8>
  </head>
  <body>
    <a href=pagina1.html> Ir para a página 1 </a>
  </body>
</html>
```



O atributo href é o caminho onde está o arquivo chamado no link. Quando os dois arquivos estão salvos na mesma pasta, basta informar o nome e a extensão do arquivo no atributo href. Para o link funcionar é preciso que a página referenciada esteja exatamente igual ao href.



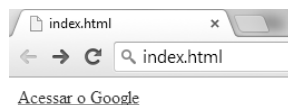
Se os arquivos não estão na mesma pasta é preciso informar o caminho. No exemplo abaixo o arquivo chamado pagina1.html está dentro da pasta link.



```
<a href=links/pagina1.html> Ir para a página 1 </a>
```

Para fazer um link externo basta informar o seu endereço na internet. Abaixo exemplo de um link que acessa o site do Google.

```
<html>
  <body>
    <a href=http://www.google.com.br> Acessar o
Google </a>
  </body>
</html>
```



Da mesma maneira que os links internos, os externos devem estar totalmente corretos. Note que todo link vem a cor azul quando criado e roxo após ser acessado. Essas cores são padrões dos navegadores que podem ser alteradas, mas esse assunto será trabalhado nesse mesmo capítulo, mas em uma próxima aula.

Por padrão os links abrem na mesma página que ele foi clicado. Para especificar o local de abertura do link utiliza-se o atributo chamado target.

```
<a href=http://www.google.com.br target=_blank>  
Google </a>
```

target=_blank configura o link para abrir em uma nova aba, mantendo o site de origem intacto.

Link para navegação na mesma página

Com páginas muito grandes, podemos inserir links para facilitar a navegação, otimizando a procura das informações. Esses links conduzem a outro local dentro da mesma página. Um exemplo de utilização é quando o menu e o conteúdo estão na mesma página. Note que nesse exemplo o link não vai chamar outra página, mas conteúdos na mesma página.

Link que vai ser clicado para redirecionar

Link que vai ser clicado para redirecionar

```
<a href=#perg1> Qual o melhor navegador? </a>  
<a href=#perg2> Você prefere notebook ou desktop?  
</a>
```

Local onde o link vai ser redirecionado quando for clicado

```
<a name="perg1" id="perg1">Na minha opinião, o  
melhor...</a>  
<a name="perg2" id="perg2">Eu prefiro  
notebook!</a>
```

Observação: o nome do href, name e id é criado pelo programador. O importante é que os três sejam iguais. Utilizamos o atributo name e id porque alguns navegadores não entendem um ou o outro.

Formatação de links

Como já vimos os links em HTML tem um padrão de cor já definidos: o azul para o link que ainda não foi acessado e o roxo para o link já acessado. Podemos alterar essas cores nos links dos nossos sites. O ideal é fazer essa formatação utilizando o CSS, mas em HTML também é possível. Definimos a cor do link para toda a nossa página HTML.

Exemplo:

```
<body link=red vlink=green>
```

Com o código acima, definimos que nessa página os links não visitados serão vermelhos e os links já visitados serão verdes.

Imagem como link

Uma imagem também pode ser utilizada como link. Para isso basta colocar o código para inserir a imagem dentro da tag <a>.

Exemplo:

```
<a href=contato.html> <img src=contato.jpg> </a>
```

Ao clicar na imagem, a página contato.html vai ser aberta.

Atividades

1) Criar 2 páginas e salvá-las como pag1.html (com cor de fundo azul) e pag2.html (com cor de fundo verde). Fazer um link em cada para chamar a outra página.

2) Montar uma página com 2 links e altere a cor padrão dos links para preto os links visitados e vermelho os links não visitados. O primeiro para abrir a página do Uol, mas deve ser aberto em outra aba e o segundo para abrir o Youtube na mesma página.

3) Montar em apenas uma página várias perguntas e respostas. As perguntas devem vir no início da página e as respostas logo abaixo. Quando clicar na pergunta, direcionar diretamente para a sua resposta. Ao final de cada resposta, ter um link para voltar para o início da página.

4) Montar uma página com 6 imagens com links para outros sites (você escolhe as imagens e os sites que vão ser direcionados). Escolher uma textura na internet e colocar como fundo da página. Formatar a página com o maior número de recursos já aprendidos.

Extra

Para ver de maneira global este exercício, transcrevemos aqui todo o código utilizado.

```
<html>
<head>
  <title>Link Aleatorio</title>
<script>
  var enderecos = new Array("cliente.html",
"contato.html", "mapa.html", "fotos.html", "vídeos.html
")

  function linkAleatorio(){
    aleat = Math.random() * enderecos.length
    aleat = Math.floor(aleat)
    window.location=enderecos[aleat]
  }

</script>
</head>

<body>
  <a
href="javascript:linkAleatorio()">LinkAleatorio</a>
</body>
</html>
```

Projeto HTML 1: Turismo em Curitiba

Desenvolver um site com no mínimo 4 páginas utilizando o conhecimento adquirido até o momento. Usar:

- formatação de texto
- alinhamentos
- tabelas
- imagens
- links
- traços
- listas

Capriche no design do site. O projeto não é simplesmente colocar todos os itens relacionados acima. Ele tem que estar em harmonia e com cores combinando. Não se preocupem esse é o primeiro projeto. Bons estudos e dediquem-se para assimilar os conhecimentos passados.

VÍDEO E ÁUDIO

Vídeo

Outra maneira de entretenimento nos vídeos. Os navegadores suportam os segs vídeos:

.mp4	O Youtube utiliza esse formato. É o formato mais recente e considerado tendência mundial.
.mpg ou .mpeg	É o mais comum, pois é suportado por todos os sistemas operacionais e navegadores mais populares.
.wmv	Desenvolvido pela Microsoft é bem comum, mas não rodam em outros sistemas operacionais sem a instalação de um plugin.
.avi	Suportado por todos os computadores que utilizam o Windows. Desenvolvido pela Microsoft.
.swf ou .flv	Este formato requer um plugin para funcionar. Desenvolvido pela Macromedia.

Há duas maneiras de inserir um vídeo na página. A primeira é utilizando a tag , já conhecida, trocando apenas de src para dynsrc (dynamic).

Exemplo:

```
<img dynsrc=video.mpeg>
```

O código acima insere o vídeo na página em tamanho em que o vídeo foi salvo. Utilizam-se os atributos width e height para formatar a largura e a altura do vídeo.

Exemplo:

```
<img dynsrc=video.mpeg width=300 height=250>
```

O vídeo acima foi inserido com 300 pixels de largura e 250 pixels de altura.

A outra opção utiliza HTML5 que não é nosso foco, mas vamos exemplificar esse conteúdo. A tag utilizada no HTML5 é a `<embed>` e é utilizada da seguinte maneira. Exemplo:

```
<embed src=video.mpeg>
```

Com a tag `<embed>` o vídeo ganha um codec específico para vídeos adicionando botões de navegação.



Áudio

Existem duas possibilidades de utilizar o recurso de áudio em páginas HTML: como som de fundo da página (automático) ou quando o usuário clica para ouvir a música.

Exemplo:

```
<a href="audio.mp3"> Clique para ouvir! </a>
```

Na página vai aparecer o texto que estiver escrito entre a tag de abertura e a tag de fechamento. Ao clicar no texto, o site chama o programa padrão instalado no computador e executa o arquivo.

Os principais formatos suportados são:

mp3 – possui uma ótima qualidade e todos os navegadores suportam

wav – desenvolvido pela Microsoft e IBM (o Google Chrome não suporta)

wma – desenvolvido pela Microsoft também possui uma ótima qualidade, mas utiliza maior espaço

Som no fundo da página

```
<bgsound src="audio.mp3" loop="-1">
```

Essa tag foi criada pela Microsoft e deve ser colocada dentro da tag <head>. O atributo loop é o número de repetições que o som vai tocar. Com o valor -1 o som vai tocar enquanto o usuário estiver na página.

Pode-se utilizar a tag <embed> para a mesma finalidade. Caso os atributos width e height são diferentes de 0 ou não colocados na tag, uma barra com controles aparecerá na tela.

Exemplo:

```
<embed          src="audio.mp3"          autostart="true"  
loop="true">
```



O atributo `autostart` pode ser `true` (para o som iniciar automaticamente) ou `false` (o usuário precisa clicar no botão play).

Atividades

- 1) Montar uma página estilo galeria com fotos e vídeos. Use a sua criatividade e o conhecimento obtido.
- 2) Criar uma página divulgando um CD. Essa página deve ter a imagem da capa do CD e a possibilidade de escutar as faixas. Use a criatividade e o número máximo de recursos.

FORMULÁRIOS

Formulários são estruturas onde os usuários podem enviar dados para uma página. Esses dados podem ou não ser tratados ou armazenados. Dependendo apenas do objetivo do formulário. Toda estrutura do formulário deve estar dentro da tag <form>, que deve ser aberta e fechada.

Exemplo:

```
<form action="enviar.php" method="POST"> //  
abertura da tag do formulário  
    // conteúdo do formulário  
</form> // fechamento da tag do formulário
```

A tag form possui alguns parâmetros:

action – é o script ou a página onde os dados serão submetidos. Caso os dados vão receber algum tratamento, é esse script que vai fazê-lo.

method – é o método de envio de dados. Os métodos existentes são dois:

POST: é o método aconselhável na utilização de formulários. Os dados são transmitidos de uma maneira transparente, invisível para o usuário.

GET: os dados são transmitidos via URL (Uniform Resource Locator - endereço de um recurso disponível em uma rede. Uma URL tem a seguinte estrutura: protocolo://máquina/caminho/recurso).

Ficando assim, visíveis as variáveis e seus conteúdos quando as mesmas forem transmitidas. Isso torna a transmissão menos utilizada, porque pessoas com interesses maléficos podem pegar essas informações.

Formulários

Um formulário é composto por entradas de dados. Existem diversos tipos de entradas de dados, sendo a maioria definida pela tag **input**. As entradas são conhecidas como elementos. Todo elemento deve possuir o parâmetro **name**, que é utilizado para identificar a variável onde o dado está contido no script de destino. Todos os elementos devem estar entre as tags de abertura e fechamento do form.

Campo Texto

Campo de texto comum. Sintaxe:

```
<input type="text" name="" size="" maxlength="" value="" title=""/>
```

Exemplo:

End: `<input type="text" name="endereco" size="30" maxlength="50" value="" title="Digite o Endereço"/>`

End:

size – define a largura do campo.

maxlength – define o número máximo de caracteres que pode ser digitado dentro do elemento.

value – quando se tem a necessidade de pré-definir uma valor para o elemento. Este valor pode ser alterado pelo usuário.

title – define uma legenda que é exibida ao posicionar o mouse sobre o campo.

Campo Senha

Os parâmetros são quase todos iguais ao do campo texto, mudando apenas o seu tipo. Quando o usuário digita um texto neste campo, o mesmo é substituído por “*”.

O campo do tipo password não possui nenhum tipo de criptografia, apenas é exibido uma máscara nos caracteres que foram digitados. Sintaxe:

```
<input type="password" name="" size=""  
maxlength="" value="" title=""/>
```

Exemplo:


Senha: `<input type="password" name="senha" size="30" maxlength="30" value="senha" title="Digite a Senha"/>`

Senha:

Um atributo bastante utilizado é o placeholder. Ele informa o usuário o que deve ser escrito dentro de um campo text ou password. Ao começar a digitar no campo, o texto informado desaparece.

Exemplo:

```
<input type="text" name="idade" placeholder="Digite sua idade"/>
```



Botão Radio (Radio Button)

Utilizado quando possui múltiplas alternativas onde o usuário só pode marcar uma das opções. Para todas as opções que pertencem ao mesmo “grupo” é preciso colocar o mesmo nome no parâmetro name em todas. Sintaxe:

```
<input type="radio" name="" value="" checked/>
```

value – se este elemento for selecionado, é o valor que será passado para sua página de destino.

checked – quando este atributo é declarado, o elemento tem o seu estado inicial como selecionado. (já abre o formulário selecionado, podendo ser alterado se o usuário preferir)

Exemplo:

```
<input type="radio" name="sexo" value="M" checked/> Masculino
```

```
<input type="radio" name="sexo" value="F" /> Feminino
```

☒ Masculino ☐ Feminino

Se o atributo name for igual simboliza que os elementos pertencem ao mesmo grupo, podendo assim somente um deles ser marcado. O atributo value é que diferencia os dois campos.

Botão de Checagem (Checkbox)

Utilizado em campos de múltiplas seleção onde o usuário pode marcar quantas opções ele quiser. Todos os campos devem ter nomes diferentes.

```
<input type="checkbox" name="" value="" id="" checked/>
```

id – inclui o texto do botão de checagem e do botão radio na hora da seleção.

Exemplo:

```
<label for="teste">  
  <input type="checkbox" name="r_email"  
  id="r_email" /> Desejo receber e-mail...  
</label>
```

☐ Desejo receber e-mail...

Utilizando o atributo id, junto com a tag <label> o usuário pode clicar no texto ou no campo para que seja feita a seleção do elemento. Esse parâmetro pode ser usado no botão de checagem e no botão radio.

Fieldset

Tag utilizada para fazer um efeito de uma borda, geralmente utilizada para separar grupos de opções, podendo ou não ser inserido um título.

```
<fieldset style="width: 30%;" >
    <legend> Legenda do Fieldset</legend>
    // as opções devem ir dentro das tags de
    inicio e fechamento do elemento fieldset
</fieldset>
```

A sub-tag legend é responsável pelo título. Para definir o tamanho do fieldset é preciso utilizar CSS. (Estudaremos CSS no decorrer da apostila)

Exemplo:

```
<fieldset style="width: 30%;" >
    <legend> Informativo</legend>
    <input type="checkbox" name="info" value="S" />
    Desejo receber informações no meu e-mail.
</fieldset>
```

Informativo

☐ Desejo receber informações no meu e-mail.

Botão Submeter (Submit)

Um elemento do tipo submit cria um botão para enviar os dados para o local ou endereço definido na action do formulário.

```
<input type="submit" name="" value="" />
```

value – nome que vai aparecer no botão.

title – define uma legenda que é exibida ao posicionar o mouse sobre o campo.

Exemplo:

```
<input type="submit" name="enviar" value="Enviar" />
```



Botão Image

Sua função é a mesma do que a do botão submit. Criando um botão personalizado utilizando uma imagem

```
<input type="image" name="" src="" title="" />
```

src – define a imagem que vai ser carregada como botão. (se a imagem não estiver na mesma pasta, é preciso informar seu caminho completo)

Exemplo:

```
<input type="image" name="calcular" src="dinheiro.gif" width="30" height="36"/>
```



Botão Reset

Quando o botão reset é utilizado todos os elementos do formulário voltam ao seu estado inicial, ou seja, os valores especificados no campo value de cada elemento.

```
<input type="reset" name="" value="" />
```

Botão

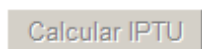
O elemento do tipo button cria um botão padrão, que é utilizado para ativar outros scripts. Um exemplo da sua utilização é nas funções javascript.

```
<input type="button" name="" value="" title="" disabled/>
```

disabled - este atributo serve para fazer com que o botão fique desabilitado.

Exemplo:

```
<input type="button" name="calcular" value="Calcular IPTU" disabled/>
```



Atividades

1) Montar os seguintes formulários:

a)

Nome:

Cidade:

Idade:

b)

Nome:

Login:

c)

Nome:

Endereço:

Telefone:

Sexo: ☐ Masculino ☐ Feminino

Select

Cria uma lista de seleção (menu drop-down) que permite que o usuário escolha uma ou mais opções. A sub-tag <option> cria as opções. O conteúdo mostrado na tela, fica entre a tag <option> conteúdo </option>. O atributo value é o valor a ser gravado/transmitido quando o usuário seleciona uma opção. A exibição da barra de rolagem é automática e fica a cargo do browser.

```
<select name="" size="" multiple>
  <option value=""> texto da opção </option>
</select>
```

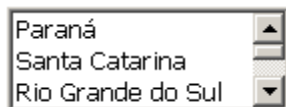
size – Define o número de opções visíveis na tela por vez. Por default esse parâmetro vem com o valor 1.

multiple – se for utilizada as teclas Ctrl ou Shift, pode ser selecionado mais de uma opção. A tecla Ctrl seleciona opções não adjacentes e a tecla Shift seleciona opções adjacentes.

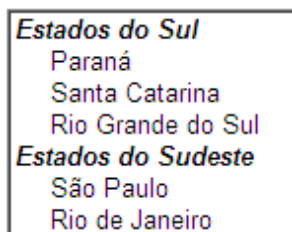
O texto que aparece na tela deve ser colocado entre as tags <option> e </option>.

Exemplo:

```
<select name="Estado" size="3" multiple>
  <option value="PR"> Paraná </option>
  <option value="SC"> Santa Catarina
</option>
  <option value="RS"> Rio Grande do Sul
</option>
  <option value="SP"> São Paulo </option>
</select>
```



```
<select name="Estado" size="7" multiple>
  <optgroup label="Estados do Sul">
    <option value="PR"> Paraná </option>
    <option value="SC"> Santa Catarina
  </option>
    <option value="RS"> Rio Grande do Sul
  </option>
</optgroup>
  <optgroup label="Estados do Sudeste">
    <option value="SP"> São Paulo </option>
    <option value="RJ"> Rio de Janeiro
  </option>
</optgroup>
</select>
```



No exemplo anterior, a sub-tag `<optgroup>` cria uma opção com finalidade de separar as opções da select. A opção criada, não pode ser clicada e tem uma formatação diferente das outras opções.

Área de Texto

Cria um campo do tipo memorando, permitindo a digitação de um texto com vários caracteres, conhecido como campo de texto multi linha.

```
<textarea cols="" rows="" wrap="on/off" readonly>
texto </textarea>
```

rows – número de linhas visíveis na área de texto.

cols – número de caracteres possíveis em uma linha.

wrap – define se a mudança de linha será automática durante a digitação (on), ou somente quando for pressionado ENTER (off). Se não for informado esse atributo, o mesmo assume on como default.

readonly – se esse atributo for usado, o texto contido no campo não poderá ser alterado. Utilizado para contratos.

Não existe o parâmetro value, o texto inicial deve ser definido entre as tags <textarea> e </textarea>.

Exemplo:

```
<textarea cols="50" rows="10" wrap="on/off"
readonly> Texto padrão </textarea>
```



Contadores de caracteres

O contador de caracteres é uma prática utilizada junto com a textarea. Para fazer um contador é preciso usar uma linguagem de programação chamada javascript. Abaixo um exemplo do código que faz a contagem dos caracteres.

Parte javascript:

```
<script language="Javascript">
function Contar(Campo)
{
    document.getElementById("Qtd").innerText = 250-
    Campo.value.length
    if((250-Campo.value.length)<0)
        alert('Atenção, você atingiu o limite máximo de
        caracteres!');
}
</script>
```

Parte HTML:

```
<textarea onkeyup="Contar(this)" ></textarea>
Você possui <span id="Qtd">250</span> caracteres.
```

Finalizando a parte de formulários temos um último atributo, o tabindex. Podemos utilizar a tecla TAB para pular entre os campos. Ele vai informar qual vai ser a sequencia de campos que o TAB vai seguir. Exemplo:

```
<form action=#>
Nome: <input type=text name=nome tabindex=1>
Idade: <input type=text name=idade tabindex=3>
Telefone: <input type=text name=telefone
tabindex=2>
</form>
```

No exemplo acima o foco está no campo nome, quando apertar a tecla TAB o foco vai para o campo telefone (tabindex=2), se apertar o TAB novamente ele vai para o campo idade (tabindex=3). Assim a ordem pode ser configurada conforme a necessidade do formulário.

Atividades

1) Montar os seguintes formulários:

a)

Sócio:	<input type="text"/>	Idade:	<input type="text"/>
Estado Civil:	<input type="radio"/> Solteiro <input type="radio"/> Casado <input type="radio"/> Divorciado	Cidade:	<input type="text" value="...: Selecione ..."/> ▼
		<input type="button" value="Limpar"/>	<input type="button" value="Enviar"/>

b)

Ficha de Cadastro de Pet's

Proprietário:	<input type="text"/>
Nome do Animal:	<input type="text"/>
Idade:	<input type="text"/>
Espécie:	<input type="text" value="...: Selecione ..."/> ▼
Sexo:	<input type="radio"/> Masculino <input type="radio"/> Feminino
<input type="button" value="Limpar"/> <input type="button" value="Enviar"/>	

c)

Mensageiro

Destinatário:

Assunto:

2) Montar o seguinte pedaço de formulário. Lembrando que ao clicar em cima do nome do esporte também deve marcar e desmarcar o checkbox.

Esportes

☐ Futebol ☐ Volei ☐ Basquete

3) Montar o select abaixo:

Homens
Barbearia
Clube do Bolinha
Mulheres
Salão
Shopping
Clube da Luluzinha

Projeto HTML 2: Música

Desenvolver um site com no mínimo 6 páginas sobre música. O site pode ser de um ou mais estilos musicais, pode ser de uma banda ou mesmo contando a história da música. Utilize todos os conhecimentos adquiridos até o momento. Não se esqueça de utilizar:

- vídeos
- sons
- tabelas
- imagens
- links
- traços
- listas
- formulários

Escolha um tema dentro da música que você goste, é mais fácil fazer trabalhos sobre o que gostamos. Esse é o nosso segundo projeto, já conseguimos ver uma evolução do primeiro para esse. Bons estudos!

CSS

O que é CSS

CSS é conhecido como folha de estilos, onde toda a formatação do site fica separada do conteúdo. O CSS pode ser interno (formatar somente uma página), externo (podendo formatar várias páginas ao mesmo tempo) ou inline (quando utilizada dentro do HTML para formatar pequenos pedaços).

CSS é o acrônimo de Cascading Style Sheet e seus arquivos são salvos com a extensão .css.

Lembre-se o CSS não insere nada na página, ele simplesmente formata os objetos já inseridos.

Uma dica muito importante é inserir na página CSS a linha que permite o CSS interpretar os caracteres brasileiros. Ela deve estar na primeira linha do arquivo.

Para isso coloque: @charset = "utf-8";

Vamos à sintaxe do CSS

```
seletor {  
  
    propriedade: valor;  
  
}
```

seletor – é onde a formatação vai ser aplicada

propriedade – qual o tipo de formatação utilizada (cor, tamanho)

valor – qual é o valor da propriedade (red, 15px, 100%)

Exemplo:

```
body{
  background-color:
  blue;
  color:red;
}
```

Como a formatação está sendo definida para o body, toda a página vai ficar com o fundo azul e com as letras em vermelho.

Existem 3 tipos de seletores: tag, class e id.

	Tag	class	Id
HTML	<code><body></code>	<code><td class=a1></code>	<code><input id=idade></code>
CSS	<code>body {</code> <code>}</code>	<code>.a1{</code> <code>}</code>	<code>#idade{</code> <code>}</code>

Seletores por tags são utilizadas quando todas aquelas tags vão possuir a mesma formatação. Se estiver formatando uma imagem (tag ``) todas as imagens do site assumiram a formatação.

Caso queira que cada imagem tenha uma formatação diferente, usa-se o class.

Exemplo:

```
<html>
<style type="text/css">
  .imagem1{
    width: 100px;
    height: 100px;
  }
  .imagem2{
```



```

        width: 200px;
        height: 200px;
    }
</style>
<body>
    <img src=image.jpg class=imagem1>
    <img src=image.jpg class=imagem2>
</body>
</html>

```

Tomar cuidado para não confundir o sinal de = com os :, pois No HTML usa-se o = antes de um valor e no CSS utiliza-se os :.

A diferença entre id e class é que um id pode ser definido apenas uma vez e para um objeto, não podendo ter mais de um objeto com o mesmo id. Já a class pode ser definida para um ou mais de um objeto. Mas lembre de que no CSS o id utiliza # antes no nome e a class utiliza um . antes do nome.

O seletor id é utilizado em formulários. Exemplo:

```

#idade {
    bgcolor: grey;
}

```

```

<input type=text name=idade id=idade>

```



Os tamanhos em CSS podem ser definidos em 3 medidas diferentes: pixels, porcentagem ou em.

Exemplo:

```
<style type="text/css">
  .tipo1 {font-size:14px;}
  .tipo2 {font-size:1em;}
  .tipo3 {font-size:120%;}
</style>
<p class=tipo1> Texto
</p>
<p class=tipo2> Texto
</p>
<p class=tipo3> Texto
</p>
```

O padrão de tamanho são 16px, isso quer dizer que 1em é igual a 16px e 100% também é igual a 16px. Pode-se colocar fracionado também. Exemplo:

```
<p style="font-size: 0.5em"> Texto </p>
```

O texto acima será formatado com um CSS inline e com tamanho de 0.5em ou 8px.

Comentário

No CSS também possuímos apenas comentários de bloco e com a mesma função dos comentários em HTML: procurar erros, deixar algum código ou informações comentados.

```
.p {
/* formatando o primeiro parágrafo */
  color: blue;
  font-size: 15px; /* formatar a fonte para o tamanho
de 15 pixels*/
}
```

Todos os navegadores possuem arquivos CSS para formatação das suas páginas. Essas formatações são as últimas na ordem da cascata do CSS, e são substituídas

por qualquer outra configuração que o programador possa fazer.

Como boa prática, usa-se o seletor `*` para configurações realizadas em toda a página. De todos os problemas causados pelo CSS do navegador, o que mais produz dor de cabeça são `margin` e `padding`. Resolve-se assim:

```
*{  
margin: 0;    Margem externa de um elemento  
padding: 0;   Margem interna de um elemento  
}
```

Assim essas duas propriedades estão zeradas em todos os elementos da página, obrigando o programador a setar elas nos elementos que ele quiser.

Um seletor pode ser declarado em mais de uma folha de estilo, com as mesmas propriedades e para o mesmo site. Existe uma ordem de relevância para as folhas de estilos. Sempre o que for declarado inline é o mais relevante. Abaixo segue uma figura explicativa.

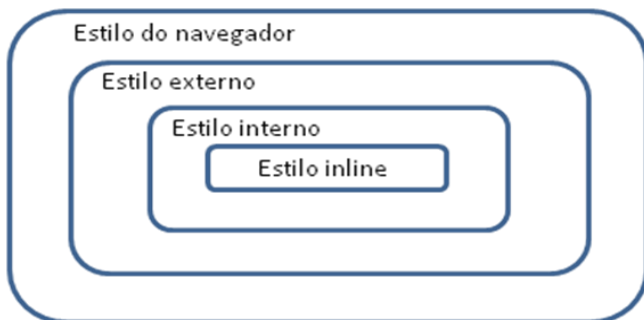
CSS Interno

O CSS interno formatará apenas os dados que estão na página. Ele fica em um local separado do HTML. Normalmente o CSS fica antes do `body`. Exemplo:

```
<html>  
  <style type="text/css">  
    body{  
      font-size: 15px;  
      color: blue;  
    }  
  </style>  
<body>
```

Eu sou a escrita da página!
Estou com tamanho 15 e cor azul.

Note que toda a página ficou com a font no tamanho 15px e na cor azul.



CSS Externo

O CSS externo é a prática mais utilizada e nada mais é que um arquivo que tem somente códigos CSS. Os códigos para o CSS interno ou para o CSS externo são os mesmos.

As páginas que vão utilizar o CSS externo precisam chamar o arquivo. Para isso utiliza-se a seguinte linha de comando:

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```

rel="stylesheet" indicando que o link é com uma folha de estilos

type="text/css" arquivo é de texto com sintaxe CSS

href="estilos.css" nome do arquivo com os estilos

Arquivo: estilo.css

```
body{
  font-size: 14px;
  color: red;
  background-color: yellow;
}
```

Arquivo: pagina.html

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="estilo.css">
  </head>
  <body>
    Eu vou aparecer formatado dentro da página!
  </body>
</html>
```

Note que no arquivo pagina.html não tem nenhuma formatação, e mesmo assim a página ficou com a fonte tamanho 14, cor vermelha e cor de fundo em amarelo. Para funcionar o exemplo acima, os dois arquivos devem estar no mesmo local (mesma pasta).

Agrupar seletores

Quando é preciso utilizar as mesmas propriedades com os mesmos valores para mais de um seletor, é possível agrupá-los. Exemplo:

Sem Agrupar

```
div{
    font-size: 20px;
    text-decoration:
underline;
}
p{
    font-size: 20px;
    text-decoration:
underline;
}
```

Agrupados

```
div, p{
    font-size: 20px;
    text-decoration:
underline;
}
```

HTML

<p> Texto dentro tag </p>
<div> Texto dentro tag </div>

Nos dois tipos o HTML não vai mudar podendo ser agrupados quantos seletores for necessário. Mas cuidado, reparem no exemplo abaixo:

```
div p{
    font-size: 20px;
    text-decoration: underline;
}
```

Ao colocar duas tags dessa forma como seletor (sem a vírgula entre os seletores), as configurações só vão funcionar quando a tag <p> estiver dentro da tag <div>. Traduzindo para um bom português ficaria assim: Só formate as tags <p> que estão dentro das tags <div>.

```
<div>
<p> Texto dentro das duas tags </p>
</div>
```

Formatação de Textos

Como no HTML, o CSS possui várias propriedades para ser utilizadas na formatação de textos. A seguir vemos as principais.

color:	red, blue, Black ou #000000 #ffffff	Pode ser informado o nome da cor ou seu equivalente em hexadecimal.
font-family:	arial, calibri	É o tipo da fonte.
font-size:	10px, 20px, 100px	O tamanho da fonte é definido por pixels, porcentagem ou em.
text-decoration:	none, underline, overline, line-through	Efeitos como sublinhado, riscado e sem efeito.
text-align:	center, right e justify	Alinhamento horizontal.
text-transform:	uppercase, lowercase e capitalize	Transforma o texto todo em maiúsculo, minúsculo ou capitular.
text-indent:	50px, 20px	Usado para dar um espaço no início do parágrafo
font-weight:	normal, bolder	Negrito (ou valores de 100 em 100 até 900)

```
<style type="text/css">
  .estilo1{ text-decoration: underline;
}
  .estilo2{ text-decoration: overline;
}
  .estilo3{ text-decoration: line-
through; }
</style>
```

underline

overline

~~line-through~~


```
<p class=estilo1> underline </p>
<p class=estilo2> overline </p>
<p class=estilo3> line-through </p>
```

Exemplo 2:

```
<style type="text/css">
  .estilo1{ text-transform: uppercase; font-size:20px;
text-indent: 10px;}
  .estilo2{ text-transform: lowercase; font-size:30px;
text-indent: 30px;}
  .estilo3{ text-transform: capitalize; font-size:40px;
text-indent: 50px;}
</style>
```

```
<p class=estilo1> pedro da silva</p>
<p class=estilo2> PEDRO DA SILVA</p>
<p class=estilo3> pedro da silva </p>
```

PEDRO DA SILVA

pedro da silva

Pedro Da Silva

Um efeito bem legal é colocar sombra nas palavras.
Para isso utiliza-se o comando CSS text-shadow;

Titulo

```
<p style=text-shadow: 2px 2px 2px black;> Titulo  
</p>
```

Explicando o código acima...

Foi criado um parágrafo e formatado com um CSS do tipo in-line (só para esse parágrafo). O text-shadow tem 4 valores, que na ordem, servem para:

- 1º Deslocamento horizontal da sombra
- 2º Deslocamento vertical da sombra
- 3º Espalhamento da sombra, podendo ficar mais difusa ou mais espalhada
- 4º Cor da sombra

Espaçamento entre linhas

É possível configurar o espaçamento entre linhas, palavras ou até letras usando o CSS. As três propriedades podem ser utilizadas com pixels ou porcentagem.

line-height:	Espaçamento entre linhas
letter-spacing:	Espaçamento entre letras ou caracteres
word-spacing:	Espaçamento entre palavras

Exemplo:

```
p{  
  line-height: 40px;  
  letter-spacing: 5px;  
  word-spacing: 10px;  
  width:50%;  
}
```

Sou um texto e estou me sentindo muito importante porque estou servindo de exemplo em uma atividade de um livro.

Formatação de Plano de Fundo

Agora com CSS vamos aprender a utilizar, da maneira certa, as imagens como plano de fundo. Como essas imagens são o plano de fundo de todo o site, então utiliza-se a tag <body> como seletor.

```
body {  
    background-image: url("imagens/fundo.jpg");  
    background-repeat: no-repeat;  
    background-position: 0px 40px;  
    background-attachment: fixed;  
}
```

background-image:	url("pasta/arquivo"); ou url("arquivo");	Indica onde o arquivo está qual o seu nome e extensão.
background-repeat:	no-repeat, repeat-x, repeat-y, repeat	A imagem de fundo pode não repetir, repetir em x (na horizontal) ou repetir em y (na vertical)
background-position:	0px 40px ou right top	O primeiro valor corresponde a horizontal e o segundo a vertical.
background-attachment:	fixed, scroll	Informa se a imagem de fundo vai se rolar junto com a página ou vai ficar fixa.
background-color:	red, blue, black ou #000000 #ffffff	As cores de fundo seguem o mesmo padrão, nome da cor ou seu hexadecimal.
background-size:	300px 400px	Configura a largura e a altura da imagem do background

A propriedade background-size não é muito utilizada, pois acarreta 2 problemas:

Se a imagem for muito pequena e for aumentar o tamanho dela com o background-size, a imagem vai ficar com uma qualidade ruim e até pixealizada.

Se ela for muito grande e for reduzida vai ser visualizada uma imagem pequena com o peso de uma imagem grande, acarretando lentidão e aumento de tráfego.

Atividades

- 1) Explique os tipos de CSS e a sua ordem de usabilidade (qual é o mais importante)?
- 2) O CSS trabalha com seletores, explique-os e exemplifique-os?
- 3) Montar um texto em uma página e formatá-lo com fonte Arial, tamanho 16px e cor vermelha.
- 4) Criar 2 parágrafos e formatá-los para que fiquem com tamanho e cores diferentes.
- 5) Usando as propriedades de parágrafos, crie o seguinte texto.

T e x t o b e m s e p a r a d o .

6) Montar uma página com um título sublinhado e alinhado ao centro, vários parágrafos formatados. Usar uma textura de fundo.

Formatação de Tabela

Utilizando o CSS podem ser realizadas várias formatações interessantes nas tabelas. O que tem que ser lembrado é que a tabela é criada no HTML e apenas formatada no CSS.

```
<table>
  <tr>
    <td> célula 1 </td>
    <td> célula 2 </td>
    <td> célula 3 </td>
  </tr>
</table>
```

célula 1 célula 2 célula 3

Agora o mesmo exemplo só que aplicado com o CSS abaixo:

```
table{
  border: 1px solid black;
}

table, td{
  border: 1px solid black;
}
```

célula 1 célula 2 célula 3

célula 1 célula 2 célula 3

Note que no primeiro exemplo, só a borda de fora da tabela ficou com as formatações. O CSS trata de forma

muito específica cada borda. No segundo exemplo foram usados 2 seletores: table e td. As formatações indicadas no bloco vão ser aplicadas a todos os seletores indicados.

Para trabalhar com formatação de tabelas é necessário saber:

border-color:	red, blue, black ou #000000 #ffffff	Pode ser informado o nome da cor ou seu equivalente em hexadecimal.
border-width:	1px, 5px	Espessura da borda
border-style:	solid, dotted, double e dashed	Tipo da borda
width:	30px, 40%	Largura da tabela, pode ser usado pixels ou porcentagem
height:	30px, 40%	Altura da tabela, pode ser usado pixels ou porcentagem
background-color:	red, blue, black ou #000000 #ffffff	Pode ser informado o nome da cor ou seu equivalente em hexadecimal.
border-collapse:	separate, collapse	O padrão é o separate. O collapse elimina o espaço entre uma borda e outra.

Note que em CSS podemos usar uma propriedade mais genérica com vários valores que são semelhantes.
Exemplo:

```
table{
  border-width: 1px;
  border-color:red;
  border-style: solid;
}
```

```
table{
  border: 1px red solid;
}
```

Os dois códigos executam a mesma coisa. Como as três propriedades são da borda, pode-se unir em apenas uma linha.

Outro exemplo:

```
body{
  font-size: 30px;
  font-family: arial;
}
```

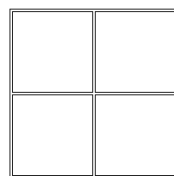
```
body{
  font: 30px arial ;
}
```

Espaço entre as bordas

Colocando borda no table e no td cada linha da borda ficará dupla. Repare no exemplo a seguir:

```
table{
  border: 1px solid black;
  width: 200px;
  height: 200px;
}
```

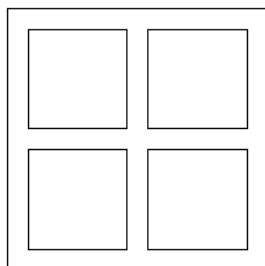
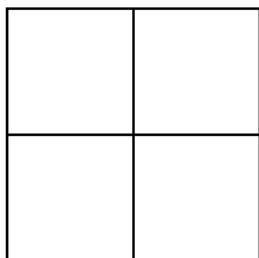
```
td{
  border: 1px solid black;
}
```



O espaço entre as duas linhas pode ser controlado com a propriedade border-spacing. Quanto maior for o valor, maior será o espaço entre as duas linhas.

```
border-spacing: 0px;
```

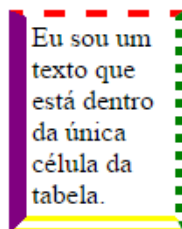
```
border-spacing: 15px;
```



Extra:

Para demonstrar que em CSS se tem o total controle dos recursos de uma página, analisamos o seguinte exemplo:

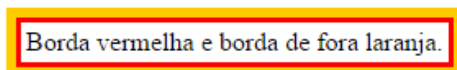
```
table{
  width:100px;
  border-top: red dashed 4px;
  border-right: green dotted 5px;
  border-left: purple solid 10px;
  border-bottom: yellow double 10px;
}
```



No exemplo acima a tabela possui formatação diferente para todas as suas bordas. Em CSS é possível controlar cada borda da tabela separadamente.

Outra formatação interessante em uma tabela é a sua “borda de fora”.

```
table{
  border: 3px red solid;
  outline: 6px solid #FFCC00;
}
```



No exemplo acima a borda está com espessura 3px, sólida e na cor vermelha e a borda de fora está com espessura 6px, sólida e na cor laranja. As propriedades são as mesmas para border ou outline (width, color e style).

Para centralizar uma tabela na página utiliza-se as margens direita e esquerda como automáticas.

```
table{  
    border: 3px solid;  
    width: 300px;  
    height: 50px;  
    margin-left: auto;  
    margin-right: auto;  
}
```



Em vez de programar a margin-left: auto e depois a margin-right: auto podemos abreviar a propriedade margin. Basta colocar margin: auto.

Alinhamento Vertical

Um recurso bastante utilizado também em tabelas ou DIVs (conteúdo mais para baixo) são os alinhamentos verticais.

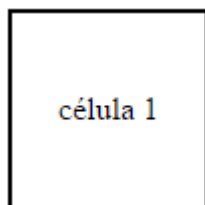
```
.td1{ vertical-align: top; }  
.td2{ vertical-align: middle; }  
.td3{ vertical-align: bottom; }
```



Arredondando cantos da tabela

Um efeito interessante é o arredondamento de cantos. Pode ser definido um valor para os 4 cantos ou o CSS trata cada canto separadamente. Exemplo:

```
table{  
  border: 2px solid black;  
  width: 100px;  
  height: 100px;  
  text-align: center;  
}
```



```
table{  
  border: 2px solid black;  
  width: 100px;  
  height: 100px;  
  text-align: center;  
  border-radius: 20px;  
}
```



A diferença entre os 2 códigos é que na segunda tabela temos um `border-radius: 20px` inserindo assim um arredondamento de 20px em todos os cantos da tabela.

Como já disse, podemos trabalhar com qualquer uma das bordas separadamente. No exemplo abaixo, vamos mexer somente nas bordas inferiores.

```
table{  
  border: 2px solid black;  
  width: 100px;  
  height: 100px;  
  text-align: center;  
  border-radius: 0px 0px 30px 30px;  
}
```



Os 4 valores seguem a direção do ponteiro do relógio. Explicando...

Border-radius: esquerda de cima, direita de cima, direita de baixo, esquerda de baixo.

Atividades

1) Montar uma tabela com 2 células na mesma linha, borda simples, 100px de largura e 150px de altura.

2) Montar uma tabela com 2 células e seguir o padrão:

Célula 1:

Cor de fundo verde
Fonte amarela
Tamanho 100px de largura e 50px de altura
Tamanho da fonte 20px

Célula 2:

Cor de fundo azul
Fonte branca
Tamanho 300px de largura e 50px de altura
Tamanho da fonte 30px

3) Montar a seguinte tabela e fazer as formatações conforme a tabela abaixo:

Célula 1	Célula 2	Célula 3
Célula 4	Célula 5	Célula 6

Células 1, 2 e 6	Células 3 e 4	Célula 5
Fundo verde	Fundo Amarelo	Fundo vermelho
Alinhado a direita	Alinhado ao centro	Alinhado a esquerda
Tamanho 18px	Tamanho 30px	Tamanho 50px
Cor azul	Cor verde	Cor branca

4) Montar 1 tabela e formatar da maneira que achar. Mudar as bordas, as cores e os tamanhos.

5) Montar as seguintes tabelas:

Eu sou um texto aleatório utilizado para montar o exercício. Me sinto orgulhoso disso.

Eu sou um texto aleatório utilizado para montar o exercício. Me sinto orgulhoso disso.

Pseudo-Classe

Uma pseudo-classe controla o estado de um elemento. Abaixo segue as sintaxes possíveis.

seletor:pseudo-classe ou seletor.class:pseudo-classe

Um elemento pode ter mais de um estado: como exemplo temos um parágrafo formatado com o tamanho da fonte em 20px. Se posicionarmos o mouse em cima do texto, ele vai mudar o tamanho da fonte para 30px. Exemplo:

```
<style>
p {
    font-size:20px;      Formatação normal
}
p:hover{
    font-size:30px;      Formatação com pseudo-classe
}
</style>

<body>
  <p> Eu sou o texto que vai ficar com 30px quando o mouse se
    posicionar em cima de mim. </p>
</body>
```

Como um ótimo exemplo temos a formatação dos links que tem 4 pseudos-classes: active, visited, link e hover. Confira logo abaixo.

Formatação de Link

Os links possuem 4 estados: sem visitação, visitados, ativos e mouse em cima. Para definir um estágio utiliza-se as pseudos-classes. Exemplo:

<code>a:link { color: red; }</code>	Os links não visitados serão vermelhos
<code>a:visited { color: green; }</code>	Os links visitados serão verdes
<code>a:hover { color: pink; }</code>	Os links com o mouse em cima serão rosa
<code>a:active { color: blue; }</code>	Os links ativos serão azuis

CSS

```
<style type="text/css">
a{
  background-color: grey;
  padding: 10px;
  margin: 20px;
  color: black;
}
</style>
```

HTML

```
<body>
  <a href="pag1.html">Página 1</a>
  <a href="pag2.html">Página 2</a>
  <a href="pag3.html">Página 3</a>
</body>
```

Página 1

Página 2

Página 3

O padding é o espaço interno entre a escrita e a borda do quadrado e margin é o espaço entre os quadrados. Note no exemplo acima que entre o primeiro e o segundo quadrado temos 40px de margin: 20px direita

do primeiro quadrado mais 20px esquerda do segundo quadrado.

Formatando Imagens

A imagem deve ser inserida pelo HTML e o CSS entra com todas as formatações necessárias. No exemplo abaixo foi definido o tamanho de 120px tanto para altura (height) quanto para largura (width). A propriedade float serve para alinhar a imagem em relação aos outros objetos. No exemplo abaixo, float: left alinhou a imagem a esquerda do texto. Exemplo:

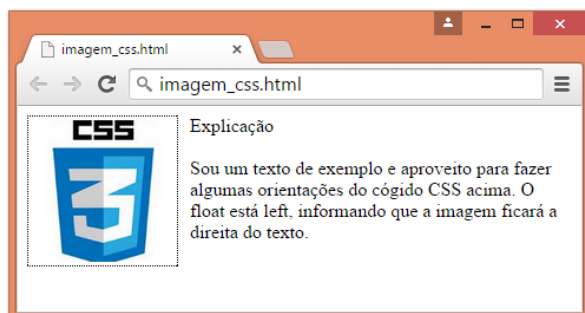
CSS

```
<style type="text/css">
img{
  float: left;
  width: 120px;
  height: 120px;
  border: thin dotted;
  padding: 3px;
  margin: 0px 10px 10px
0px;
}
</style>
```

HTML

```
<body>

Explicação <br><br>Sou
um texto de exemplo e
aproveito para fazer
algumas orientações do
código CSS acima. O float
está left, informando que a
imagem ficará a esquerda
do texto.
</body>
```



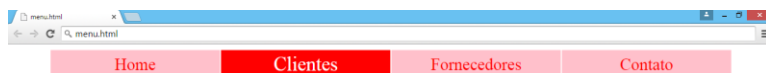
Completando a explicação com a propriedade margin: 0px 10px 10px 0px onde a sequencia é: margin: em cima, direita, em baixo, esquerda.

Imagens transparentes - Alguns efeitos são interpretados de maneiras diferentes em determinados navegadores. Por isso o responsável pela formatação do site deve estar atento e testar o site em diversos navegadores. A parte boa é que todos tem suporte aos efeitos, mas usam códigos diferentes:

```
.img_transparente {  
    -moz-opacity:0.65;           Navegador Mozilla (0 até 1)  
    filter: alpha(opacity=65);   Navegador IE (0 até 100)  
}
```

Atividades

1) Montar uma tabela com uma linha e quatro células. Formata-la usando somente CSS conforme a imagem abaixo:



Quando o mouse for em cima dos menus, ele deve ficar com a cor de fundo em vermelho, letras escritas em branco e um pouco maior que a original.

Obs: Utilizamos essa pseudo-classe quando estudamos os links.

Posicionando Imagens (e outros elementos)

Para posicionar elementos na tela utiliza-se uma propriedade muito importante chamada position. Ela pode ter 3 valores possíveis: absolute, relative e fixed.

Fixed mantém a DIV fixa no local especificado.

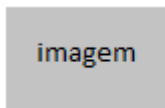
Os dois que causam maiores dificuldades são o relative e o absolute.

Relativo toma como ponto de partida o local onde o elemento esta. Não sobrepondo os outros elementos.

Absoluto toma como ponto de partida o canto da tela e sobrepõe os outros elementos.

Exemplo 1

```
.img1{  
  position:absolute;  
}  
.img2{  
  position:absolute;  
}
```



Exemplo 2

```
.img1{  
  position:relative;  
}  
.img2{  
  position:relative;  
}
```



Note que no primeiro exemplo temos position: absolute nas duas imagens. Como elas tem o mesmo tamanho e a mesma posição, só está aparecendo uma, pois a outra

está em baixo. O absolute não leva em conta outros elementos do site.

No segundo exemplo temos position:relative nas duas imagens. O relative quer dizer que elas vão se posicionar uma em relação à outra. Por isso aparecem as duas imagens.

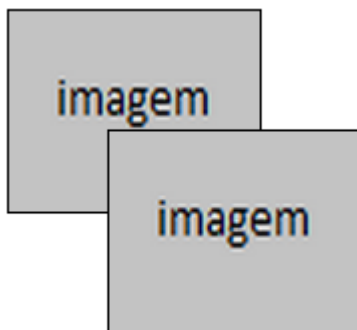
Exemplo:

CSS

```
img{  
  width: 125px;  
  height: 100px;  
  border: 1px solid;  
}  
  
.img1{  
  position:absolute;  
  margin: 0px 0px 0px  
  0px;  
}  
  
.img2{  
  position:relative;  
  margin: 60px 0px 0px  
  50px;  
}
```

HTML

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



Explicando o Exemplo:

Temos 2 imagens e como as duas tem o mesmo tamanho, formatamos a altura, a largura e a borda com o seletor img.

A primeira imagem foi fixada com absolute com todas as margens em 0px. Ela ficará absolutamente na posição informada pela margin.

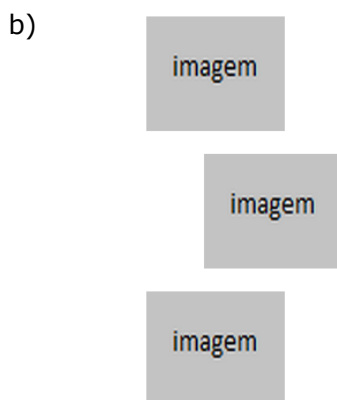
A segunda imagem está com position:relative, e está relativo a outros elementos relativos do documento HTML.

Atividades

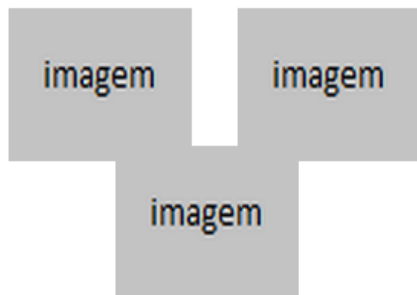
1) Fazer 4 links em uma página e formatá-los da seguinte maneira: os visitados em rosa, os não visitados em verde, os ativos em roxo e os com o mouse em cima amarelo.

2) Montar uma página com links internos. Ao clicar no link na parte de cima da página, o site deve ser redirecionado a outra parte da mesma página. Usar texto e imagens. Mas não esqueça tudo formatado em CSS.

3) Fazer as seguintes montagens com imagens. Para isso usar 3 imagens mais ou menos do mesmo tamanho.



c)



4) Fazer uma página para mostrar fotos como se fosse um álbum de aniversário. Utilizar 10 imagens distribuídas de maneira aleatória ou como achar melhor, e utilizar o a pseudo-classe hover nas imagens. Não se esqueça do título do álbum e do plano de fundo.


5) Inserir uma imagem como plano de fundo. Coloca-lá centralizada e com transparência de 50%.

Formatando Formulários

Os formulários são os principais meios de interação do usuário com o site ou sistemas. Sistemas? Isso mesmo, várias linguagens de programação para internet utilizam os formulários HTML.

Pode-se fazer efeitos bem interessantes com o CSS. Abaixo exemplifico alguns efeitos com campos do tipo input.

```
input{  
  background:grey;  
  height: 30px;  
  color: white;  
  font-size: 20px;  
}
```



Joaquim da Silva

No exemplo acima, mudamos 4 propriedades que não seriam possíveis mudar somente com HTML.

Os formulários tem uma pseudo-classe específica: focus. Essa propriedade é executada quando o campo recebe o focus, ou para um bom português, quando o cursor fica piscando dentro do campo.

```
input:focus  
{  
  background:black;  
}
```



No exemplo acima, quando o campo input receber o foco ele ficará com o fundo preto. As duas programações tem que estar no CSS. Note que usado a tag input como seletor temos um problema, pois a

maioria dos elementos do formulário utilizam a tag `input`.

Problema:

Usando o mesmo código CSS acima, mas adicionando um botão no HTML as configurações do seletor `input` vão ser aplicadas ao botão também.



Poderia criar uma class para os elementos do formulário, isso resolveria. Mas para a formatação de formulários temos algo especial, como no exemplo:

```
input[type="text"]{  
    background:grey;  
    height: 30px;  
    color: white;  
    font-size: 20px;  
}
```



Observe que no seletor do CSS é informado em qual tipo de campo `input` deve ser aplicado as formatações. Outros exemplos:

```
input[type="radio"]  
input[type="submit"]  
input[type="select"]  
input[type="checkbox"]
```

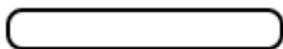
Lembrando que esses seletores podem ser combinados com as pseudo-classes. Exemplo:

```
input[type="submit"]:focus {  
  
}
```

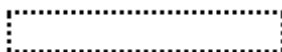
As formatações acima vão ser aplicadas em campos input, do tipo submit e ainda quando o foco estiver no campo.

Uma formatação interessante para fazer nos campos do formulário é colocar uma borda arredondada ou utilizar bordas (escolhendo o tipo e a espessura como nas tabelas): Exemplo:

```
input[type="text"]{  
  border: 2px solid;  
  width:100px;  
  border-radius: 8px;  
}
```



```
input[type="text"]{  
  border: 2px dotted;  
  width:100px;  
}
```



Uma dica é aplicar a mesma formatação para diferentes elementos no formulário. O uso mais comum é os campos do tipo text e password. Para aplicar a mesma formatação nos dois seletores é só separa-los por uma vírgula. Exemplo:

```
input[type="text"], input[type="password"]  
{  
  background-color: grey;  
}
```

Atividades

1) Para as caixas de texto utilizar a cor cinza claro (#CCCCCC) e para os botões cinza escuro (#333333).

Cadastro de Alunos

Nome:

Escola:

Sexo: ☐ Masculino ☐ Feminino

2) Utilizando o mesmo HTML do exercício acima, formata-lo da seguinte maneira (caixa de texto cor FF99FF, tabela cor 663366 e botões cor 330033):

Cadastro de Alunos

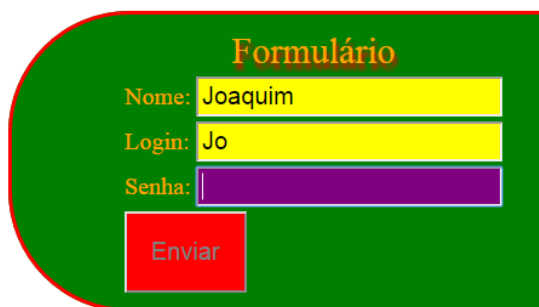
Nome:

Escola:

Sexo: ☐ Masculino ☐ Feminino

Projeto CSS 1: Formulário de Login

Montar um formulário e formatá-lo da seguinte maneira:



Formulário

Nome: Joaquim

Login: Jo

Senha:

Enviar

Preste atenção que na palavra **Formulário** tem uma sombra. Os campos nome, login e senha alteram seu background para roxo e a cor da fonte para branca quando o cursor estiver dentro deles.

Criando Layout com DIVs

Quando estamos aprendendo HTML é muito normal utilizar as tabelas para montar os layouts dos sites. Esse recurso, embora funcional e rápido, não é utilizado na criação de layouts. Para esse fim, utilizam-se blocos formatados em CSS. O mais utilizado e famoso são as DIVs.

O DIV também é chamado de elemento de Container e é um elemento HTML que não tem nenhuma

propriedade aparente, sendo um elemento neutro. Por isso ele é utilizado para montar os blocos do layout do site.

Vamos à prática!

```
<style type="text/css">
div{
  width: 500px;
  height: 150px;
  background-color: grey;
}
</style>
```

```
<body>
  <div>
    Sou um texto e estou dentro da div.
  </div>
</body>
```



Foi montado um bloco simples com um texto dentro. Note que o HTML é muito simples e toda a formatação está no CSS.

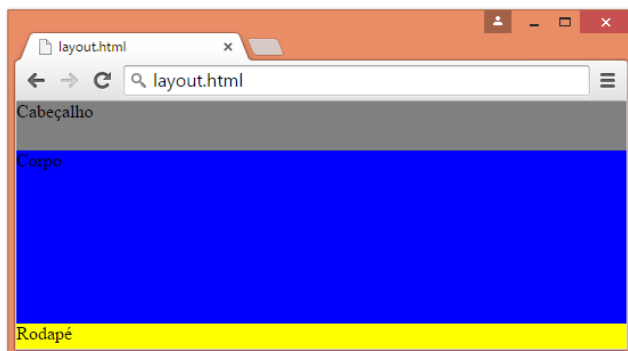
Exemplo 2:

CSS	HTML
<pre>*{ margin:0px; padding:0px; }</pre>	<pre><body></pre>
<pre>div.cabecalho{ width: 100%; height: 20%; background-color: grey; }</pre>	<pre><div class=cabecalho> Cabeçalho </div></pre>
<pre>div.corpo{</pre>	<pre><div class=corpo> Corpo</pre>

```

width: 100%;
height: 70%;
background-color: blue;
}
div.rodape{
width: 100%;
height: 10%;
background-color: yellow;
}
</div>
<div class=rodape>
Rodapé
</div>
</body>

```



No exemplo acima o layout foi desenvolvido com 3 DIVs uma embaixo da outra. E seus conteúdos não se misturam.

Tipos de Posicionamentos

Existe 3 tipos de posicionamento de DIVs: relative, absolute e fixed. O padrão é a div ser posicionada como relative.

A posição relative fica em relação aos outros objetos inseridos na página HTML, não podendo ficar sobreposto por nenhum deles.

A posição absoluta fica onde as propriedades top e left no CSS mandarem. Não importando se vão ficar em cima de algum outro objeto. Quando houver a rolagem da página, tanto a absolute quanto a relative rolam junto com a página.

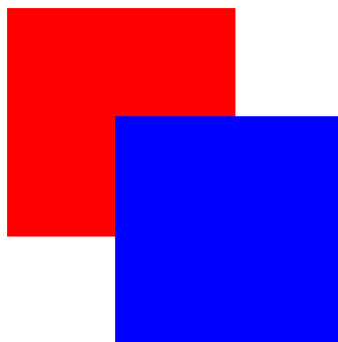
A posição fixed é como a absoluta, mas quando possui rolagem de página, a mesma não acompanha a rolagem, permanecendo fixa no local programado.

CSS

```
div#cx1{
  width: 300px;
  height: 300px;
  background-color: red;
  position:fixed;
}
div#cx2{
  width: 300px;
  height: 300px;
  background-color: blue;
  position:absolute;
  left: 150px;
  top: 150px;
}
```

HTML

```
<body>
  <div id=cx1> </div>
  <div id=cx2> </div>
</body>
```



Para fazer um teste insira um texto grande logo após a segunda div no HTML até que a sua página fique com uma barra de rolagem. Ao descer a página, note que a

div vermelha não sai da posição dela, ficando até em cima do texto.

Relative X Absolute

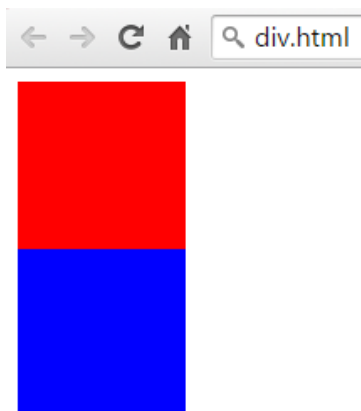
Como já explicado a position relative não sobrepõe os outros elementos. A position absolute não respeita os outros elementos e sim a posição da margin definida.

Para as explicações abaixo vou utilizar o seguinte HTML.

```
<body>
  <div id=cx1> </div>
  <div id=cx2> </div>
</body>
```

Exemplo 1: Duas DIVs relatives

```
<html>
<style type="text/css">
div#cx1{
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  top: 0px;
}
div#cx2{
  width: 100px;
  height: 100px;
  background-color: blue;
  position: relative;
```



```
    top: 0px;
}
</style>
```

Note que a div#cx2 esta com top: 0px, mas não está a 0px da página. Ele está relative ao outro elemento, estando assim com top: 0px da div#cx1.

Exemplo 2: Uma DIV relative e outra absolute

```
<html>
```

```
  <style type="text/css">
```

```
    div#cx1{
```

```
      width: 100px;
```

```
      height: 100px;
```

```
      background-color: red;
```

```
      position: relative;
```

```
      top: 0px;
```

```
    }
```

```
    div#cx2{
```

```
      width: 100px;
```

```
      height: 100px;
```

```
      background-color: blue;
```

```
      position: absolute;
```

```
      top: 0px;
```

```
    }
```

```
  </style>
```

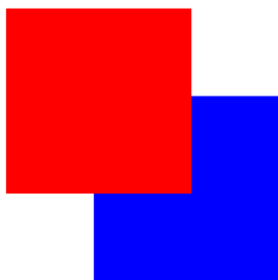


Note que a div#cx2 ficou em cima da div#cx1, não respeitando a outra DIV. a div#cx2 ficou mais para cima ainda porque a div#cx1 está respeitando a margin da página.

Ordenação das DIVs

Note que a div cx2 (azul) está em cima da div cx1 (vermelha). Isso ocorre porque a div cx1 está criada antes que a div cx2 no HTML. Usando o CSS da para controlar isso, sem precisar alterar as ordens no HTML. Utiliza-se a propriedade z-index para ordenar objetos na página.

A z-index é a ordem em relação ao eixo z. As DIVs que tem o z-index maior sobrepõem as DIVs com z-index menor.



Como exemplo, vamos adicionar a propriedade z-index no CSS criado para o exemplo acima. Na div#cx1 adicione z-index:2 e na div#cx2 adicione z-index:1. Visualizando...

Exemplo 3:

CSS

```
*{ margin:0px; padding:0px; }
div.menu{
  width: 10%;
  height: 100%;
  background-color: grey;
  position:absolute;
}
div.corpo{
  width: 90%;
  height: 90%;
  background-color: blue;
```

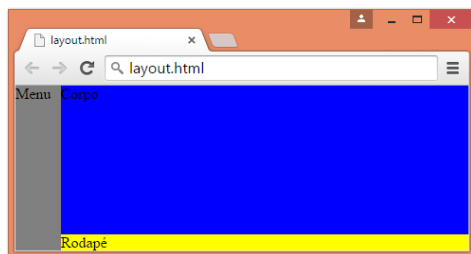
HTML

```
<body>
  <div class=menu>
    Menu
  </div>
  <div class=corpo>
    Corpo
  </div>
```

```

    position:relative;
    margin-left: 10%;
}
div.rodape{
    width: 90%;
    height: 10%;
    background-color: yellow;
    position:relative;
    margin-left: 10%;
}
<div class=rodape>
    Rodapé
</div>
</body>

```



Exemplificando DIVs – relative e absolute

Para as explicações abaixo, utilizamos a seguinte estrutura HTML.

```

<body>
  <div id=cx1>
    <div id=cx2>

      </div>
    </div>
  </body>

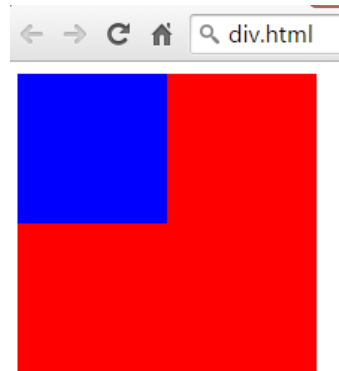
```

Note que foi criado duas divs e uma está dentro da outra.

Exemplo 1: Se as duas divs não possuem posicionamento, vão estar alinhadas a esquerda na ordem em que foram criadas. A azul está dentro da vermelha.

```
div#cx1{
  width: 200px;
  height: 200px;
  background-color: red;
}

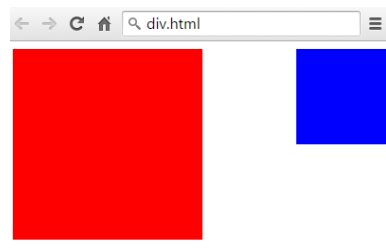
div#cx2{
  width: 100px;
  height: 100px;
  background-color: blue;
}
```



Exemplo 2: A div azul com posicionamento absoluto e a div vermelha sem posicionamento. Como a div vermelha não tem posicionamento, a azul vai ficar 0px da esquerda da página.

```
div#cx1{
  width: 200px;
  height: 200px;
  background-color: red;
}

div#cx2{
  width: 100px;
  height: 100px;
  background-color: blue;
  position: absolute;
}
```

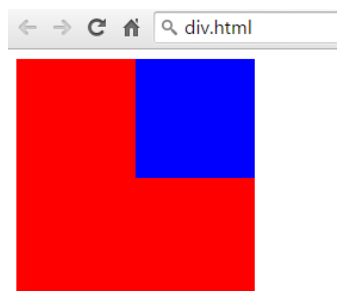


```
    right: 0px;  
}
```

Exemplo 3: A div azul com posicionamento absoluto e a div vermelha está com o position relative. Como a div vermelha tem um posicionamento (pode ser absolute ou relative), a azul vai ficar 0px da esquerda da div vermelha.

```
div#cx1{  
    width: 200px;  
    height: 200px;  
    background-color: red;  
    position: relative;  
}
```

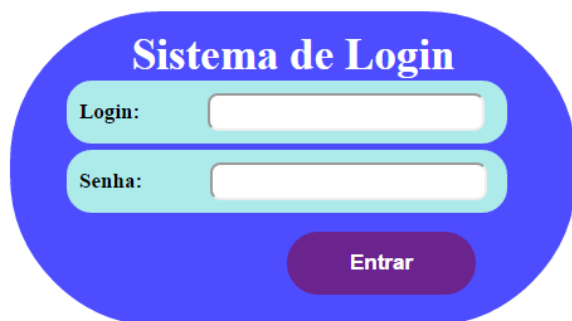
```
div#cx2{  
    width: 100px;  
    height: 100px;  
    background-color: blue;  
    position: absolute;  
    right: 0px;  
}
```



Atividades

1) Usando o código HTML abaixo, formatar o formulário para que ele fique com a seguinte aparência:

```
<body>
  <div class=principal>
    <div class=titulo>Sistema de Login </div>
    <div class=formulario> Login: <input type=text> </div>
    <div class=formulario> Senha: <input type=text> </div>
    <input type=submit value="Entrar">
  </div>
</body>
```

A visual representation of the login form. It features a blue rounded rectangle background. At the top, the text "Sistema de Login" is written in white. Below this, there are two light blue rounded rectangles for input fields. The first is labeled "Login:" and the second is labeled "Senha:". Both labels are in white. Below the input fields, there is a dark blue rounded rectangle button with the word "Entrar" in white.

Position Fixed

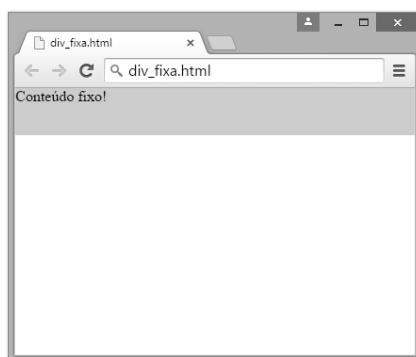
Um recurso interessante é fixar uma DIV no site. Um exemplo de utilização é montar uma DIV para colocar o menu e deixa-la fixa na parte superior do site.

CSS

```
div {  
    position:fixed;  
    width:100%;  
    height:50px;  
    background:#CCC;  
}
```

HTML

```
<body>  
    <div>Conteúdo fixo!</div>  
    Aqui vem o restante do site!  
</body>
```



Quando o site estiver com o seu conteúdo e gerar uma barra de rolagem, a DIV vai ficar sempre visível e fixa no topo do site.

Utilizando esse exemplo vamos conhecer uma nova forma de atribuir tamanho para uma DIV. Vimos que

podemos definir o tamanho de uma DIV em porcentagem ou em pixels.

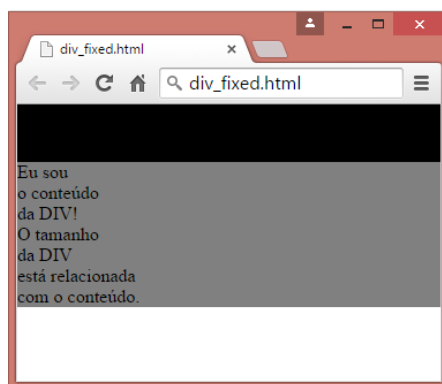
CSS

```
div#cabecalho{  
  width: 100%;  
  height: 50px;  
  background-color: black;  
  position: fixed;  
  top: 0px;  
  z-index:2;  
}
```

```
div#corpo{  
  width: 100%;  
  height: auto;  
  background-color: grey;  
  top: 50;  
  position: absolute;  
  z-index:1;  
}
```

HTML

```
<div id=cabecalho> </div>  
  <div id=corpo>  
    Conteúdo  
  </div>
```



Explicação do exemplo acima:

Temos a `div#cabecalho` fixa na parte superior do site e a `div#corpo` absoluta em baixo. O tamanho da `div#corpo` vai depender do tamanho do seu conteúdo (`height: auto;`). Caso o seu conteúdo gere uma barra de rolagem, a `div#cabecalho` sempre vai aparecer na tela, pois ela está fixa e com o `z-index` maior que a `div#corpo`.

Projeto CSS 2: Layout de Formulário

Cadastro de Pet's

Dados Pessoais

Nome:

E-mail:

Telefone:

Dados do Pet

Nome:

Tipo:

Observação:



Rua das Andorinhas, 740
Curitiba / Paraná

Enviar Dados

Limpar Dados



Livro**Rápido**

Nunca foi tão fácil publicar!!!

www.livrorapido.com

Rua Dr. João Tavares de Moura, 57/99 - Peixinhos Olinda-PE CEP: 53230-290
Fone (81) 2121.5307/2121.5313 e-mail: livrorapido@webologica.com