

Curso de

HTML



Este material é exclusivo do canal (youtube): [canalfessorbruno](https://www.youtube.com/c/canalfessorbruno) e do site www.cfbcursos.com.br não pode ser distribuído ou comercializado por outra fonte.



Apostila: versão 1.0
29/11/2015

www.youtube.com/canalfessorbruno

www.cfbcursos.com.br

canalfessorbruno@gmail.com

www.facebook.com/canalfessorbruno

twitter: @fessorBruno



Sumário

Introdução.....	5
Iniciando HTML5, código básico	5
Outras tags importantes	6
Outras meta tags importantes “keywords e description”	6
Anexando CSS e Javascript externos.....	6
Adicionando ícone à página “favicon”	6
Novas tag importantes de HTML5, de olho na nova semântica	7
section.....	7
nav.....	7
article	7
aside.....	7
header	7
footer	7
time	7
mark.....	8
Os novos elementos de HTML5	8
Criando nossa primeira página em HTML5.....	11
Iniciando em CSS.....	14
Anexando folha de estilos externas.....	14
Propriedades margin, width e background-color	14
A propriedade padding e border-radius para arredondar os cantos.....	15
Entendendo uso das propriedades float e clear	18
A propriedade border	21
color e text-align	22
font-size	22
Segunda página em HTML5	23
Criando um menu horizontal	24
Centralizando a página e configurando fundo do body.....	27
<section> e <aside>.....	29
<footer>	31
Criando um slider básico em javascript	32
Novos elementos de formulário	37
tel	37
search.....	37
email.....	38



url	38
Datas e Horas	38
number	40
range	41
color	42
Criando o formulário para nosso site “Escola de Inglês”	42
Pattern para validação dos formulários	47
Detalhes e Sumário <details> <summary>	48
Transparência	52
Gradiente	53
linear-gradient (variação linear)	53
radial-gradient (variação circular)	57
Repetição de gradientes	58
Áudio e Vídeo	59
SVG – Gráficos vetoriais direto pelo browser	60
Conteúdo editável	69
Revisão ortográfica	69
MathML – Criando equações	69
Variável	70
Números	70
Operadores	70
Exponenciação / Superscripts	71
Agrupar expressões sup	71
Raizes	72
Frações	72
Juntando tudo	73
meter	73
progress	73
Terceira página em HTML5	74
Resumo das tags HTML	81
Novas tags HTML 5	81
Tags já existentes em HTML 4 que foram continuadas	82
Tags descontinuadas, não devem ser usadas	84



Introdução

HTML 5 é a mais nova versão da linguagem padrão para desenvolvimento Web, uma atualização necessária para o HTML a versão 5 traz novidades interessantes, uma nova semântica de desenvolvimento e novas tags, em conjunto com novos comandos para formatação CSS as possibilidades de design de um site aumentam bastante.

A maioria das tags que usamos em HTML4 continuam em operação, algumas tags foram descontinuadas outras foram reformuladas e muitas foram mantidas exatamente como são, portanto, todo conhecimento adquirido em desenvolvimento HTML4 será aproveitado em HTML5.

Neste curso não iremos focar em HTML4, então é importante que você tenha esse conhecimento prévio para que o aprendizado seja mais fluido, se você não sabe HTML é recomendado que você assista todas as aulas do curso de HTML do CanalFessorBruno no link "www.youtube.com/canalfessorbruno".

Vamos iniciar nosso curso de HTML5 e aprender esse novo conceito para desenvolvimento Web.

Iniciando HTML5, código básico

Vamos começar do começo, nosso primeiro assunto sobre HTML5 é sobre o novo padrão do código básico da página, a seguir vamos ver o código que forma o esqueleto de uma página HTML5, o código essencial, básico, sempre iremos começar uma nova página com este código.

```
<!doctype html>
<html lang="pt-br">
  <head>
    <title>Título da página</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <!-- Conteúdo da página -->
  </body>
</html>
```

Vamos descrever as principais mudanças no código básico, doctype, meta charset e html.

`<!doctype html>` = O "doctype" deve ser a primeira tag a ser inserida, através desta apresentação do "doctype" podemos identificar que a página é desenvolvida em HTML5. O "doctype" indica ao browser qual especificação de código usar, neste caso HTML5.

`<html lang="pt-br">` = Marca o início do código HTML e já informa qual a linguagem principal do documento.

`<meta charset="UTF-8">` = Esta meta define qual página de código será usada na página, veja que ficou muito mais simples em relação à HTML4.

HTML4 charset = `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`

HTML5 charset = `<meta charset="UTF-8">`

Este é o código básico principal, toda página HTML é iniciada e estruturada neste código, lembre-se, ao iniciar uma nova página o primeiro trabalho a ser feito é inserir este código básico.

Outras tags importantes

A seguir vamos ver outras tags importantes em uma página HTML que irão complementar a construção de um site, não são tags que irão influenciar no visual da página, mas são comandos importantes que irão afetar na organização e configuração página.

Outras meta tags importantes “keywords e description”

Outras duas meta tags importantes em uma página são “keywords” e “description”, são usadas para indexar a página em sites de busca.

keywords = Configura as palavras-chave para que sua página seja mais fácil ser encontrada de acordo com o assunto pesquisado.

description = Contém uma breve descrição sobre o conteúdo do site.

Veja a seguir como usar essas meta tags.

```
<meta name="keywords" content="curso,tutorial,apostila,livro,html">
<meta name="description" content="Curso de HTML 5">
```

Anexando CSS e Javascript externos

É muito importante aprendermos a trabalhar com arquivos de formatação CSS e arquivos com código javascript externos, isso dá uma flexibilidade e portabilidade enorme da formatação e scripts entre páginas.

Trabalhar com arquivos CSS e Javascript externos facilita até mesmo na manutenção do código concentrando todo CSS por exemplo em uma mesma página, além de deixar a página HTML mais “limpa/pura”.

A seguir veja os comandos necessários para anexarmos página CSS e Javascript em página HTML, note que estes comandos são usados na tag <head>

```
<head>
<link rel= "stylesheet" href= "estilo.css">
<script src= "javascript.js"></script>
</head>
```

Adicionando ícone à página “favicon”

Favicon é um ícone que é adicionado à página, representam as página adicionada no favoritos do browser e são mostradas na barra de título ao abrir o site, podemos usar imagens em formato ico, png e jpg por exemplo e quanto ao tamanho antigamente se usava imagens de “16px X 16px” mas hoje os browsers e sistemas já usam imagens maiores como mostra a lista a seguir.

- **16px:** Barra de endereço, abas e favoritos usado na maioria dos navegadores.
- **24px:** Barra de ferramentas do I.E.9
- **32px:** Nova aba no Internet Explorer, site fixo na barra de tarefas do Windows e sidebar do Safari
- **57px:** iPhone sem tela retina e iPod Touch
- **72px:** iPad sem tela retina
- **96px:** GoogleTV
- **114px:** iPhone com tela retina
- **128px:** Chrome Web Store
- **144px:** iPad Retina, Windows 8 Tiles
- **160px:** Opera Speed Dial
- **256px:** Opera Speed Dial



A seguir o código para inserir o ícone na página, como as outras tags link, note que é uma tag “link”, então deve ser inserida dentro da tag <head>.

```
<head>  
<link rel="shortcut icon" href="favicon.ico">  
</head>
```

Novas tag importantes de HTML5, de olho na nova semântica

Nesta parte vamos aprender sobre as novas tags principais de HTML5. É importante entendermos estas tags para não criar confusão na hora da construção da página, a seguir vamos descrever e entender a utilização destes novos elementos.

section

A tag section define uma nova seção “genérica” no documento. É uma tag de estruturação e organização da página, a section passa a ser o primeiro elemento de organização da página, dentro do <body> teremos várias tags <section> organizando cada parte da página. Por exemplo, a home de um website pode ser dividida em diversas seções: introdução ou destaque, novidades, informação de contato, artigos, notícias, etc. A tag <section> é bem genérica e tem a função de organizar os elementos dentro de uma página, basicamente uma <section> serve para agrupar elementos que fazem parte do mesmo bloco ou tipo

nav

O elemento nav representa uma seção da página que contém links para outras partes do website. Nem todos os grupos de links devem ser elementos nav, apenas aqueles grupos que contém links importantes. O elemento nav é bem aplicado para criar o menu principal do site, links na barra lateral, grupo de links no rodapé, etc.

article

Como o próprio nome diz, o elemento article representa um artigo, um post, uma notícia, de uma forma geral um bloco de texto comum.

aside

O elemento aside representa uma barra lateral por exemplo bloco de conteúdo que referencia o conteúdo que envolve o elemento aside. O aside pode ser representado por conteúdos em sidebars em textos impressos, publicidade ou até mesmo para criar um grupo de elementos nave outras informações separados do conteúdo principal do website.

header

O elemento header representa um cabeçalho, uma introdução ou um título de um artigo por exemplo. O elemento header pode ser usado para cabeçalho da página ou de um article por exemplo, devemos tomar cuidado para não confundir com a tag <head>, são tags distintas, tem funções bem diferentes. Com a tag <header> aplicada à página de uma forma geral, podemos inserir logotipo, nome do site, slogan, barra de pesquisa, etc, geralmente é a parte superior do site.

footer

O elemento footer representa o rodapé da página. Seria o último elemento antes de fechar a tag HTML. O elemento footer pode ser usado para a página ou também no final de uma seção.

time

Este elemento serve para marcar parte do texto referente a horário ou data.

mark

Com esta tag podemos marcar um trecho de um texto, colocar em evidência parte do texto.

Pronto agora que já aprendemos sobre as principais novas tags de HTML5 já podemos iniciar o desenvolvimento de uma página, estas são só as tags principais para a nova semântica de HTML5, existem muitas outras tags que iremos aprender, para vídeo, áudio, canvas, etc, a medida que formos evoluindo em nosso curso vamos aprendendo as novas tags de HTML5.

Outro fator importante é sobre o código de formatação CSS, na página HTML iremos inserir somente código HTML, toda parte de formatação será feita em um arquivo externo CSS, portanto à medida que formos evoluindo em HTML veremos paralelamente CSS e em alguns casos também Javascript.

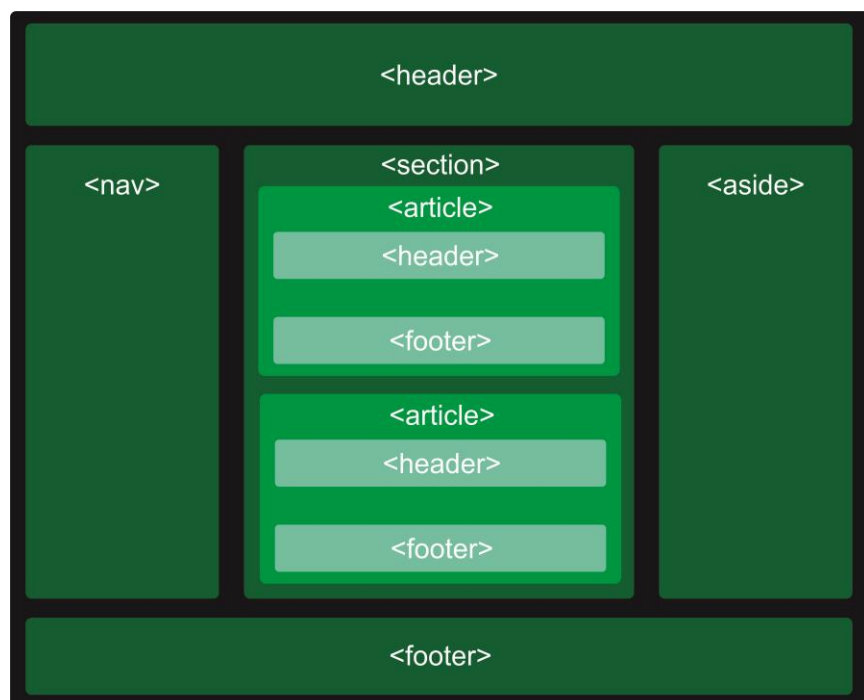
Então, com todo conhecimento básico adquirido, vamos começar logo nossa primeira página em HTML5.

Os novos elementos de HTML5

A seguir vamos entender os principais novos elementos para agrupamento de texto/conteúdo em HTML5, antes de começar é importante destacar que a tag `<div>` permanece ativa, embora a maioria dos tutoriais e textos sobre HTML5 encontrados na web não usam ou se referem à a tag `<div>` para formatação e disposição dos elementos, alguns materiais citam até uma certa proibição ou grande restrição no uso desta tag, a verdade é que não existe essa proibição, podemos usar a tag `<div>` da mesma forma que anteriormente que nossa página vai continuar sendo HTML5, o que se pode entender é uma nova forma de trabalho, uma segunda opção de estruturação da página.

Nesta parte vamos entender a funcionalidade dos novos elementos para agrupamento de texto/conteúdo de HTML5, vamos entender como utilizar estes novos elementos, iremos criar duas estruturas usando somente os novos elementos, mas por razões de aprendizagem e não restritivas.

Veja na ilustração a seguir um modelo simples que mostra a semântica dos novos elementos de HTML5, é uma proposta interessante para mostrar a utilização dos novos elementos, se fizer uma pesquisa na web verá outras alternativas e aqui no próprio curso vamos usar de outras maneiras, não se preocupe com a maneira X ou Y ser a correta ou a errada, o importante é que você entenda estes novos elementos.



Veja a seguir o código HTML que representa esta ilustração.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Semântica 1</title>
</head>
<body>

  <header>
    <!-- Conteúdo referente ao cabeçalho da página -->
  </header>

  <nav>
    <!-- Menu principal da página, pode estar dentro do <header> -->
  </nav>

  <aside>
    <!-- Caixa/Barra lateral da página -->
  </aside>

  <section>

    <article>
      <header>
        <!-- Cabeçalho do article -->
      </header>

      <!-- Texto/conteúdo principal da seção -->

      <footer>
        <!-- Rodapé do article -->
      </footer>
    </article>

    <article>
      <header>
        <!-- Cabeçalho do article -->
      </header>

      <!-- Texto/conteúdo principal da seção -->

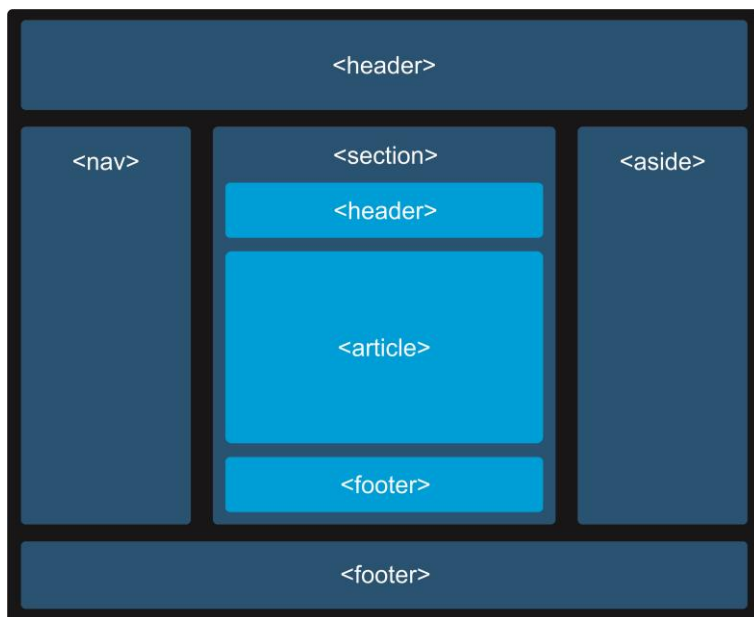
      <footer>
        <!-- Rodapé do article -->
      </footer>
    </article>

  </section>

  <footer>
    <!-- Conteúdo referente ao rodapé da página -->
  </footer>

</body>
</html>
```

Vamos ver outra forma de utilização dos elementos, note na ilustração a seguir que os elementos <header> e <footer> estão sendo usados para o elemento <section> e não para o <article> como no modelo anterior.



Vamos ao código para esta estrutura.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Semântica 2</title>
</head>
<body>

  <header>
    <!-- Conteúdo referente ao cabeçalho da página -->
  </header>

  <nav>
    <!-- Menu principal da página, pode estar dentro do <header> -->
  </nav>

  <aside>
    <!-- Caixa/Barra lateral da página -->
  </aside>

  <section>
    <header>
      <!-- Cabeçalho do section -->
    </header>

    <article>
      <!-- Texto/conteúdo principal da seção -->
    </article>

    <footer>
      <!-- Rodapé do section -->
    </footer>
  </section>

  <footer>
    <!-- Conteúdo referente ao rodapé da página -->
  </footer>

</body>
</html>
```

Obviamente que se você digitar os códigos anteriores, salvar e tentar abrir no browser, você não verá nada, pois inserimos somente o código para a estrutura/esqueleto, mesmo se adicionarmos textos, ainda precisaremos da formatação CSS, mas o intuito nesta parte do curso é fazer com que entenda o uso das novas tags e não construir de fato as páginas, esta tarefa nós faremos no próximo passo.

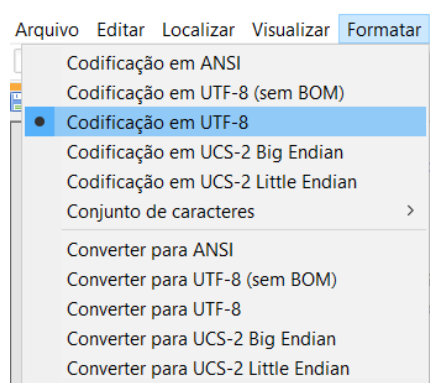
Criando nossa primeira página em HTML5

Vamos começar o trabalho, pegue sua xícara de café e vamos começar.

Para o desenvolvimento das páginas propostas aqui no curso podemos usar dois aplicativos, um o “Bloco de Notas” está disponível no Windows ou outro semelhante de outro S.O, outro programa um pouco mais completo é o Notepad++ que você pode baixar pelo link oficial (notepad-plus-plus.org).

Vamos usar o “Notepad++” por facilitar o entendimento e visualização do código.

Antes de iniciar a digitação do código, vamos configurar o “Notepad++” para que salve o arquivo com a página de códigos “utf-8” como padrão. Clique no menu “Formatar” e selecione a opção “Codificação em UTF-8”.



Com a página de códigos configurada no programa já podemos iniciar o trabalho.

Caso ainda não tenha um arquivo novo, em branco, clique no menu “Arquivo – Novo”, na nova página digite o código básico, mostrado a seguir.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Primeira página do curso HTML 5</title>
</head>
<body>

  <header>
  </header>

  <section>
  </section>

  <footer>
  </footer>

</body>
</html>
```

Salve este arquivo com nome “pagina1.html” em alguma pasta de sua preferência, em seguida, copie os arquivos pac.jpg, pacbqr.jpg e pasl.jpg que estão na pasta “pg1” para a mesma pasta em que acabou de salvar o arquivo html.

Até aqui nada de muito complicado, em relação a HTML4 as únicas novidades são a presença das tags <header>, <section> e <footer>.

Vamos adicionar o conteúdo do cabeçalho, na tag <header> digite o código destacado em vermelho no modelo a seguir.

```
<header>
  <h1>Parques das Águas / Sul de Minas</h1>
  <h2>Confira alguns parques com instâncias minerais localizados no Sul do Estado de Minas gerais</h2>
  <p>
    Nesta página vamos mostrar um pouco de alguns dos diversos parques das águas de Minas gerais.
  </p>
</header>
```

Sem grandes novidades, com somente tags conhecidas como <h1>, <h2> e <p>, salve as alterações nesta página e abra o arquivo em seu browser para ver os resultados.

Com o cabeçalho pronto, vamos adicionar nosso primeiro artigo, dentro da tag <section> adicione todo o conteúdo em vermelho mostrado na ilustração a seguir.

```
<section>
  <header>
    <h1>Parque das Águas de Caxambu</h1>
    <h2>Caxambu - Minas Gerais - Rua João Carlos, 100 - Centro</h2>
  </header>
  <article>
    
    <p>
      O parque possui 12 fontes de água mineral com propriedades curativas diferentes. Opções para a prática de esportes. Piscina semi-olímpica com água mineral (adulto e criança) e um gêiser que explode três vezes ao dia.
    </p>
  </article>
</section>
```

Note que dentro da tag <section> temos um novo <header>, porém este não irá marcar o cabeçalho da página e sim o cabeçalho da tag <article> logo abaixo, na tag <article> temos uma imagem e um texto parágrafo abaixo da imagem, salve estas alterações e atualize ou abra a página no browser, compare com a ilustração a seguir, seu conteúdo deverá estar bem semelhante.



Um ponto importante a observar é que não tem uma necessidade extrema de usar a tag <header> dentro da <section>, simplesmente para marcar um bloco de cabeçalho, mas poderíamos usar direto <h1> e <h2>.

O texto principal da <section> foi inserido em um bloco <article>, que neste caso tem uma imagem e um parágrafo, mas tenha em mente que este texto poderia ser maior e conter vários parágrafos.

E quanto à formatação? Sei que muitos de vocês estão ansiosos pela formatação, mas precisamos ter calma, vamos inserir todo conteúdo da página e deixar a formatação para o final.

Nosso próximo passo é inserir o restante do conteúdo da nossa página, digite o restante do código da <section> como destacado em vermelho no modelo a seguir.

```
<section>
  <header>
    <h1>Parque das Águas de Caxambu</h1>
    <h2>Caxambu - Minas Gerais - Rua João Carlos, 100 - Centro</h2>
  </header>
  <article>
    
    <p>
      O parque possui 12 fontes de água mineral com propriedades curativas diferentes. Opções
      para a prática de esportes. Piscina semi-olímpica com água mineral (adulto e criança) e um gêiser que explode
      três vezes ao dia.
    </p>
  </article>

  <header>
    <h1>Parque das Águas de São Lourenço</h1>
    <h2>São Lourenço - Sul de Minas Gerais</h2>
  </header>
  <article>
    
    <p>
      No Parque das águas você encontrará: atrações, banhos e massagens, gruta & igreja, jardim
      japonês além de restaurantes e lojas.
      O Parque das águas tem ao todo nove fontes.
    </p>
  </article>

  <header>
    <h1>Parque das Águas de Cambuquira</h1>
    <h2>Cambuquira - Sul de Minas Gerais</h2>
  </header>
  <article>
    
    <p>
      O Parque das Águas de Cambuquira reúne em suas fontes diversos tipos de águas, sendo elas
      ferruginosa, gasosa, sulfurosa, magnesiana e litinada.
    </p>
  </article>
</section>
```

Pronto, agora vamos adicionar o texto do rodapé.

```
<footer>
  <p>Página desenvolvida para o curso de HTML5 do Canal Fessor Bruno. (todos os direitos reservados)</p>
</footer>
```

Assim finalizamos o conteúdo da página e podemos partir para a formatação, confira a imagem com o código a seguir.

```

1 <!DOCTYPE HTML>
2 <html lang="pt-br">
3 <head>
4   <meta charset="utf-8">
5   <link rel="stylesheet" href="estilos.css">
6   <title>Primeira página do curso HTML 5</title>
7 </head>
8 <body>
9
10 <header>
11   <h1>Parques das Águas / Sul de Minas</h1>
12   <h2>Confira alguns parques com instâncias minerais localizados no Sul do Estado de Minas Gerais</h2>
13 <p>
14   Nesta página vamos mostrar um pouco de alguns dos diversos parques das águas de Minas Gerais.
15 </p>
16 </header>
17
18 <section id="parques">
19   <header>
20     <h3>Parque das Águas de Caxambu</h3>
21     <h4>Caxambu - Minas Gerais - Rua João Carlos, 100 - Centro</h4>
22   </header>
23   <article>
24     
25     <p>
26       O parque possui 12 fontes de água mineral com propriedades curativas diferentes. Opções para a prática de esportes. Piscina semi-olímpica com água mineral (adulto e
27     </p>
28   </article>
29
30   <header>
31     <h3>Parque das Águas de São Lourenço</h3>
32     <h4>São Lourenço - Sul de Minas Gerais</h4>
33   </header>
34   <article>
35     
36     <p>
37       No Parque das águas você encontrará: atrações, banhos e massagens, gruta e igreja, jardim japonês além de restaurantes e lojas.
38       O Parque das águas tem ao todo nove fontes.
39     </p>
40     <div class="vazio"></div>
41   </article>
42
43   <header>
44     <h3>Parque das Águas de Cambuquira</h3>
45     <h4>Cambuquira - Sul de Minas Gerais</h4>
46   </header>
47   <article>
48     
49     <p>
50       O Parque das Águas de Cambuquira reúne em suas fontes diversos tipos de águas, sendo elas ferruginosa, gasosa, sulfurosa, magnesiana e litinada.
51     </p>
52     <div class="vazio"></div>
53   </article>
54 </section>
55
56 <footer>
57   <p>Página desenvolvida para o curso de HTML5 do Canal Fessor Bruno. (todos os direitos reservados)</p>
58 </footer>
59
60 </body>
61 </html>

```

Agora já podemos iniciar a formatação, salve as alterações feitas e sem delonga vamos prosseguir.

Iniciando em CSS

No “Notepad++” ou no “Bloco de Notas do Windows” crie um novo arquivo e o salve com nome “estilos.css” na pasta que está todo o trabalho anterior.

Anexando folha de estilos externas

Com este arquivo criado agora precisamos anexá-lo em nossa página HTML, para isto usamos a tag <link> configurada em “stylesheet” (folha de estilos), adicione o código destacado em vermelho em sua página HTML.

```

<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <title>Primeira página do curso HTML 5</title>
</head>

```

Agora no arquivo “estilos.css” vamos adicionar nosso primeiro código de formatação do corpo da página <body>, não irei abordar CSS do início com tanto afincio, então, se você não sabe CSS, precisa assistir as aulas de HTML do “CanalFessorBruno”.

Propriedades margin, width e background-color

Digite os comandos a seguir no arquivo “estilos.css”.

```

body{
  margin: 0 auto;
  width: 960px;
}

```

```
background-color: #F0F0E0;
```

No código acima, adicionamos comandos de formatação para a tag <body>, configuramos as margens, largura e cor de fundo, vejamos a seguir.

margin: 0 auto; → Este comando configura as margens, neste caso, configura todas as margens em “0px” e o parâmetro “auto” configura o conteúdo no centro, neste caso alinha o <body> no centro da página.

width: 960px; → Nesta linha configuramos a largura do body para 960 pixels.

background-color: #F0F0E0; → Aqui usamos o comando para configurar a cor do fundo da página, neste caso usamos hexadecimal para referenciar a cor desejada.

Salve estas alterações e atualize a página HTML para ver as mudanças, veja que agora o fundo está em uma tonalidade de bege e todo conteúdo está centralizado com largura máxima de 960px.



Nosso próximo passo é formatar o cabeçalho da página, neste caso, como temos vários <header> em nossa página não vamos criar uma formatação CSS para a tag <header>, pois se não, todas os elementos <header> ficarão iguais e não queremos isto.

Para formatarmos somente o <header> da página, podemos criar uma classe ou usar um ID, neste caso, vamos formatar usando a referência do ID do elemento.

No arquivo HTML, vamos alterar a tag <header> como no código a seguir, adicione o código destacado em vermelho, assim definimos que esta tag <header> será identificada pelo ID “cab”.

```
<header id="cab">
  <h1>Parques das Águas / Sul de Minas</h1>
  <h2>Confira alguns parques com instâncias minerais localizados no Sul do Estado de Minas gerais</h2>
  <p>
    Nesta página vamos mostrar um pouco de alguns dos diversos parques das águas de Minas gerais.
  </p>
</header>
```

A propriedade padding e border-radius para arredondar os cantos

Agora vamos criar a formatação para o <header> identificado por “cab”, vamos configurar a cor de fundo e os “paddings”.

*padding → A forma mais fácil de entender padding é entender como uma margem interna, podemos traduzir ao pé da letra como “preenchimento” como substantivo traduzimos como acolchoamento que fica mais fácil de entender, veja a ilustração a seguir.



Agora que já sabemos onde a configuração do padding afeta, vamos configurar nosso <header> com ID “cab”. Adicione os comandos a seguir no nosso arquivo “estilos.css”.

```
#cab{
  background-color: #508050;
  padding: 10px 20px 10px 20px;
  border-radius: 0px 0px 20px 20px;
  color: #FFF;
}
```

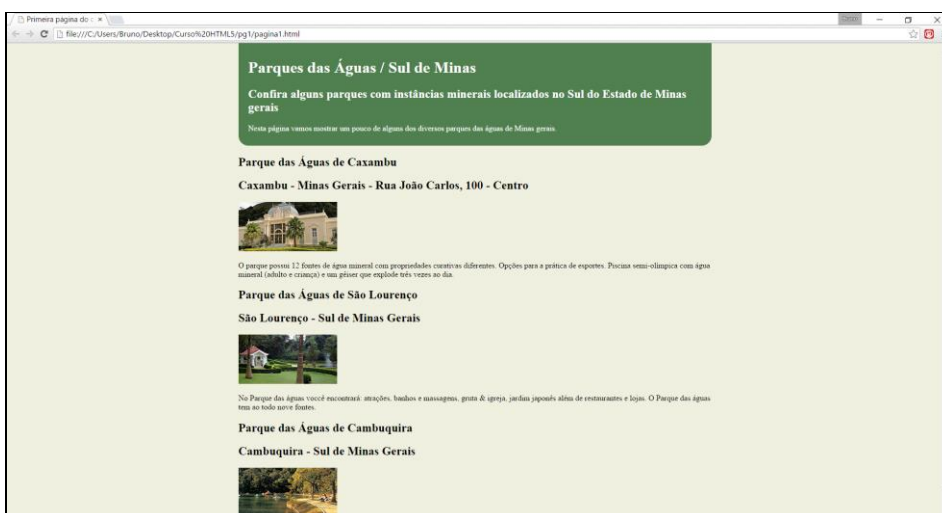
#cab → Identificador para referenciar o elemento com ID “cab” no HTML.

padding: 10px 20px 10px 20px; → Formatação do padding, neste caso, configuramos os quatro lados, iniciando pelo topo (top) com valor 10px na sequência 20px na direita (right), 10px para baixo (bottom) e 20px para esquerda (left), para ficar mais fácil a associação siga o modelo: “padding: top right bottom left”.

border-radius: 0px 0px 20px 20px; → Este comando de formatação está na especificação de CSS3, e serve para arredondar os cantos, no nosso caso, vamos arredondar somente os cantos de baixo, então, usamos “0px 0px 20px 20px”, lembre-se que neste caso, iniciamos com o canto superior esquerdo (se), depois canto superior direito (sd), canto inferior direito (id) e por último canto inferior esquerdo (ie).

border-radius: se sd id ie;

color: #FFF; → O comando color formata a cor da fonte, para a cor, estamos usando o formato hexadecimal resumido, como o R e o G e o B serão FF, não precisamos colocar #FFFFFF, podemos resumir usando um dígito para cada canal, #FFF. Salve as alterações e atualize a página HTML para ver o resultado.



Agora já podemos formatar a <section> com o conteúdo principal. Vamos adicionar o parâmetro ID na tag <section>, siga o código a seguir destacado em vermelho para alterar sua página.

```
</header>

<section id="parques">
  <header>
    <h1>Parque das Águas de Caxambu</h1>
    <h2>Caxambu - Minas Gerais - Rua João Carlos, 100 - Centro</h2>
  </header>
```

Salve o arquivo HTML e vamos adicionar a formatação no arquivo “estilos.css” como no código a seguir.

```
#parques{
  background-color: #505080;
  padding: 0px 10px 10px 10px;
  border-radius: 20px;
}
```

Como não há nada de novo a formatação acima, não vou dar muitos detalhes, salve o arquivo CSS e atualize a página HTML para ver o resultado.



Vamos formatar o rodapé para que fique parecido com o cabeçalho, para esta formatação não vamos precisar identificar com o parâmetro ID, pois, só temos um elemento <footer>, assim, basta formatar este elemento, adicione o código a seguir no arquivo “estilos.css”.

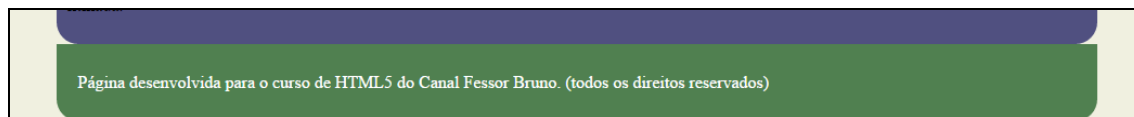
```
footer{
  background-color: #508050;
  padding: 10px 20px 10px 20px;
  border-radius: 20px 20px 0px 0px;
  color: #FFF;
  margin-top: 20px;
}
```

Note que é praticamente a mesma formatação que utilizamos para o ID “cab”. É muito importante otimizar nosso código, mesmo que se trate de um código de formatação, podemos reutilizar o código do ID “cab” para nosso <footer> da seguinte maneira.

Apague este código que acabamos de adicionar para o <footer> e adicione o elemento “footer” no CSS de formatação do “#cab” como no destaque em vermelho no código a seguir.

```
#cab, footer{
  background-color: #508050;
  padding: 10px 20px 10px 20px;
  border-radius: 0px 0px 20px 20px;
  color: #FFF;
}
```

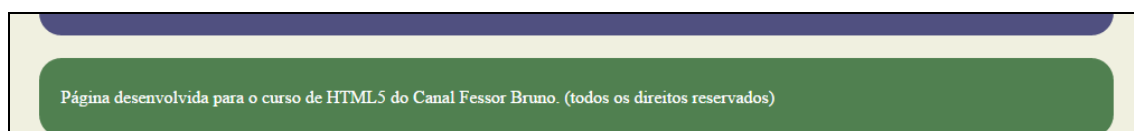
Salve estas alterações e atualize a página HTML para ver os resultados, veja que o rodapé ficou idêntico ao cabeçalho.



Vamos adicionar novamente o elemento “footer” no CSS e alterar somente as propriedades que serão diferentes, siga o código mostrado a seguir.

```
footer{
  border-radius: 20px;
  margin-top: 20px;
}
```

Note que alteramos os cantos que serão arredondados, neste caso, como todos os cantos serão arredondados, basta inserir o valor do raio uma única vez e adicionamos uma margem de 20px no topo do rodapé.



Para esta página, já terminamos com o cabeçalho e o rodapé, agora vamos nos concentrar nos elementos principais da página que estão no <section>.

Vamos formatar as imagens contidas na nossa <section>, iremos usar uma referência para formatar somente as imagens que estão na <section> com ID “#parques” e não qualquer imagem da página.

Entendendo uso das propriedades float e clear

No arquivo CSS adicione os comandos a seguir.

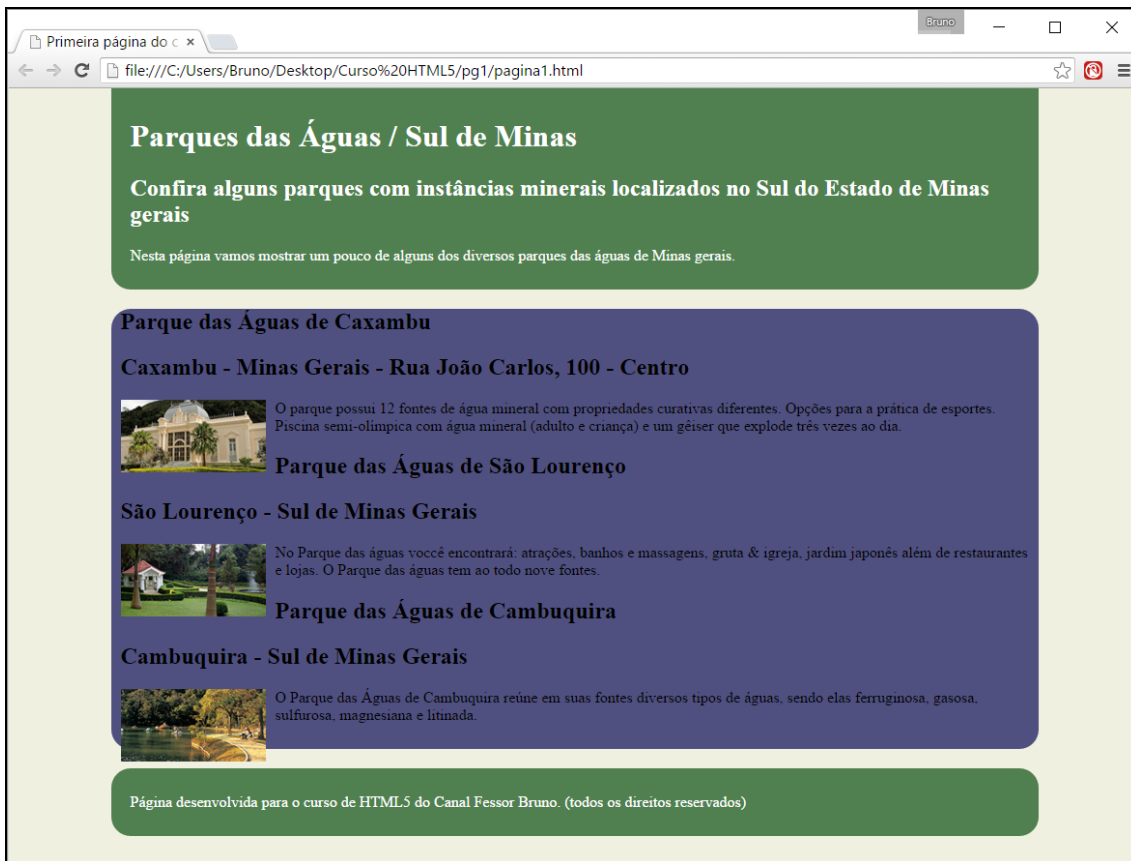
```
#parques img{
  float: left;
  margin: 0px 10px 0px 0px;
  width: 150px;
}
```

#parques img → Esta é a maneira de referenciar as imagens que estão na <section> com ID “#parques”, desta maneira, somente estas imagens receberão esta formatação e não as imagens que estiverem fora da <section>.

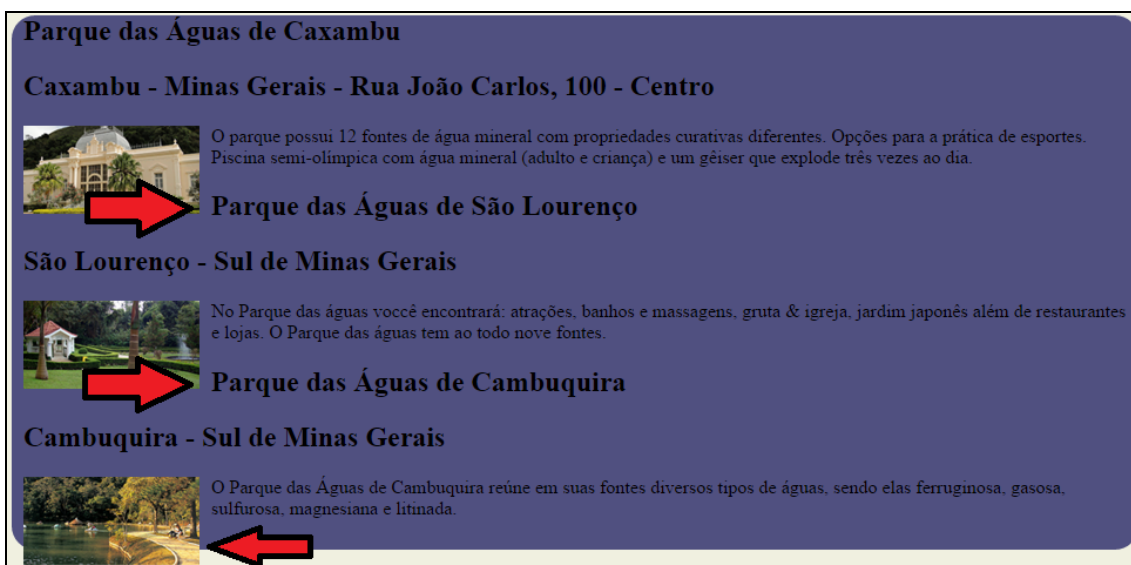
float: left; → Esta propriedade especifica que o elemento deve flutuar em relação aos outros elementos, basicamente retira o elemento do fluxo normal dos objetos. Nesta formatação as imagens irão flutuar e ficarão alinhadas à esquerda (left) ou direita (right), qualquer elemento subsequente a um elemento “float” irá contorná-lo.

As propriedades margin e width nós já conhecemos então não vou explicar novamente.

Salve e atualize para ver os resultados, compare com a ilustração a seguir.



Com as imagens alinhadas à esquerda resolvemos parte do problema, mas veja na ilustração a seguir que ainda não está 100% resolvido, pois, agora para a <section> a imagem que está flutuando sai do fluxo normal dos elementos e não ocupa mais espaço como uma caixa porque ela está flutuando.



Como resolver este novo problema? Usando “clear” é claro, quase sempre que usarmos “float” vamos precisar usar “clear”, vamos entender isto.

A propriedade “clear” é muito importante no controle de elementos flutuantes, basicamente ela define que do ponto onde adicionamos “clear” em diante os elementos subsequentes não irão mais contornar o “float”.

Então, basicamente “clear” não é para o elemento “float” e sim para os elementos subsequentes, “clear” limpa a ação do “float” e daí para frente os elementos não irão mais contornar o “float”.

Os valores possíveis para a propriedade “clear” são:

none: Não impede o contorno de elementos flutuantes (default).

left: Evita a fluidez do conteúdo à direita dos elementos com float:left; permanecendo em baixo dele.

right: O mesmo que “left”, evitando fluir o conteúdo à esquerda dos elementos com float:right.

both: O mesmo que “left” e “right”, o conteúdo fica em baixo dos elementos flutuantes não importando para que lado flutuem.

inherit: Herda o valor de clear do elemento pai, caso haja.

Agora que entendemos o problema e a solução, vamos criar a classe para a formatação “clear” no CSS e adicionar as tags que irão aplicar o clear no HTML.

No arquivo “estilos.css” adicione o código a seguir.

```
.vazio{
    clear: both;
}
```

No código acima criamos uma classe chamada “vazio”, o ponto antes do nome simboliza que é uma classe, como parâmetro usamos simplesmente “clear:both”.

Com a classe criada, podemos adicionar <div> ou
 para usar a classe “.vazio”. Iremos usar o
, então adicione os comandos destacados no seu código de acordo com o modelo a seguir.


```
<section id="parques">
  <header>
    <h1>Parque das Águas de Caxambu</h1>
    <h2>Caxambu - Minas Gerais - Rua João Carlos, 100 - Centro</h2>
  </header>
  <article>
    
    <p>
      O parque possui 12 fontes de água mineral com propriedades curativas diferentes. Opções
      para a prática de esportes. Piscina semi-olímpica com água mineral (adulto e criança) e um gêiser que explode
      três vezes ao dia.
    </p>
    <br class="vazio">
  </article>
  <header>
    <h1>Parque das Águas de São Lourenço</h1>
    <h2>São Lourenço - Sul de Minas Gerais</h2>
  </header>
  <article>
    
    <p>
      No Parque das águas vocccê encontrará: atrações, banhos e massagens, gruta & igreja, jardim
      japonês além de restaurantes e lojas. O Parque das águas tem ao todo nove fontes.
    </p>
    <br class="vazio">
  </article>
  <header>
    <h1>Parque das Águas de Cambuquira</h1>
    <h2>Cambuquira - Sul de Minas Gerais</h2>
  </header>
  <article>
    
    <p>
      O Parque das Águas de Cambuquira reúne em suas fontes diversos tipos de águas, sendo elas
      ferruginosa, gasosa, sulfurosa, magnesiana e litinada.
    </p>
    <br class="vazio">
  </article>
</section>
```

<br class="vazio"> → Aplicamos a classe "vazio" que tem a propriedade "clear" no elemento
, então o conteúdo subsequente a este
 para de contornar a imagem que está configurada com a propriedade "float".

Salve as alterações feitas no arquivo CSS e HTML e atualize a página para ver as alterações, compare com a ilustração a seguir.

Parque das Águas de Caxambu


Caxambu - Minas Gerais - Rua João Carlos, 100 - Centro



O parque possui 12 fontes de água mineral com propriedades curativas diferentes. Opções para a prática de esportes. Piscina semi-olímpica com água mineral (adulto e criança) e um géiser que explode três vezes ao dia.

Parque das Águas de São Lourenço


São Lourenço - Sul de Minas Gerais



No Parque das águas você encontrará: atrações, banhos e massagens, gruta & igreja, jardim japonês além de restaurantes e lojas. O Parque das águas tem ao todo nove fontes.

Parque das Águas de Cambuquira

Cambuquira - Sul de Minas Gerais



O Parque das Águas de Cambuquira reúne em suas fontes diversos tipos de águas, sendo elas ferruginosa, gasosa, sulfúrea, magnésiana e litinada.

Veja que agora nosso problema com o "float" das imagens foi resolvido.

Vamos continuar a formatação da página.

A propriedade border

Vamos adicionar uma linha tracejada abaixo das imagens, porém, como as imagens estão dentro de um bloco <article> basta configurar a borda de baixo da caixa <article>, assim sempre que tiver um <article> será traçada a borda de baixo, vamos a esta configuração.

No arquivo "estilos.css" digite o código a seguir.

```
#parques article{
    border-bottom: 1px dashed #A0FFFF;
    padding-bottom: 10px;
}
```

Note que estamos formatando somente os elementos <article> que estão dentro da <section> com ID "#parques".

A sintaxe do comando que usamos pode ser descrita da seguinte forma.

border-bottom: 1px dashed #A0FFFF;
border-bottom: **largura da borda** **tipo de borda** **cor da borda**;

dashed = pontilhada

caso queira formatar as quatro bordas da mesma maneira, basta usar somente o comando border, o exemplo a seguir irá formatar as quatro bordas iguais.

border: 1px dashed #A0FFFF;

De forma semelhante a border-bottom podemos formatar as quatro bordas de formas diferentes, usando os parâmetros border-bottom, border-top, border-left e border-right.

Com a propriedade configurada salve as alterações e atualize a página para ver o resultado.

color e text-align

Vamos formatar os textos principais da nossa página que estão na <section id="parques">, são os textos <h1>, <h2> e <p>.

O parâmetro "color" formata a cor do texto, basta indicar a cor é muito simples e para alinhar o texto usamos o parâmetro "text-align".

Adicione os comandos a seguir no arquivo CSS.

```
#parques h1, #parques h2{
    color:#FFF;
}

#parques p{
    text-align: justify;
    color:#A0FFFF;
}
```

Para "text-align" podemos usar os parâmetros left (esquerdo), right (direito), center (centro) e justify (justificado).

Salve as alterações e atualize a página para ver os resultados.

font-size

O parâmetro font-size tem a função de controlar o tamanho da fonte, para concluir a formatação da nossa primeira página HTML5 vamos formatar a fonte das tags <h1> e <h2>, para <h2> vamos mudar o tamanho e <h1> o padding-top, como não especificamos qual <h1> e <h2>, todos receberão estas formatações, então adicione o código a seguir no arquivo "estilos.css".

```
h2{
    font-size: 15px;
}

h1{
    padding-top:20px;
}
```

Salve as alterações e atualize a página, a ilustração a seguir, mostra como sua página deverá estar sendo apresentada.



Pronto, terminamos nossa primeira página em HTML5, neste ponto, o importante é você ter entendido o funcionamento das novas tags apresentadas <section>, <header>, <article> e <footer>, além da estrutura básica, nosso próximo trabalho vai ser um pouco mais elaborado, usaremos <nav> e <aside> em nossa próxima página.

Segunda página em HTML5

Nossa segunda página será mais elaborada que a primeira, iremos trabalhar com elementos como <nav> e <aside> além de aprender novos parâmetros CSS.

A seguir temos a ilustração do layout que iremos criar nesta etapa do curso, os arquivos necessários estão na pasta “pg2”, mãos a obra.



Crie um novo arquivo no “Notepad++” e o salve com nome “index.html” na pasta “pg2”.

Digite o código básico como o modelo a seguir.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <title>Escola de Inglês</title>
</head>
<body>

  <header id="cab">
  </header>

  <section id="banner">
  </section>

  <section id="cxLateral">
  </section>

  <section id="main">
  </section>

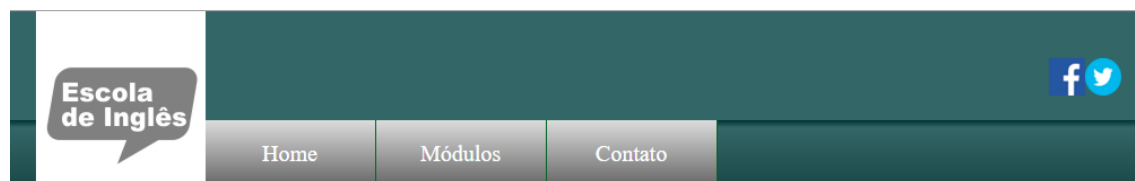
  <footer>
  </footer>

</body>
</html>
```

Note que mesmo que ainda não tenhamos o arquivo “estilos.css” nós já inserimos a o link e já inserimos três seções banner, cxLateral e main.

Criando um menu horizontal

Nosso primeiro passo será construir o <header> da nossa página, neste trabalho iremos adicionar o menu principal <nav> dentro do cabeçalho <header>, veja na ilustração a seguir como ficará nosso cabeçalho.



Neste cabeçalho vamos usar três imagens logo.jpg, facebook.png e twitter.png, as três imagens estão localizadas na pasta “pg2”.

```
<header id="cab">
  
  <div id="social">
    
    
  </div>
  <nav id="menu">
    <ul>
      <li><a href="index.html" target="_self">Home</a></li>
      <li><a href="modulos.html" target="_self">Módulos</a></li>
      <li><a href="contato.html" target="_self">Contato</a></li>
    </ul>
  </nav>
</header>
```

As imagens e a div não são novidade, portanto, só vou dar detalhes a estes itens na parte da formatação, a parte mais importante desse código é o menu, note que temos o elemento <nav> com uma estrutura de lista não ordenada dentro dele, salve as alterações abra o arquivo para ver o que temos até o momento.



Vamos nos concentrar na construção do menu. Para isso precisamos do arquivo CSS, crie um novo arquivo e o salve na pasta “pg2” com nome “estilos.css”.

Observe que a tag <nav> contém um parâmetro ID configurado, <nav id=“menu”>, vamos configurá-lo no arquivo CSS, digite o código a seguir.


```
#menu{
    float:left;
    margin-top:93px;
    width: 453px;
}

#menu a:hover {
    color:#000;
}

#menu ul {
    list-style: none;
    background: linear-gradient(to bottom, #DDD, #666);
    padding: 0px;
    margin: 0px;
}

#menu li {
    height: 57px;
    display: table-cell;
    text-align: center;
    vertical-align: middle;
    width:150px;
    border-right: 1px solid #062;
}

#menu ul li a{
    text-decoration: none;
    font-size:20px;
    color: #FFF;
}
```

Veja na ilustração a seguir como deverá ficar nosso menu, logo em seguida vamos detalhar esta formatação.



Os primeiros comandos de formatação são a base do posicionamento e tamanho do menu, como não são comandos novos aqui não tem novidades.

```
#menu{
    float: left;
    margin-top: 93px;
    width: 453px;
}
```

A próxima formatação diz respeito ao estado “hover” que é acionado quando o mouse está sobre o link, em nossa configuração estamos mudando a cor da fonte para branco.

```
#menu a:hover {  
    color: #000;  
}
```

Na sequência adicionamos os comandos para formatação da lista não ordenada (UL) já que nosso menu foi construído com base em uma .

```
#menu ul {  
    list-style: none;  
    background: linear-gradient(to bottom, #DDD, #666);  
    padding: 0px;  
    margin: 0px;  
}
```

list-style: none; → Parâmetro que configure o tipo de marcador da lista, configurado em “none” os marcadores são retirados da lista.

background: linear-gradient(to bottom, #DDD, #666); → Esta é uma configuração nova do CSS3 para o background, cor gradient, vou dar mais detalhes sobre os gradientes depois que acabarmos a construção desta página.

padding e margin não são mais novidades.

Os próximos comandos formatam o local “container” onde os textos dos menus ficam, os itens da lista .

```
#menu li {  
    height: 57px;  
    display: table-cell;  
    text-align: center;  
    vertical-align: middle;  
    width: 150px;  
    border-right: 1px solid #062;  
}
```

height: 57px; → Medida da altura do elemento.

display: table-cell; → Configura o element para se “comportar” como se fosse uma célula de uma tabela, basicamente como se fosse um <td>.

text-align: center; → Alinhamento horizontal centralizado.

vertical-align: middle; → Alinhamento vertical no meio, como se estivesse no meio da célula da tabela.

width: 150px; → Medida da largura.

border-right: 1px solid #062; → Configuração da borda direita, espessura 1px, linha sólida e cor verde, esta linha servirá como uma linha divisória para os menus.

Os últimos comandos formatam o texto, como está configurado como link, usamos a formatação para a tag <a>.

```
#menu a {  
    text-decoration: none;  
    font-size: 20px;
```

```
color: #FFF;
}
```

text-decoration: none; → Retira a formatação do link, remove o sublinhado padrão do link.

font-size: 20px; → Tamanho da fonte.

color: #FFF; → Cor padrão do link.

Centralizando a página e configurando fundo do body

Com o menu devidamente configurado, vamos configurar a imagem de fundo do body, na pasta “pg2” temos uma imagem com nome “fundo.gif” é uma imagem pequena de 10x150 pixels onde iremos aplicar no fundo do body e configurar a repetição somente no eixo X.



```
body{
  margin: 0 auto;
  width: 960px;
  background-color: #DDD;
  padding: 0px;
  background: url(fundo.gif) repeat-x;
}
```

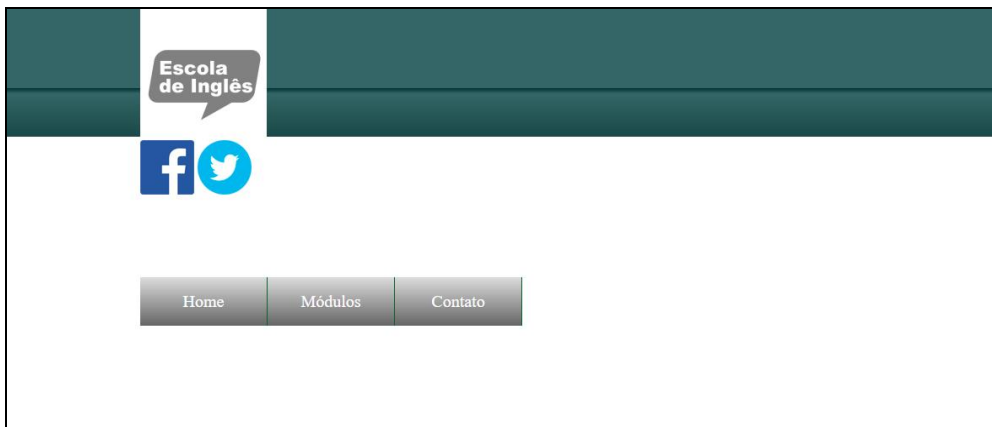
margin: 0 auto; → Configuração de margens em zero de forma que o conteúdo ficará centralizado na janela.

width: 960px; → Largura da página

background-color: #DDD; → Cor de fundo da página.

padding: 0px; → Padding da página.

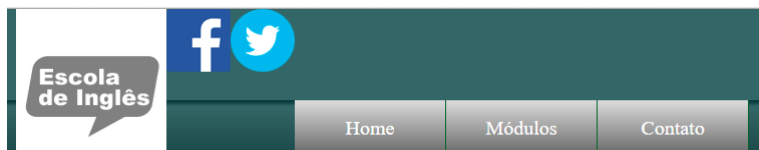
background: url(fundo.gif) repeat-x; → Imagem de fundo configurada para repetir somente no eixo X.



Vamos posicionar o menu ao lado da imagem, para isto basta configurar a imagem em “float”, então adicione o comando de configuração da imagem no arquivo CSS.

```
#cab img{
  float: left;
}
```

Já sabemos a função da propriedade “float”, precisamos configurar a imagem com “float:left;” para que o menu e os ícones “sociais” fiquem ao lado da imagem.



Mas ainda precisamos de um ajustar os ícones sociais para o lado direito. Adicione os comandos a seguir para configurar os ícone sociais.

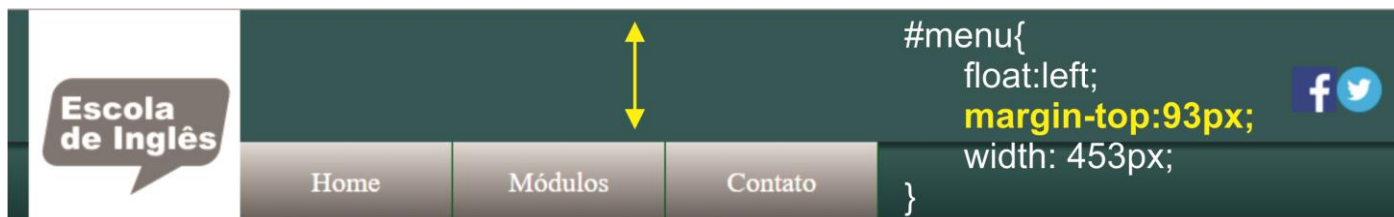
```
#social{
    float:right;
    margin-top:40px;
}

#social img{
    width: 32px;
    height: 32px;
}
```

```
#social{
    float:right; → Flutuando e posicionado à direita
    margin-top:40px; → Margem superior de tamanho 40px
}

#social img{
    width: 32px; → Largura dos ícones sociais
    height: 32px; → Altura dos ícones sociais
}
```

Com estas alterações tudo vai se posicionar de acordo em nosso <header>, salve as alterações e atualize a página, veja que agora está tudo arrumado, mas como o menu se posicionou na parte de baixo exatamente alinhado com a base do logo? Vamos revisar a formatação do menu observando a ilustração a seguir.



Antes de editar o conteúdo principal, precisamos adicionar o “clear”, lembre-se, se usamos “float” precisamos do “clear”.

Então adicione o comando <br class=“fix”> logo abaixo do <header> como mostra o código em vermelho no exemplo a seguir.

```
</ul>
</nav>
</header>

<br class=“fix”>

<section id=“banner”>
</section>
```

Agora vamos criar a classe “fix” no arquivo CSS.

```
.fix{
  clear: both;
}
```

<section> e <aside>

Agora que adicionamos o
 com a propriedade “clear” e finalizamos o <header> vamos adicionar o conteúdo da <section id=“cxLateral”> e da <section id=“main”>, então digite todo o conteúdo de acordo com o modelo em vermelho a seguir.

```
<section id="cxLateral">
  <aside id="lateral">
    <h1>Notícias</h1>
    <article>
      <h2>Aluno aprovado com 100% de aproveitamento</h2>
      <p>O aluno Batman da turma noturna se destacou a ter seu aproveitamento avaliado em 100%, Batman não teve nenhuma falta e nunca deixou de fazer as atividades das aulas, em todas as provas Batman tirou total.</p>
    </article>
    <article>
      <h2>Novas vagas para intercâmbio abertas</h2>
      <p>Aos alunos que desejarem realizar intercâmbio as vagas já estão abertas e são limitadas, procure hoje mesmo a secretaria da escola e garanta esta oportunidade.</p>
    </article>
  </aside>
</section>

<section id="main">
  <article>
    <h1>O melhor curso de inglês do Brasil</h1>
    <p>Com mais de 30 anos no mercado a Escola de Inglês vem oferecendo ensino de qualidade a seus alunos aliado a um preço justo, nossos horários são flexíveis e nossos alunos ainda contam com uma gama de serviços adicionais, como Web Teacher e Home Teacher.</p>
    <p>Estamos sempre prontos a atender a necessidade de nossos alunos e garantir o aprendizado é nosso dever.</p>
    <p>Venha você também para a Escola de Inglês e abra mais portas no mercado de trabalho.</p>
  </article>
  <article>
    <h1>TOEFL</h1>
    <p>O Test of English as a Foreign Language (TOEFL) ou Teste de Inglês como uma Língua Estrangeira é um exame que tem o objetivo de avaliar o potencial individual de falar e entender o inglês em nível acadêmico. É requerido para a maior parte dos estudantes estrangeiros que tentam ingressar em uma universidade de um país em que inglês é a primeira língua. Além disso, instituições como agências governamentais, empresas e programas de estudo podem exigir o teste. A nota no TOEFL é válida por apenas dois anos - após esse prazo, ela é eliminada do banco de dados.</p>
    <p>Turma especiais preparatórias para o TOEFL, consulte as vagas e horários na recepção e faça seu agendamento.</p>
  </article>
  <article>
    <h1>Web Teacher e Home Teacher</h1>
    <p>Sempre pensando em trazer conforto e comodidade aos nossos alunos a Escola de Inglês criou o PAP (Programa de aprendizado pessoal), são dois serviços exclusivos para você que quer ou precisa estudar inglês de qualquer lugar ou qualquer hora.</p>
    <h2>Web Teacher</h2>
    <p>O Web Teacher é nosso programa de estudo on-line, onde nosso aluno tem em sua inteira disposição, 24 horas por dia e 7 dias por semana, um professor virtual para realizar seus estudos e atividades, nosso sistema conta com uma poderosa ferramenta de IA (Inteligência Artificial) programada para atender nossos alunos de acordo com nosso conteúdo e método.</p>
    <h2>Home Teacher</h2>
    <p>Outro serviço de comodidade para nossos alunos é o Home Teacher, um professor que vai até sua casa ou trabalho, você estuda sem sair de casa.</p>
  </article>
</section>
```

Com o conteúdo de texto adicionado, precisamos formatar, então vamos ao arquivo CSS e adicione o código para formatação da <section id=“lateral”> e <section id=“main”>.

```
#lateral{
  float:right;
  background: #DDD;
  width:190px;
```

```
        border-radius:10px;
        padding: 5px;
    }

    #lateral h1{
        font-size:15pt;
        text-align: center;
        border-bottom: 1px solid #000;
        margin-top:-5px;
    }

    #lateral h2{
        font-size:15pt;
    }

    #lateral p{
        text-align: justify;
    }

    #main{
        background: #EEE;
        width:730px;
        border-radius:10px;
        padding: 5px;
    }

    #main h1{
        font-size:20pt;
        border-bottom: 1px solid #000;
    }

    #main h2{
        font-size:15pt;
    }

    #main p{
        text-align: justify;
    }
}
```

Vamos às explicações dos códigos de formatações anteriores.

#lateral{ → <section id="lateral">

float:right; → Flutuar a janela lateral e posicionar à direita

background: #DDD; → Cor de fundo da caixa lateral

width:190px; → Largura da caixa lateral

border-radius:10px; → Arredondamento das bordas da caixa lateral

padding: 5px; → Borda interna da caixa lateral.

}

#lateral h1{ → <h1> dentro da <section id="lateral">

font-size:15pt; → Tamanho da fonte

text-align: center; → Alinhamento do texto ao centro

border-bottom: 1px solid #000; → Borda de baixo do texto <h1>

margin-top:-5px; → Margem superior do texto <h1>, note que o valor é "-5"

}

#lateral h2{ → <h2> dentro da <section id="lateral">

font-size:15pt; → Tamanho da fonte

}

#lateral p{ → <p> dentro da <section id="lateral">

text-align: justify; → Alinhamento justificado do texto (alinha na esquerda e na direita)

}

```
#main{ → <section id="main">
    background: #EEE; → Cor do fundo da caixa main
    width:730px; → Largura
    border-radius:10px; → Arredondamento dos cantos
    padding: 5px; → Margem interna
}

#main h1{ → <h1> dentro da <section id="main">
    font-size:20pt; → Tamanho da fonte
    border-bottom: 1px solid #000; → Borda inferior de cada <h1> dentro da <section id="main">
}

#main h2{ → <h2> dentro da <section id="main">
    font-size:15pt; → Tamanho da fonte
}

#main p{ → <p> dentro da <section id="main">
    text-align: justify; → Alinhamento justificado do texto
}
```

Salve e atualize a página para ver o resultado.



<footer>

Ótimo, estamos evoluindo bem, vamos adicionar o conteúdo do <footer> e formata-lo, digite o conteúdo destacado em vermelho na página HTML.

```
<footer>
    <p>Layout Curso de Inglês</p>
    <p>Canal Fessor Bruno</p>
    <p>www.youtube.com.br/canalfessorbruno</p>
</footer>
```

Note que temos três parágrafos <p> nossa tarefa será se posicionar estes três parágrafos em uma linha só, então vamos ao CSS.

```

footer{
  color:#888;
  border: 1px solid black;
  margin-top: 15px;
}

footer p{
  display:inline-block;
  width:316px;
  text-align:center;
}

```

footer{ → Formatação geral dos elementos do <footer>

color:#888; → Cor do texto

border: 1px solid black; → Bordas, largura 1px linha sólida e cor preto

margin-top: 15px; → Margem superior com altura de 15px

}

footer p{ → Formatação das tags <p> presentes dentro do <footer>

display:inline-block; → Pulo do gato, esta configuração do “display” faz com que os elementos fiquem posicionados na mesma linha.

width:316px; → Largura de cada <p>

text-align:center; → Alinhamento do texto centralizado dentro do <p>

}

Salve as alterações e atualize a página HTML para ver o resultado do <footer>.

Layout Curso de Inglês

Canal Fessor Bruno

www.youtube.com.br/canalfessorbruno

Agora só nos resta o “slider” que iremos construir com as imagens “banner1.jpg”, “banner2.jpg” e “banner3.jpg”. A construção deste banner irá envolver um processo um pouco mais trabalhoso em javascript, **mais uma vez é recomendado que você tenha assistido todas as aulas de HTML do Canal Fessor Bruno, pois o conhecimento básico de javascript que iremos precisar é mostrado nestas aulas.**

Vamos abordar a criação do nosso slider em um capítulo novo, inteiramente dedicado a ele, descanse um pouco e volte com a cabeça fresca para iniciar o próximo capítulo.

Criando um slider básico em javascript

Primeiramente vamos adicionar os elementos necessários na página HTML, na <section id=“banner”> adicione as tags <div> e conforme o código destacado em vermelho a seguir.

```

<section id="banner">
  <div id="slider">
    <div id="b1">
      
    </div>
    <div id="b2">
      
    </div>
    <div id="b3">
      
    </div>
  </div>
</section>

```


Note que temos uma caixa `<div id="slider">` geral e dentro dela temos três caixas `<div id="b1">`, `<div id="b2">` 2 `<div id="b3">` dentro das divs b1, b2 e b3 temos três imagens banner1.jpg, banner2.jpg e banner3.jpg, todas estas imagens estão na pasta pg2, confira esta montagem na ilustração a seguir.



Ao adicionar estas imagens nosso layout irá se “bagunçar” um pouco, não se preocupe, pois, ainda não formatamos este conteúdo.



Vamos adicionar a formatação geral do slider no arquivo CSS, digite os comandos a seguir.

```
#slider{
    width:960px;
    height:384px;
    margin-top:15px;
    margin-bottom:10px;
}

#slider img{
    width:960px;
    height:384px;
}

#b1{
    position: relative;
```

```
    left: 0px;
    top: 0px;
}

#b2{
    position: relative;
    left:0px;
    top: -388px;
}

#b3{
    position: relative;
    left:0px;
    top: -776px;
}
```

#slider{ → Formatação geral da <div id="slider">

width:960px; → Largura do slider

height:384px; → Altura do slider

margin-top:15px; → Margem superior para não ficar muito junto com o <header>

margin-bottom:10px; → Margem inferior para não ficar muito junto com os <section id="cxLateral"> e

<section id="main">

}

#slider img{ → Formatação das imagens

width:960px; → Largura da imagem

height:384px; → Altura da imagem

}

#b1{ → Formatação da <div id="b1"> que contém a imagem banner1.jpg

position: relative; → Posicionamento relativo ao elemento anterior

left: 0px; → Posição X em 0 em relação à div anterior

top: 0px; → Posição Y em 0 em relação à div anterior

}

#b2{ → Formatação da <div id="b2"> que contém a imagem banner2.jpg

position: relative;

left:0px;

top: -388px; → Como o position é relativo ao elemento anterior ele está em baixo da imagem anterior, por isso precisamos diminuir Y/top em -388px

}

#b3{ → Formatação da <div id="b3"> que contém a imagem banner3.jpg

position: relative;

left:0px;

top: -776px; → Mesma lógica do anterior

}

Com esta formatação posicionamos as três divs com as imagens respectivas exatamente no mesmo local, uma por cima da outra.



Agora vamos dar o toque de mágica para fazer o slider funcionar.

Vamos criar nosso arquivo de scripts e anexá-lo ao nosso HTML. Crie um novo arquivo e o salve com nome “scripts.js”.

Após criar o arquivo de scrips, vamos anexar ao arquivo HTML, então, digite o comando destacado em vermelho a seguir.

```
<head>
  <meta charset="ISO-8859-1">
  <link rel="stylesheet" href="estilos.css">
  <script src="scripts.js"></script>
  <title>Escola de Inglês</title>
</head>
```

Mais uma vez eu ressalto que todo esse conteúdo javascript tem explicação nas aulas de HTML do Canal Fessor Bruno, então não deixe de assistir estas aulas.

Com o arquivo “scripts.jp” devidamente criado e anexado ao arquivo HTML, vamos construir o script básico para nosso slider, digite o código a seguir no arquivo de scripts.

```
satual=1;
smax=3;
stmp=3000;

function troca(){
  document.getElementById("b1").style.visibility="hidden";
  document.getElementById("b2").style.visibility="hidden";
  document.getElementById("b3").style.visibility="hidden";
  document.getElementById("b"+satual).style.visibility="visible";
  satual=satual+1;
  if(satual > smax){
    satual=1;
  }
}

function slider(){
  document.getElementById("b1").style.visibility="hidden";
  document.getElementById("b2").style.visibility="hidden";
  document.getElementById("b3").style.visibility="visible";
  sliderTimer=setInterval(troca,stmp);
}
```

Vamos à explicação deste código.

Primeiramente criamos três variáveis para controle do slider.



```
satual=1; //Guarda o slide atual para ser exibido  
smax=3; //Informa o número máximo de slides  
stmp=3000; //Tempo de exibição de cada slide
```

O próximo passo é criar uma função para gerenciar a troca dos slides.

```
function troca(){ //Nome da função para gerenciar a troca dos slides  
  
    //Comandos para ocultar os slides, na verdade ocultar as divs, b1, b2 e b3  
    document.getElementById("b1").style.visibility="hidden";  
    document.getElementById("b2").style.visibility="hidden";  
    document.getElementById("b3").style.visibility="hidden";  
  
    //Após ocultar as divs, vamos mostrar somente a div com o slide atual, informado na variável "satual"  
    document.getElementById("b"+satual).style.visibility="visible";  
  
    //Depois de exibir o slide atual, precisamos preparar a variável para o próximo slide, então incrementamos o  
    //valor da variável satual em 1  
    satual=satual+1;  
  
    //Este IF controla para que o incremento anterior não ultrapasse o número máximo de slides, então, quando  
    //ultrapassar o número máximo de slides, iremos voltar para o primeiro slide  
    if(satual > smax){  
        satual=1;  
    }  
}
```

//A próxima função é responsável por iniciar o trabalho de troca do slider, esta função será chamada quando a página for carregada, irá ocultar dois slides e deixar o último visível e ativará o "timer" que chamará a função de gerenciamento de troca dos slides de três em três segundos.

```
function slider(){  
  
    //Ocultar dois slides e deixar o último visível  
    document.getElementById("b1").style.visibility="hidden";  
    document.getElementById("b2").style.visibility="hidden";  
    document.getElementById("b3").style.visibility="visible";  
  
    //Configuração do interval que irá chamar a função de troca de três em três segundos.  
    sliderTimer=setInterval(troca,stmp);  
}
```

Quase pronto pra funcionar, todo script já está pronto, só precisamos adicionar a chamada da função "slider" na página HTML.

Vamos chamar a função "slider" quando a página for carregada, então, adicione o comando destacado em vermelho na tag <body> como mostra o modelo a seguir.

```
<body onload="slider()">
```

<body onload="slider()"> → Estamos adicionando o evento "onload" que é acionado no carregamento da página, então, quando a página for carregada automaticamente será chamada a função "slider()", isso irá iniciar nosso slider.

Salve todas as alterações e atualize a página para ver o slider funcionando, veja que de três em três segundos a imagem será trocada.

Novos elementos de formulário

HTML5 trouxe uma nova leva de elementos para formulário, além dos tradicionais agora existem uma série de elementos interessantes que vamos ver aqui neste capítulo.

tel

O campo “tel” serve para entrada de número de telefone, porém, como não existe um padrão mundial, não existe máscara nem validação nativas.

```
<form>
  <label>Telephone:</label>
  <input type="tel" name="telefone">
</form>
```

Propriedades para tel		
name	Nome do campo	name=“telefone”
disabled	Desabilita o campo não podendo ser editado	disabled=“disabled” ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form=“id do formulário pai”
autocomplete	Permite que o campo seja auto completado ou não	autocomplete=“on” ou “off”
autofocus	Posicione o cursor no elemento automaticamente ao abrir a página	autofocus=“autofocus” ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength=“50”
pattern	Especifica uma expressão para validação do campo	pattern=“\d{5}”
readonly	Especifica que o campo é “somente leitura”, não permite editar o conteúdo	readonly=“readonly” ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required=“required” ou required
size	Tamanho visível do elemento	size=“30”
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder=“Digite seu nome aqui”
value	Valor inicial do campo	value=“jogador1”
list	Especifica uma lista pelo ID que contém os valores para o campo	list=“id da lista”

search

Campo de texto pré-configurado para pesquisa,

```
<form>
  <label>Search Google:</label>
  <input type="search" results="10" name="busca">
</form>
```

Propriedades para search		
name	Nome do campo	name=“busca”
disabled	Desabilita o campo não podendo ser editado	disabled=“disabled” ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form=“id do formulário pai”
autocomplete	Permite que o campo seja auto completado ou não	autocomplete=“on” ou “off”
autofocus	Posicione o cursor no elemento automaticamente ao abrir a página	autofocus=“autofocus” ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength=“50”
pattern	Especifica uma expressão para validação do campo	pattern=“\d{5}”
readonly	Especifica que o campo é “somente leitura”, não permite editar o conteúdo	readonly=“readonly” ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required=“required” ou required
size	Tamanho visível do elemento	size=“30”
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder=“Digite seu nome aqui”
value	Valor inicial do campo	value=“jogador1”
list	Especifica uma lista pelo ID que contém os valores para o campo	list=“id da lista”
multiple	Especifica que o campo permite múltiplos valores	multiple=“multiple” ou multiple



email

Campo para entrada de e-mails, possui validação padrão, ou seja, se digitar um email inválido e pressionar enter é emitido uma mensagem de alerta.

```
<form>
  <label>E-mail:</label>
  <input type="email" name="email">
</form>
```

Propriedades para email		
name	Nome do campo	name="email"
disabled	Desabilita o campo não podendo ser editado	disabled="disabled" ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form="id do formulário pai"
autocomplete	Permite que o campo seja auto completado ou não	autocomplete="on" ou "off"
autofocus	Posicione o cursor no elemento automaticamente ao abrir a página	autofocus="autofocus" ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength="50"
pattern	Especifica uma expressão para validação do campo	pattern="\d{5}"
readonly	Especifica que o campo é "somente leitura", não permite editar o conteúdo	readonly="readonly" ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required="required" ou required
size	Tamanho visível do elemento	size="30"
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder="Digite seu nome aqui"
value	Valor inicial do campo	value="jogador1"
list	Especifica uma lista pelo ID que contém os valores para o campo	list="id da lista"
multiple	Especifica que o campo permite múltiplos valores	multiple="multiple" ou multiple

url

Campo para endereço de email, possui validação, ao pressionar enter e for um endereço inválido é emitida uma mensagem.

```
<form>
  <label>Endereço do site</label>
  <input type="url" name="site">
</form>
```

Propriedades para url		
name	Nome do campo	name="site"
disabled	Desabilita o campo não podendo ser editado	disabled="disabled" ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form="id do formulário pai"
autocomplete	Permite que o campo seja auto completado ou não	autocomplete="on" ou "off"
autofocus	Posicione o cursor no elemento automaticamente ao abrir a página	autofocus="autofocus" ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength="50"
pattern	Especifica uma expressão para validação do campo	pattern="\d{5}"
readonly	Especifica que o campo é "somente leitura", não permite editar o conteúdo	readonly="readonly" ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required="required" ou required
size	Tamanho visível do elemento	size="30"
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder="Digite seu nome aqui"
value	Valor inicial do campo	value="jogador1"
list	Especifica uma lista pelo ID que contém os valores para o campo	list="id da lista"

Datas e Horas

Campo pré configurado para entrada de valores como datas e horas, possui algumas variações descritas a seguir.

datetime-local – campo para entrada de data e hora

```
<form>
  <label>Data de Nascimento</label>
```

```
<input type="datetime-local" name="dtNascimento">
</form>
```

Data de Nascimento

novembro de 2015

<
 •
 >

dom	seg	ter	qua	qui	sex	sáb
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

date – campo para entrada de data

```
<form>
  <label>Data</label>
  <input type="date" name="data">
</form>
```

Data de Nascimento

novembro de 2015

<
 •
 >

dom	seg	ter	qua	qui	sex	sáb
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

month – campo para entrada de mês e ano

```
<form>
  <label>Mês e ano</label>
  <input type="month" name="mes">
</form>
```

Mês de

novembro de 2015

<
 •
 >

dom	seg	ter	qua	qui	sex	sáb
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

week – campo para entrada de dia da semana, dia do mês e ano

```
<form>
  <label>Semana, dia e ano</label>
  <input type="week" name="semana">
</form>
```

Mês e ano

novembro de 2015

Semana	dom	seg	ter	qua	qui	sex	sáb
45	1	2	3	4	5	6	7
46	8	9	10	11	12	13	14
47	15	16	17	18	19	20	21
48	22	23	24	25	26	27	28
49	29	30	1	2	3	4	5

time – Campo para entrada de hora

```
<form>
  <label>Hora</label>
  <input type="time" name="hora">
</form>
```

Hora

datetime – Campo para entrada de datas e horas gerais, não possui pré formatação

```
<form>
  <label>Datas e Horas</label>
  <input type="datetime" name="datahora">
</form>
```

Datas e Horas

Propriedades para os campos datetime		
name	Nome do campo	name="dataHora"
disabled	Desabilita o campo não podendo ser editado	disabled="disabled" ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form="id do formulário pai"
autocomplete	Permite que o campo seja auto completado ou não	autocomplete="on" ou "off"
autofocos	Posicione o cursor no elemento automaticamente ao abrir a página	autofocos="autofocus" ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength="50"
pattern	Especifica uma expressão para validação do campo	pattern="\d{5}"
readonly	Especifica que o campo é "somente leitura", não permite editar o conteúdo	readonly="readonly" ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required="required" ou required
size	Tamanho visível do elemento	size="30"
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder="Digite seu nome aqui"
value	Valor inicial do campo	value="jogador1"
list	Especifica uma lista pelo ID que contém os valores para o campo	list="id da lista"
min	Valor mínimo do campo	min="valor mínimo"
max	Valor máximo permitido no campo	max="valor máximo"
step	Passo de incremento ou decremento do campo	step="valor do passo"

number

Campo especial para entrada de números pode ser configurado como o modelo a seguir.

```
<form>
  <label>Valor</label>
  <input type="number" name="valor" min="1" max="12" step="2">
</form>
```

min="1" → Valor mínimo do campo.

max="12" → Valor máximo admitido no campo.

step="2" → Incremento ou decremento, será aumentado ou diminuído de dois em dois.

Valor

Propriedades para number		
name	Nome do campo	name="numero"
disabled	Desabilita o campo não podendo ser editado	disabled="disabled" ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form="id do formulário pai"
autocomplete	Permite que o campo seja auto completado ou não	autocomplete="on" ou "off"
autofocos	Posicione o cursor no elemento automaticamente ao abrir a página	autofocos="autofocus" ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength="50"
pattern	Especifica uma expressão para validação do campo	pattern="\d{5}"
readonly	Especifica que o campo é "somente leitura", não permite editar o conteúdo	readonly="readonly" ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required="required" ou required
size	Tamanho visível do elemento	size="30"
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder="Digite seu nome aqui"
value	Valor inicial do campo	value="jogador1"
list	Especifica uma lista pelo ID que contém os valores para o campo	list="id da lista"
min	Valor mínimo do campo	min="5"
max	Valor máximo permitido no campo	max="50"
step	Passo de incremento ou decremento do campo	step="2"

range

Campo de valor, porém possui uma interface diferente onde selecionamos o valor por uma barra deslizante.

```
<form>
  <label>Valor</label>
  <input type="range" name="barraValor" value="10" min="1" max="100" step="1">
</form>
```

value="10" → Valor inicial.

min="1" → Valor mínimo do campo.

max="12" → Valor máximo admitido no campo.

step="1" → Incremento ou decremento, será aumentado ou diminuído de um em um.

Valor

Propriedades para range		
name	Nome do campo	name="valorRange"
disabled	Desabilita o campo não podendo ser editado	disabled="disabled" ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form="id do formulário pai"
autocomplete	Permite que o campo seja auto completado ou não	autocomplete="on" ou "off"
autofocos	Posicione o cursor no elemento automaticamente ao abrir a página	autofocos="autofocus" ou autofocus
maxlength	Número máximo de elementos que podem ser inseridos no campo	maxlength="50"
pattern	Especifica uma expressão para validação do campo	pattern="\d{5}"
readonly	Especifica que o campo é "somente leitura", não permite editar o conteúdo	readonly="readonly" ou readonly
required	Especifica que o campo é requerido, tornando-o obrigatório	required="required" ou required
size	Tamanho visível do elemento	size="30"
placeholder	Texto (hint) de exemplo não editável é apagado ao inserir o valor no campo	placeholder="Digite seu nome aqui"
value	Valor inicial do campo	value="jogador1"
list	Especifica uma lista pelo ID que contém os valores para o campo	list="id da lista"
min	Valor mínimo do campo	min="5"
max	Valor máximo permitido no campo	max="50"

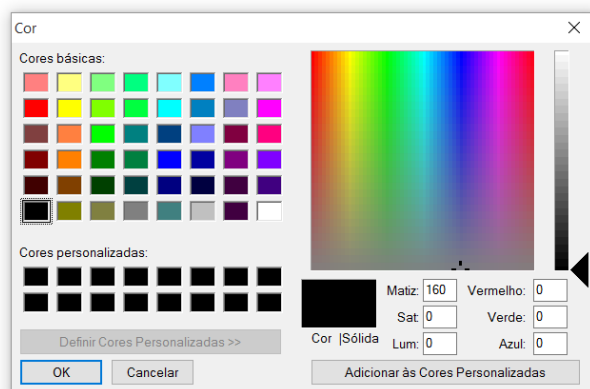
step	Passo de incremento ou decremento do campo	step="2"
------	--	----------

color

Campo pré configurado para seleção de cor “hexadecial”, ao clicar no campo é aberta a janela para seleção da cor conforme mostra a ilustração.

```
<form>
  <label>Selecione uma cor</label>
  <input type="color" name="cor">
</form>
```

Selecione uma cor



Propriedades para color		
name	Nome do campo	name="cor"
disabled	Desabilita o campo não podendo ser editado	disabled="disabled" ou disabled
form	Permite especificar a qual formulário o elemento pertence através do ID	form="id do formulário pai"
autocomplete	Permite que o campo seja auto completado ou não	autocomplete="on" ou "off"
autofocos	Posicione o cursor no elemento automaticamente ao abrir a página	autofocos="autofocus" ou autofocus
value	Valor inicial do campo	value="jogador1"
list	Especifica uma lista pelo ID que contém os valores para o campo	list="id da lista"

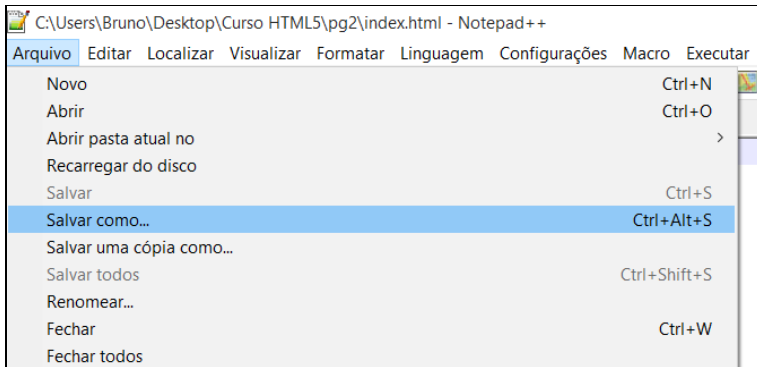
Este são os novos elementos de formulário adicionados no HTML5, os demais elementos, text, password, radio, checkbox, submit, reset, button, etc, continuam válidos.

Para aprender sobre a construção de formulários e os demais elementos não deixe de assistir as aulas no Canal Fessor Bruno.

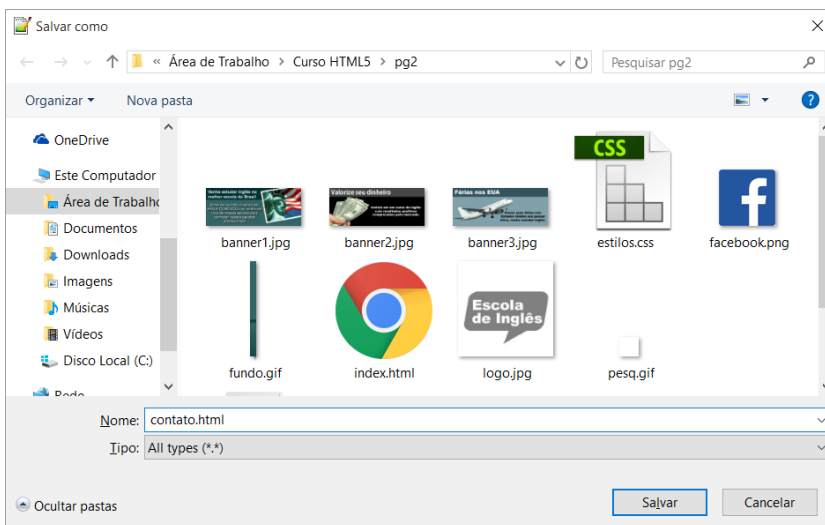
Criando o formulário para nosso site “Escola de Inglês”

Vamos continuar com a montagem no site “Escola de Inglês”, criaremos uma página a partir da página existente, adicionaremos o formulário e a salvaremos com outro nome.

Com a página anterior aberta no Bloco de Notas ou Notepad++ “index.html” clique no menu “Arquivo – Salvar Como”.



Salve com nome “contato.html”.



Precisamos apagar parte do código que não será usamos na página “contato.html”, apague o código desnecessário para que fique de acordo com o modelo a seguir, observe as alterações destacadas em vermelho.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <script src="scripts.js"></script>
  <title>Escola de Inglês</title>
</head>
<body>

  <header id="cab">
    
    <div id="social">
      
      
    </div>
    <nav id="menu">
      <ul>
        <li><a href="index.html" target="_self">Home</a></li>
        <li><a href="modulos.html" target="_self">Módulos</a></li>
        <li><a href="contato.html" target="_self">Contato</a></li>
      </ul>
    </nav>
  </header>

  <br class="fix">

  <section id="contato">

</section>

<br class="fix">
```

```
<footer>
  <p>Layout Curso de Inglês</p>
  <p>Canal Fessor Bruno</p>
  <p>www.youtube.com.br/canalfessorbruno</p>
</footer>

</body>
</html>
```

Agora vamos digitar os comandos para criação do formulário siga o modelo a seguir e digite todo conteúdo destacado em vermelho para criar o formulário.

```
<section id="contato">

  <form name="fContato" method="post" action="#">

    <label>Nome</label><br>
    <input type="text" maxlength="50" size="60" name="fNome" autofocus required><br><br>

    <label>Telefone de contato</label><br>
    <input type="tel" maxlength="15" size="20" name="fFone" placeholder="(xx) xxxx-xxxx"
required><br><br>

    <label>Email para contato</label><br>
    <input type="email" maxlength="50" size="60" name="fEmail" required><br><br>

    <label>Cep</label><br>
    <input type="text" maxlength="10" size="15" name="fCep" placeholder="xxxxx-xxx" pattern="\d{5}-
?\d{3}"required><br><br>

    <label>Data de Nascimento</label><br>
    <input type="date" name="fNascimento" required><br><br>

    <label>Módulo que pretende iniciar (sujeito a teste de conhecimento)</label><br>
    <input type="number" min="1" max="10" step="1" value="1" name="fModulo"><br><br>

    <label>Horário de início</label><br>
    <input type="time" name="fHorario" step="1800" value="10:00"><br><br>

    <label>Cor da pasta - Ao matricular você ganha uma pasta para guardar seu material</label><br>
    <input type="color" name="fCor" value="#008800"><br><br>

    <input type="submit" value="Enviar">
    <input type="reset" value="Limpar">

  </form>

</section>
```

Salve o arquivo, atualize a página e veja o resultado conforme a ilustração a seguir.

Escola de Inglês

Home Módulos Contato

Nome

Telefone de contato

Email para contato

Cep

Data de Nascimento

Módulo que pretende iniciar (sujeito a teste de conhecimento)

Horário de início

Cor da pasta - Ao matricular você ganha uma pasta para guardar seu material

Enviar Limpar

Layout Curso de Inglês Canal Fessor Bruno www.youtube.com.br/canalfessorbruno

Pronto, com o formulário devidamente criado precisamos formatá-lo, então vamos ao CSS, adicione os comandos no arquivo “estilo.css”.

```
#contato label{
    font-size:15pt;
    font-weight:bold;
    font-style: italic;
    font-family: Arial;
    color:#777;
}

#contato input{
    font-size:15pt;
    color:#000;
}
```

#contato label{ → Formatação dos elementos <label>

font-size:15pt; → Tamanho da fonte

font-weight:bold; → Negrito

font-style: italic; → Itálico

font-family: Arial; → Tipo de fonte Arial

color:#777; → Cor do texto

}

#contato input{ → formatação dos elementos <input>

font-size:15pt; → Tamanho da fonte



color:#000; → Cor da fonte

}

Agora vamos formatar os botões, iremos criar uma classe e adicionar aos dois botões, adicione os comandos em vermelho na sua página HTML

```
<input type="submit" value="Enviar" class="botao">
<input type="reset" value="Limpar" class="botao">
```

Agora para formatar vamos ao CSS.

```
.botao{
    width:200px;
    height:40px;
    border-radius:20px;
    background-color:#669999;
    border:0px;
    cursor:pointer;
}
```

.botao{ → Classe botao

width:200px; → Largura do elemento

height:40px; → Altura do elemento

border-radius:20px; → Arredondamento da borda

background-color:#669999; → Cor de fundo

border:0px; → Largura da borda

cursor:pointer; → Cursor do mouse quando estiver sobre o botão

}

Salve as alterações e atualize a página para ver os resultados.

Escola de Inglês

Home Módulos Contato

Nome

Telefone de contato

Email para contato

Cep

Data de Nascimento

Módulo que pretende iniciar (sujeito a teste de conhecimento)

Horário de início

Cor da pasta - Ao matricular você ganha uma pasta para guardar seu material

Enviar Limpar

Agora que terminamos a página de formulário vamos para última página deste site, a página “modulos.html”.

A partir da página “contato.html” vamos criar a página “modulos.html”, com a página de formulário aberta, clique no menu “Arquivo – Salvar Como”, salve esta nova página com o nome “modulos.html”, com a nova página criada vamos para o próximo capítulo.

Pattern para validação dos formulários

Os inputs dos formulários agora contam com uma propriedade interessante que se chama pattern, basicamente definimos uma expressão para que o valor do campo seja validado, para verificar se o valor do campo está no formato correto ou não, muito interessante para garantirmos a fidelidade dos dados de um formulário.

Vamos entender através de exemplos como criar um “pattern” para validação, para isto vamos criar um simples formulário para os testes, a seguir confira algumas opções de pattern.

[A-Za-z]{5} → Este “pattern” valida o campo com qualquer letra, maiúscula ou minúscula e o máximo 5 caracteres, não permitindo caracteres diferentes de letras de a até z, números, pontos, caracteres especiais e nem acentuação não são permitidos.

```
<form name="fPattern" id="fPattern" method="post" action="#">

    <input type="text" name="fTeste" size="30" pattern="[A-Za-z]{5}">
    <input type="submit" value="Testar">

</form>
```

.{8,} → Permite que sejam inseridos oito caracteres ou mais, não serão aceitos textos com menos de 8 caracteres, neste caso permite qualquer caractere.

```
<input type="text" name="fTeste" size="30" pattern=".{8,}">
```

[0-9]+\$ → Precisa de um campo que aceite somente números? Então use este pattern que só valida numerais entre 0 e 9.

```
<input type="text" name="fNumeros" size="50" pattern="[0-9]+$">
```

[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}\$ → Esta validação parece bastante complicada, mas não é, veja a explicação. É um pattern para validação de e-mails, verifica se o formato do email está correto. Vou explicar parte a parte.

[a-z0-9._%+-] → Esta primeira parte permite que sejam usados qualquer tipo de caractere no prefixo do email, antes do @.

+@[a-z0-9.-] → A segunda parte diz que o prefixo, primeira parte, “deverá” ser seguida de um @ e um texto com uma série de caracteres aceitáveis, note que a primeira parte aceita mais tipos de caracteres do que a segunda, o caractere mais “+” no início diz que obrigatoriamente este texto deverá vir na sequência do anterior.

+\.[a-z]{2,3}\$ → A terceira parte diz respeito ao tipo de domínio, .com, .com.br, etc, então aqui validados que poderão conter qualquer letra de a até z minúsculas e diz que cada palavra deverão conter duas ou três letras.

```
<input type="email" name="fmail" size="50" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$">
```

[^'\x22]+ → Este pattern não permite que sejam inseridos os caracteres apóstrofo ‘ e nem aspas “ no texto.

```
<input type="text" name="fTexto" size="50" pattern="^[^'\x22]+$">
```



https?://.+ → Esta é uma validação para URLs, verifica se o conteúdo da é precedido por http ou https, só é permitido texto com estes caracteres.

```
<input type="url" name="fSite" size="50" pattern="https?://.+">
```

\d{3}\.\d{3}\.\d{3}-\d{2}}|(\d{11})\$ → Precisa de um validador para CPF? Então está aqui, este pattern permite três sequências numéricas (d) separadas por pontos e no final um traço e mais dois números, ainda verifica se existem onze dígitos numéricos.

```
<input type="text" name="fCPF" size="50" pattern="\d{3}\.\d{3}\.\d{3}-\d{2}}|(\d{11})$">
```

[0-9]{2}/[0-9]{2}/[0-9]{4}\$ → Este pattern faz validação de datas associado aos demais comandos max, min, maxlength conforme o código a seguir, [0-9]{2}\ (dois dígitos para o dia entre 0 e 9) /[0-9]{2}\ (/ dois dígitos entre 0 e 9 para o mês) /[0-9]{4}\$ (e finalmente / e quatro dígitos para o ano).

```
<input type="date" maxlength="10" name="fDate" pattern="[0-9]{2}/[0-9]{2}/[0-9]{4}$" min="2012-01-01" max="2014-02-18">
```

[0-9]{2}:[0-9]{2} [0-9]{2}\$ → Podemos validar hora também, [0-9]{2}: (dois dígitos numéricos seguidos de dois pontos) [0-9]{2} (mais dois dígitos numéricos seguidos de espaço) [0-9]{2}\$ (por último mais dois dígitos numéricos correspondentes aos segundos)

```
<input type="time" maxlength="8" name="fHora" pattern="[0-9]{2}:[0-9]{2} [0-9]{2}$">
```

\([0-9]{2}\) [0-9]{4,6}-[0-9]{3,4}\$ → Validação para campos de telephone, \([0-9]{2}\) (Dois dígitos numéricos entre parênteses) [0-9]{4,6}- (seguido de quatro dígitos e um traço) [0-9]{3,4}\$ (finalizando com três ou quatro dígitos numéricos).

```
<input type="tel" maxlength="15" name="fTelefone" pattern="\([0-9]{2}\) [0-9]{4,6}-[0-9]{3,4}$">
```

([0-9]{1,3}\.)?[0-9]{1,3},[0-9]{2}\$ → Validação para formato monetário, moeda.

```
<input type="text" name="fMoeda" pattern="([0-9]{1,3}\.)?[0-9]{1,3},[0-9]{2}$">
```

Aprendemos várias formas de validação de dados, você pode criar sua própria validação seguindo como base as validações anteriores, não é um bicho de sete cabeças.

Detalhes e Sumário <details> <summary>

Outro recurso interessante de HTML5 são os elementos <details> e <summary> que trabalham em conjunto para formar um elemento interativo bem interessante do tipo elemento expansível.

Vamos deletar o código do formulário, deixe o código da página como o modelo a seguir, note a mudança destacada em vermelho.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <script src="scripts.js"></script>
  <title>Escola de Inglês</title>
</head>
```



```
<body>

  <header id="cab">
    
    <div id="social">
      
      
    </div>
    <nav id="menu">
      <ul>
        <li><a href="index.html" target="_self">Home</a></li>
        <li><a href="modulos.html" target="_self">Módulos</a></li>
        <li><a href="contato.html" target="_self">Contato</a></li>
      </ul>
    </nav>
  </header>

  <br class="fix">

  <section id="modulos">

</section>

  <br class="fix">

  <footer>
    <p>Layout Curso de Inglês</p>
    <p>Canal Fessor Bruno</p>
    <p>www.youtube.com.br/canalfessorbruno</p>
  </footer>

</body>
</html>
```

Vamos adicionar o conteúdo principal desta página, adicione todo código destacado em vermelho do modelo a seguir, logo em seguida vou explicar este código.

```
<section id="modulos">

  <details>
    <summary>Módulo 1 - Aprendendo o básico</summary>
    <p>No primeiro módulo iremos aprender o básico necessário para alinhamento da turma e para que
passe para o próximo nível.</p>
    <p>Embora pareça um módulo com conteúdo básico, a importância deste módulo é enorme, se você já
tem um conhecimento básico em inglês aproveite para melhorar este conhecimento neste módulo.</p>
  </details>
  <details>
    <summary>Módulo 2 - Inglês intermediário</summary>
    <p>No segundo módulo se prepare para novos horizontes</p>
    <p>Neste módulo vamos iniciar as conversações, temos um conteúdo mais elaborado e devidamente
estudado para desenvolver todo seu potencial em Inglês.</p>
  </details>
  <details>
    <summary>Módulo 3 - Avançado</summary>
    <p>O terceiro módulo é focado em conversação avançada</p>
    <p>Nosso objetivo nesta parte do curso é capacitar o aluno a se virar em qualquer tipo de conversa
em Inglês.</p>
  </details>

</section>
```

A explicação sobre <details> é simples, basicamente tem a função de mostrar ou esconder informações extras sobre um conteúdo, quando usamos o elemento <details> é interessante que usemos a tag <summary> que é o título deste <details>, basicamente é o texto que será exibido sempre, ao clicar neste título o conteúdo será expandido ou retraído.

<details> → Podemos entender como o container para o conteúdo do tipo mostra/esconde.

<summary> → É o sumário/título para o conteúdo que será mostrado ou escondido.



Caso deseje que a lista já se inicie aberta basta usar o parâmetro open dentro da tag details, desta forma mostrada a seguir.

<details open>

Itens retraídos/ocultos

- ▶ Módulo 1 - Aprendendo o básico
 - ▶ Módulo 2 - Inglês intermediário
 - ▶ Módulo 3 - Avançado

Itens expandidos/exibidos

▼ Módulo 1 - Aprendendo o básico

No primeiro módulo iremos aprender o básico necessário para alinhamento da turma e para que passe para o próximo nível.

Embora pareça um módulo com conteúdo básico, a importância deste módulo é enorme, se você já tem um conhecimento básico em inglês aproveite para melhorar este conhecimento neste módulo.

▼ Módulo 2 - Inglês intermediário

No segundo módulo se prepare para novos horizontes

Neste módulo vamos iniciar as conversações, temos um conteúdo mais elaborado e devidamente estudado para desenvolver todo seu potencial em Inglês.

▼ Módulo 3 - Avançado

O terceiro módulo é focado em conversação avançada

Nosso objetivo nesta parte do curso é capacitar o aluno a sevirar em qualquer tipo de conversa em Inglês.

Abaixo do <summary> podemos adicionar qualquer elemento, imagens, tabelas, formulários, divs, etc.

Vamos à formatação.

Adicione os comandos a seguir no arquivo CSS e logo a seguir vamos às explicações.

```
details summary::-webkit-details-marker {
    display: none;
}

details{
    margin-top:30px;
}

summary{
    height: 40px;
    border-radius:10px;
    padding: 15px 0px 0px 10px;
    font: 15pt Arial;
    font-weight:bold;
    background: linear-gradient(to bottom, #EEE, #999);
}

details p{
    font-size: 20px;
}

summary:before{
    margin-right:5px;
    content:"+";
}

details[open] summary:before {
    content: "-";
}
```

```
details summary::-webkit-details-marker { → Este comando irá resetar as configurações padrões do <details>
    display: none;
}
```

```
details { → Formatação básica da tag <details>
    margin-top: 30px; → Margem superior de 30px em todos os <details>
    cursor: pointer; → Cursor da mão.
}
```

```
summary { → Formatação padrão dos títulos <summary>
    height: 40px; → Altura dos <summary>
    border-radius: 10px; → Arredondamento das bordas dos <summary>
    padding: 15px 0px 0px 10px; → Margens internas
    font: 15pt Arial; → Formatação da fonte, tamanho e tipo de fonte
    font-weight: bold; → Negrito
    background: linear-gradient(to bottom, #EEE, #999); → Preenchimento gradiente do fundo
}
```

```
details p { → Formatação dos parágrafos <p> do <details>
    font-size: 20px; → Tamanho da fonte
}
```

```
summary:before { → Formatação do elemento “anterior” do <summary> (elemento que vai substituir a seta)
    margin-right: 5px; → Margem direita
    content: “+”; → Substituir a seta por “+”.
}
```

```
details[open] summary:before { → Formatação do elemento à direita (seta) quando estiver aberto
    content: “-”; → Quando estiver aberto troca para o “-”
}
```

Salve as alterações e atualize a página.



Terminamos nosso segundo site, vimos uma série de novidades de HTML5, mas ainda restam algumas novidades a serem vistas ainda, tome um fôlego novo e vamos continuar nosso aprendizado.

Transparência

Um dos novos recursos de CSS3 é a possibilidade de trabalhar com transparências, canal alpha, é bem simples e você vai aprender este recurso a seguir.

Crie uma nova pasta chamada “transparência”, crie um novo arquivo e o salve nesta pasta com nome “transp.html”, neste arquivo digite o código a seguir.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <title>Transparência</title>
</head>
<body>

  <h1>Canal Fessor Bruno</h1>
  <div id="cx"></div>

</body>
</html>
```

Note que temos um <h1> e uma <div> no corpo da página, vamos formatar a <div> de forma que ela fique sobre o texto <h1>, então, crie um novo arquivo e o salve com nome “estilos.css” digite os comandos de formatação da <div> mostrados a seguir.

```
#cx{ position:absolute;
top:0px;
left:0px;
width:200px;
height:100px;
background: #FF0000;
}
```

Salve e abra a página HTML para ver que a <div> está sobre o <h1> como mostra o exemplo a seguir.



Vamos aprender sobre a propriedade “rgba” para trabalhar com transparência.

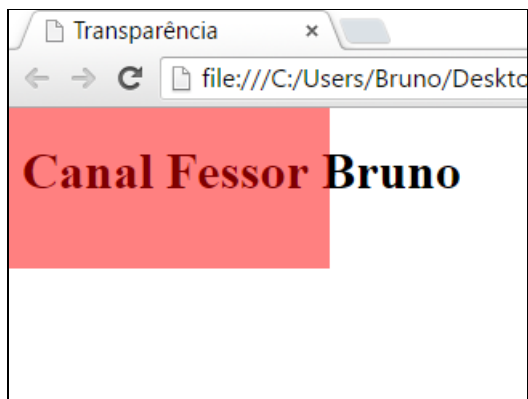
rgba(vermelho, verde, azul, alpha);

Os valores para os canais vermelho, verde e azul recebem valores decimais com variação de 0 a 255, o último valor “alpha” é a definição de transparência que varia de 0 a 1, sendo zero sem transparência e 1 com transparência total, para meia transparência usamos 0.5.

Vamos configurar nossa <div> com transparência de 50%, troque o comando “background: #FF0000;” para o comando a seguir.

```
background: rgba(255,0,0,0.5);
```

rgba(255,0,0,0.5); → Vermelho no máximo 255, nada de verde e nada de azul, transparência configurada em 50%.



Salve e atualize a página para ver o resultado, para aumentar ou diminuir a transparência altere o valor 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, por exemplo.

Gradiente

Um recurso de formatação interessantíssimo presente no CSS3 é a possibilidade de trabalhar com gradientes, esse recurso deixa nossos projetos mais leves já que pode substituir o uso de imagens com uma facilidade muito grande.

Crie uma nova pasta com nome “gradiente”, crie um novo arquivo e insira o código básico a seguir.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <title>Aprendendo gradientes</title>
</head>
<body>

  <div id="g1" class="grad"></div><br>

</body>
</html>
```

Salve este arquivo com nome “gradientes.html” e crie um novo arquivo com nome “estilos.css” neste novo arquivo de estilos vamos criar os gradientes configurando o parâmetro background (cor de fundo).

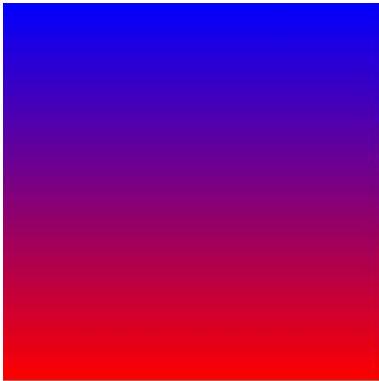
```
.grad{
  width:300px;
  height:300px;
}

#g1{
  background: linear-gradient(#0000FF, #FF0000);
}
```

linear-gradient (variação linear)

```
#gl{
  background: linear-gradient(#0000FF, #FF0000);
}
```

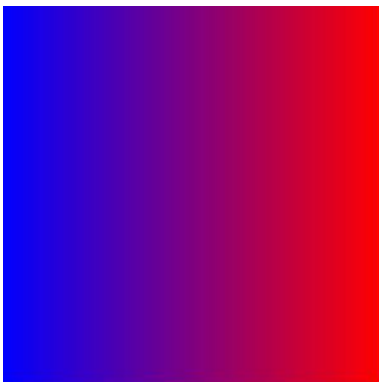
Linear-gradient(#0000FF, #FF0000); → Gradiente linear, a variação padrão e na vertical, nesta configuração com duas cores do azul (#0000FF) para o vermelho (#FF0000).



Vou mostrar outras opções de configuração de gradientes, você pode ir testando alterando o CSS e visualizando a alteração.

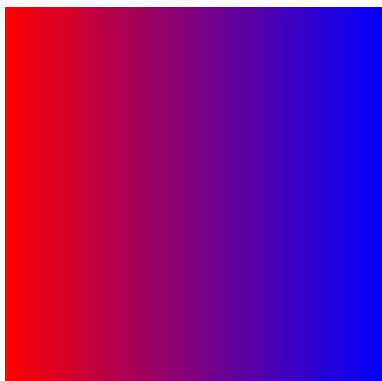
```
#gl{
  background: linear-gradient(to right, #0000FF , #FF0000);
}
```

Variação para direita (to right), de azul para vermelho.



```
#gl{
  background: linear-gradient(to left, #0000FF , #FF0000);
}
```

Variação para esquerda (to-left), cores azul e vermelho.



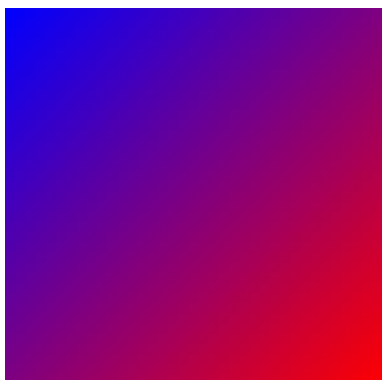
```
#gl{
  background: linear-gradient(to top, #0000FF , #FF0000);
}
```

Variação para cima (to-top), cores azul e vermelho.



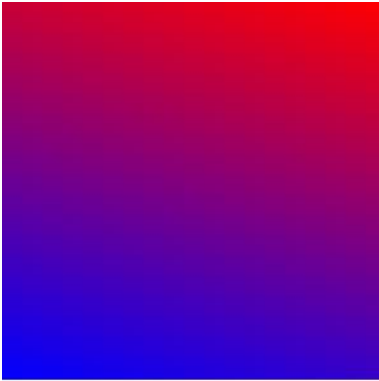
```
#gl{
  background: linear-gradient(to bottom right, #0000FF , #FF0000);
}
```

Variação diagonal para baixo a direita (to bottom right), cores azul e vermelho.



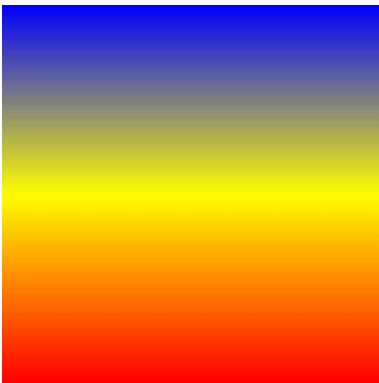
```
#gl{
  background: linear-gradient(15deg, #0000FF, #FF0000);
}
```

Variação em 15 graus, cores azul e vermelho.



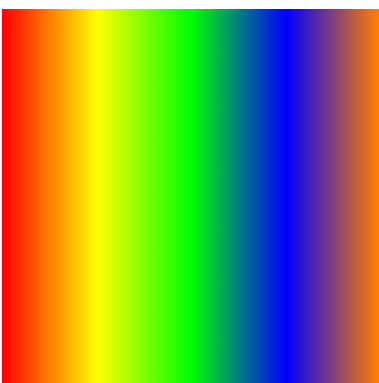
```
#gl{
  background: linear-gradient(#0000FF, #FFFF00, #FF0000);
}
```

Variação padrão (para baixo), três cores azul, amarelo e vermelho.



```
#gl{
  background: linear-gradient(to right, #FF0000, #FFFF00, #00FF00, #0000FF, #FF8000);
}
```

Variação para direita (to right), cinco cores, vermelho, amarelo, verde, azul e laranjado.



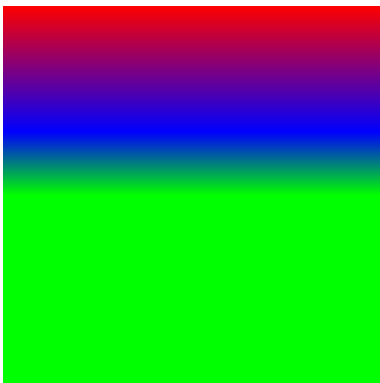
```
#gl{
  background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
}
```

Variação para direita (to right) com transparência vermelho 0% de opacidade para vermelho 100% de opacidade.



```
#gl{
  background: linear-gradient(to bottom, #FF0000 0px, #0000FF 100px, #00FF00 150px);
}
```

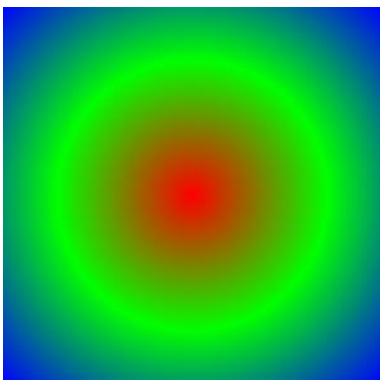
Variação para baixo (to bottom), três cores em posições pré definidas, vermelho em 0px, azul em 100px e verde em 150px.



radial-gradient (variação circular)

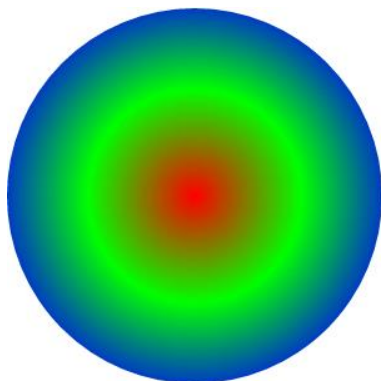
```
#gl{
  background: radial-gradient(#FF0000, #00FF00, #0000FF);
}
```

Variação radial, três cores, vermelho, verde e azul.



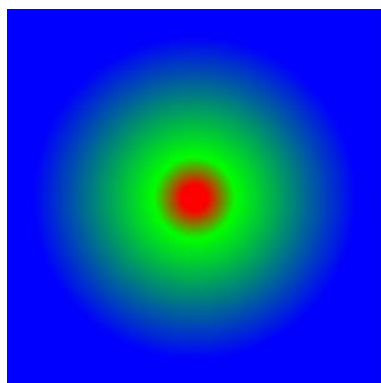
```
#gl{
  background: radial-gradient(#FF0000, #00FF00, #0000FF 80%);
  border-radius: 100%;
}
```

Variação radial, três cores com uma posição definida, vermelho, verde, azul em 80%, ainda com parâmetro para arredondamento dos cantos.



```
#gl{
  background: radial-gradient(#FF0000 5%, #00FF00 15%, #0000FF 60%);
}
```

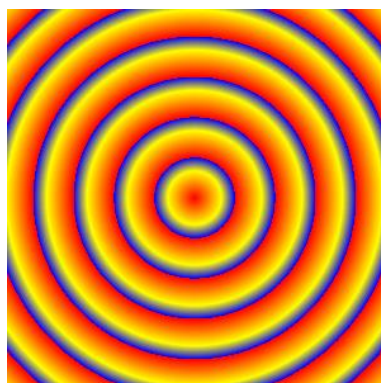
Variação radial, três cores em posições definidas, vermelho em 5%, verde em 15% e azul em 60%.



Repetição de gradientes

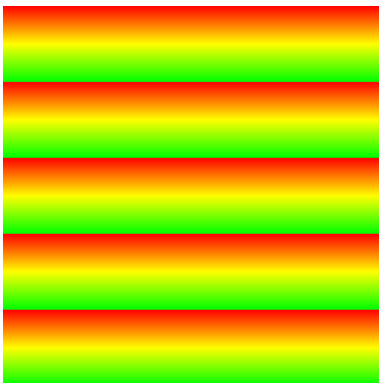
```
#gl{
  background: repeating-radial-gradient(#FF0000, #FFFF00 10%, #0000FF 15%);
}
```

Variação circular com repetição, três cores duas com posições definidas, vermelho, amarelo em 10% e azul em 15%.



```
#gl{
  background: repeating-linear-gradient(#FF0000, #FFFF00 10%, #00FF00 20%);
}
```

Variação linear com repetição, três cores duas com posições definidas, vermelho, amarelo em 10% e verde em 20%.



Áudio e Vídeo

HTML5 trouxe novas tags para trabalharmos com áudio e vídeo, são elementos bem simples e funcionais que cumprem bem seu objetivo, ambas tem a sintaxe bem parecida.

Vamos começar com a tag <audio>, como o próprio nome diz, esta tag serve para inserir sons em nossas páginas, veja um exemplo a seguir de utilização da tag <audio>.

Veja um exemplo de uso da tag <audio>

```
<audio controls="true" autoplay="true">
  <source src="musica.oga">
  <source src="musica.mp3">
  <source src="musica.wma">
  <p>Seu navegador não tem suporte a HTML5</p>
</audio>
```

//Inserir áudio/música na sua página, o parâmetro “controls” com valor “true” mostra a barra com comandos play, pause, volume, etc, caso não queira mostrar os controle basta configurar controls=“false”. Já o parâmetro “autoplay” configurado em “true” faz com que o áudio seja executado automaticamente, caso não queira que o áudio seja reproduzido automaticamente basta configurar autoplay=“false”.

```
<audio controls="true" autoplay="true">
```

//A tag <source> serve para que possamos oferecer várias opções de formato de arquivo com o mesmo áudio, caso não consiga executar um formato, por causa de plug-ins ou outra causa, automaticamente é tentado outro formato, você pode especificar quantos formatos desejar, desde que tenha os arquivos é claro.

```
<source src="musica.oga">
<source src="musica.mp3">
<source src="musica.wma">
```

//Este texto no final será mostrado caso o browser não possuir suporte a HTML5 ou a tag <audio>

```
<p>Seu navegador não tem suporte a HTML5</p>
```

Vamos entender a tag <vídeo> que é bem parecida.

```
<video controls="true" autoplay="true" width="600" height="300">
  <source src="video.ogv">
  <source src="video.mp4">
  <source src="video.wmv">
```

```
<p>Seu navegador não tem suporte a HTML5</p>
</video>
```

//Inserir vídeo na sua página, o parâmetro “controls” com valor “true” mostra a barra com comandos play, pause, volume, etc, caso não queira mostrar os controle basta configurar controls=“false”. Já o parâmetro “autoplay” configurado em “true” faz com que o vídeo seja executado automaticamente, caso não queira que o vídeo seja reproduzido automaticamente basta configurar autoplay=“false”.

Para configurar o tamanho da janela usamos os parâmetros width (largura) e height(altura).

```
<video controls="true" autoplay="true" width="600" height="300">
```

//A tag <source> serve para que possamos oferecer várias opções de formato de arquivo com o mesmo vídeo, caso não consiga executar um formato, por causa de plug-ins ou outra causa, automaticamente é tentado outro formato, você pode especificar quantos formatos desejar, desde que tenha os arquivos é claro.

```
<source src="musica.oga">
```

```
<source src="video.ogv">
```

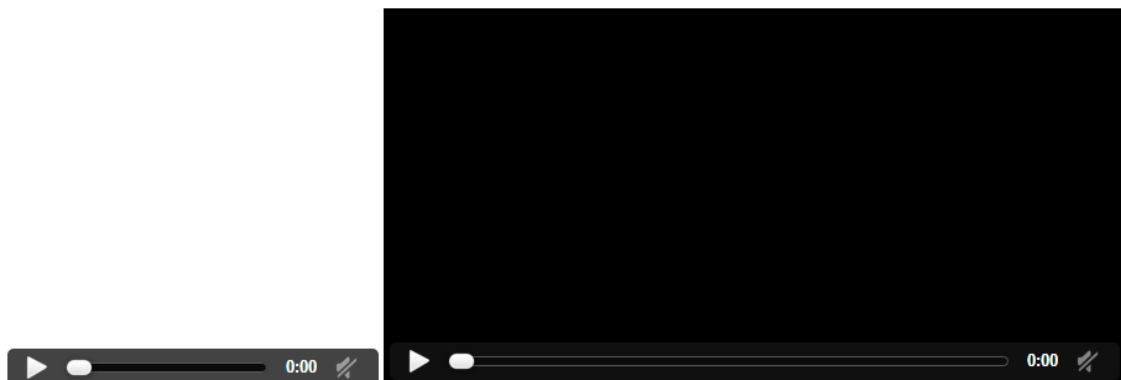
```
<source src="video.mp4">
```

```
<source src="video.wmv">
```

//Este texto no final será mostrado caso o browser não possuir suporte a HTML5 ou a tag <video>

```
<p>Seu navegador não tem suporte a HTML5</p>
```

Confira na ilustração a seguir, à esquerda o controle de áudio e a direita o janela com os controle de vídeo.



SVG – Gráficos vetoriais direto pelo browser

SVG significa “Scalable Vector Graphics”. É um novo recurso de HTML5 para criação de gráficos vetoriais diretamente pelo browser, basicamente é usado para definir gráficos para a Web.

É importante entender bem SVG para que possamos “desenhar” pelo browser e substituir arquivos de imagens sempre que possível, para o browser e para o carregamento web é bem mais simples e leve usar SVG do que baixar uma imagem, então sempre que possível prefira usar SVG.

Uma das melhores vantagens de usar imagens SVG é quanto a qualidade, não importa o quanto você aplique de zoom ou aumente a imagem, ela nunca perderá qualidade, sempre estará sendo exibida em qualidade total sem desfoque.

Vamos entender como trabalhar com gráficos XML SVG.

A primeira tarefa é definir a “área de trabalho” para os gráficos, definimos esta área com a tag <svg> como mostrado a seguir.

```
<svg width="500" height="500" id="svgxml">

  <!-- Aqui montamos os gráficos -->

</svg>
```

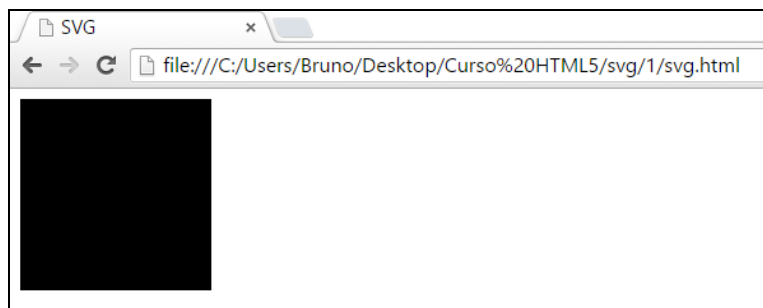
No código acima criamos uma área para gráficos SVG de tamanho 500x500 pixels, dentro desta área podemos adicionar quantos gráficos desejarmos, vamos adicionar um quadrado nesta área conforme o código destacado em vermelho a seguir.

```
<svg width="500" height="500" id="svgxml">

  <rect width="150" height="150"/>

</svg>
```

`<rect width="150" height="150">` ➔ Com este comando desenhamos um retângulo com largura (width) e altura (height) de 150 pixels.

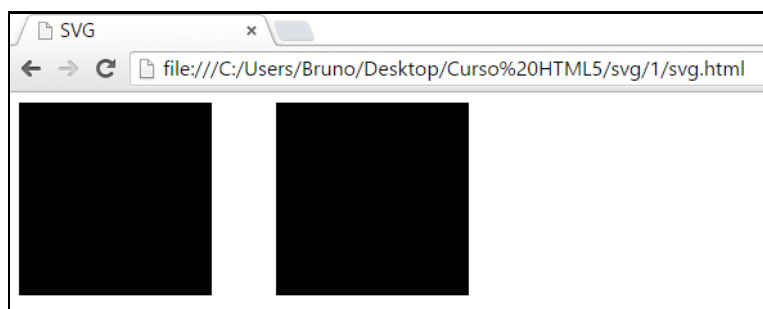


Vamos criar um segundo quadrado ao lado do primeiro.

```
<svg width="500" height="500" id="svgxml">

  <rect width="150" height="150"/>
  <rect width="150" height="150" x="200" y="0"/>

</svg>
```



`<rect width="150" height="150" x="200" y="0"/>` ➔ Este Segundo quadrado tem o mesmo tamanho do primeiro largura (width) e altura (height) de 150px, porém, note que definimos os parâmetros x e y para determinar a posição do quadrado dentro da área <svg>.

Dentro da área delimitada pela tag <svg> podemos inserir quantas imagens forem necessárias, em teoria podemos inserir fora da área também só que não serão visíveis.

Vamos diminuir a largura do <svg> e ver o que acontece, altere seu código conforme o destaque em vermelho a seguir.

```
<svg width="250" height="500" id="svgxml">

    <rect width="150" height="150"/>
    <rect width="150" height="150" x="200" y="0"/>

</svg>
```

Salve e atualize a página, veja que o segundo quadrado ficará cortado, pois, excede o tamanho do <svg>



Vamos formatar o <svg> e os <rect> por CSS, crie o arquivo “sgvestilos.css” e anexe-o ao arquivo conforme o código em vermelho a seguir.

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="sgvestilos.css">
    <title>SVG</title>
</head>
<body>

    <svg width="250" height="500" id="svgxml">

        <rect width="150" height="150"/>
        <rect width="150" height="150" x="200" y="0"/>

    </svg>

</body>
</html>
```

Remova as formatações das tags <svg> e <rect> conforme o modelo a seguir e adicione os comandos em vermelho.

```
<svg id="svgxml">

    <rect id="r1"/>
    <rect id="r2"/>

</svg>
```

No arquivo CSS recém criado, adicione os comandos a seguir para formatação dos elementos.

```
#svgxml{
    background: #EEE;
    width:500px;
    height:500px;
}

#r1{
    width:150px;
    height:150px;
    fill:#F00;
    x:0px;
    y:0px;
```

```
}

#r2{
  width:150px;
  height:150px;
  fill:#00F;
  x:200px;
  y:0px;
}
```

Quase nada de novo nestas formatações, width, height, background, já conhecemos estes parâmetros, vou descrever as novidades.

x:0px; → Posicionamento horizontal, eixo X.

y:0px; → Posicionamento vertical, eixo Y.

fill:#F00; → O parâmetro fill é usado nos elementos de <svg> para indicar a cor de preenchimento, neste caso a cor vermelho.

fill:#00F; → Aqui usamos a cor azul para preencher.

Salve as alterações e compare com a ilustração a seguir.



Vamos desenhar outras formas. Adicione os comandos em vermelho conforme o código a seguir.

```
<svg id="svgxml">

  <rect id="r1"/>
  <rect id="r2"/>
  <circle id="c1"/>
  <ellipse id="e1"/>
  <text fill="#888" font-size="45" font-family="Verdana" x="20" y="450">
    Canal Fessor Bruno
  </text>

</svg>
```

<circle id="c1"/> → Círculo

<ellipse id="e1"/> → Elipse, se diferencia do círculo por permitir que a largura seja diferente da altura

<text fill="#888" font-size="45" font-family="Verdana" x="20" y="450"> → Texto

Canal Fessor Bruno

</text>

Com os elementos adicionados, vamos adicionar as formatações necessárias no arquivo CSS.

```
#c1{
  cx:85px;
  cy:275px;
  r:75px;
  stroke:#080;
  stroke-width:10;
  fill:#FF0;
}

#e1{
  cx:275px;
  cy:275px;
  rx:85px;
  ry:55px;
  fill:#F0F;
}
```

```
#c1{
  cx:85px; → Posição X, horizontal
  cy:275px; → Posição Y, vertical
  r:75px; → Raio do círculo
  stroke:#080; → Cor da borda
  stroke-width:10; → Largura da borda
  fill:#FF0; → Cor do preenchimento.
}

#e1{
  cx:275px; → Posição X, horizontal
  cy:275px; → Posição Y, vertical
  rx:85px; → Largura da elipse
  ry:55px; → Altura da elipse
  fill:#F0F; → Cor de preenchimento
}
```

Salva as alterações e compare com a ilustração a seguir.



Vamos aprender mais um elemento importante, o polígono.

Substitua os elementos anteriores pelo polígono a seguir.

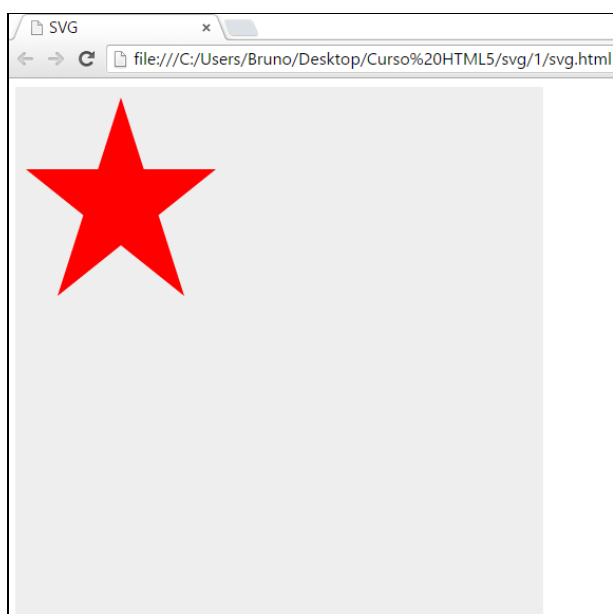
```
<svg id="svgxml">

  <polygon points="100,10 40,198 190,78 10,78 160,198" fill="red">

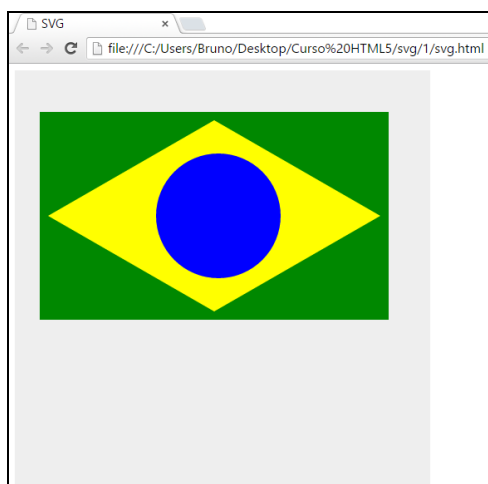
</svg>
```

`<polygon points="100,10 40,198 190,78 10,78 160,198" fill="red">` ➔ O element `<polygon>` é criado por pares de coordenadas (X,Y), no caso no nosso polígono as coordenadas são:

X	Y
100	10
40	198
190	78
10	78
160	198



Agora que já conhecemos os elementos básicos para desenho em `<svg>`, tente desenhar a bandeira do brasil como no modelo a seguir, logo após a imagem de como deve ficar a bandeira, vou disponibilizar o código HTML e CSS, mas tente fazer a bandeira antes de olhar o código.





HTML

```
<svg id="svgxml">

  <rect id="r1"/>
  <polygon id="p1" points="40,175 240,60 440,175 240,290"/>
  <circle id="c1"/>

</svg>
```

CSS

```
#svgxml{
  background: #EEE;
  width:500px;
  height:500px;
}

#r1{
  width:420px;
  height:250px;
  fill:#080;
  x:30px;
  y:50px;
}

#p1{
  fill:#FF0;
}

#c1{
  cx:245px;
  cy:175px;
  r:75px;
  fill:#00F;
}
```

Outro recurso bastante interessante que iremos aprender sobre gráficos SVG são os gradientes.

A princípio pode parecer complicado trabalhar com gradientes, mas praticando você verá que é bem simples na verdade, vamos substituir a bandeira por um quadrado com preenchimento gradiente, adicione os comandos a seguir no arquivo HTML.

```
<defs>
  <linearGradient id="grad1" x1="0%" y1="0%" x2="0%" y2="100%">
    <stop offset="0%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
    <stop offset="50%" style="stop-color:rgb(0,0,255);stop-opacity:1" />
    <stop offset="100%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
  </linearGradient>
</defs>
<rect x="50" y="50" width="400" height="400" fill="url(#grad1)">
```

Ficou meio perdido no código acima? Não se preocupe, vou explicar.

<defs> → Para criar o gradiente usamos a tag <defs>, aqui iremos definir nossos gradientes.

<linearGradient id="grad1" x1="0%" y1="0%" x2="0%" y2="100%"> → Nesta tag iremos definir as configurações básicas do nosso gradiente, posição inicial (x1,y1) e posição final (x2,y2), com esta configuração nosso gradiente tem a variação na vertical, de cima para baixo.

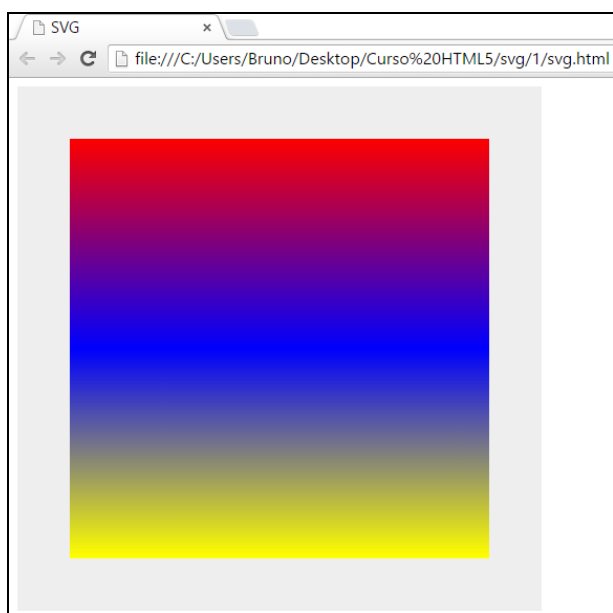
<stop offset="0%" style="stop-color:rgb(255,0,0);stop-opacity:1" /> → Primeira cor, offset define a posição, neste caso a primeira cor, posição 0%, em stop-color definimos a primeira cor como vermelho (255,0,0) e opacidade 100% (1).

`<stop offset="50%" style="stop-color:rgb(0,0,255);stop-opacity:1" />` → Segunda cor, offset define a posição, neste caso a segunda cor, posição 50% (metade), em stop-color definimos a segunda cor como azul (0,0,255) e opacidade 100% (1).

`<stop offset="100%" style="stop-color:rgb(255,255,0);stop-opacity:1" />` → Terceira cor, offset define a posição, neste caso a terceira e última cor, posição 100%, em stop-color definimos a terceira cor como amarelo (255,255,0) e opacidade 100% (1).

`<rect x="50" y="50" width="400" height="400" fill="url(#grad1)">` → Aqui criamos um retângulo e no parâmetro “fill” adicionamos o gradiente com id “grad1”, note que este ID foi configurado na tag `<linearGradient>`.

Salve as alterações e compara com a ilustração a seguir.



Podemos adicionar quantas cores forem necessárias em nosso gradiente e aplicar em qualquer uma das formas aprendidas.

Nosso próximo passo com imagens SVG é criar imagens em arquivo externo “.svg” e usar estas imagens em nossa página.

Vamos criar a imagem de fundo usada na página 2 “Escola de Inglês” em SVG e vamos substituir a imagem original JPG usada no layout.

Crie um novo arquivo e o salve com nome “fundo.svg”, neste novo arquivo adicione o código mostrado a seguir.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve" width="10px" height="150px">

  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" style="stop-color:rgb(51,102,102);stop-opacity:1" />
      <stop offset="62%" style="stop-color:rgb(51,102,102);stop-opacity:1" />
      <stop offset="62%" style="stop-color:rgb(0,51,51);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(51,102,102);stop-opacity:1" />
    </linearGradient>
  </defs>
  <rect x="0" y="0" width="10" height="150" fill="url(#grad1)" />
</svg>
```

Parece complicado? Mas não é, vou explicar.

A primeira parte é simplesmente o cabeçalho do arquivo de acordo com as especificações da W3C, então a tarefa aqui é simplesmente copiar e colar.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-
20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve"
width="10px" height="150px">
```

O mais importante é o código destacado em vermelho que diz respeito ao tamanho da área de trabalho do <svg>, ou seja, o tamanho da imagem, neste caso 10 pixels de largura por 150 pixels de altura.

As próximas linhas dizem respeito da definição do preenchimento gradiente.

```
<defs>
  <linearGradient id="grad1" x1="0%" y1="0%" x2="0%" y2="100%">
    <stop offset="0%" style="stop-color:rgb(51,102,102);stop-opacity:1" />
    <stop offset="62%" style="stop-color:rgb(51,102,102);stop-opacity:1" />
    <stop offset="62%" style="stop-color:rgb(0,51,51);stop-opacity:1" />
    <stop offset="100%" style="stop-color:rgb(51,102,102);stop-opacity:1" />
  </linearGradient>
</defs>
```

E por fim, criamos um retângulo de largura (width) 10 pixels e altura (height) 150 pixels usando o gradiente criado anteriormente como preenchimento.

```
<rect x="0" y="0" width="10" height="150" fill="url(#grad1)" />
```

Salve as alterações.

Pronto, agora já temos um arquivo SVG e vamos copiar este arquivo para mesma pasta onde estão os arquivos da página 2 “Escola de Inglês”.

No arquivo “estilos.css” da página “Escola de Inglês” modifique o código de formatação do <body> de acordo com o código destacado em vermelho da ilustração a seguir.

```
body{
  margin: 0 auto;
  width: 960px;
  background-color: #DDD;
  padding: 0px;
  background: url(fundo.svg) repeat-x;
}
```

Agora basta substituir a imagem do logo nas página index.html, modulos.html e contato.html.

Altere o código destacado em vermelho para usar a imagem “logo.svg” nestas páginas de acordo com o exemplo a seguir.

```
<header id="cab">
  
  <div id="social">
    
    
  </div>
  <nav id="menu">
```

```

        <ul>
            <li><a href="index.html" target="_self">Home</a></li>
            <li><a href="modulos.html" target="_self">Módulos</a></li>
            <li><a href="contato.html" target="_self">Contato</a></li>
        </ul>
    </nav>
</header>

```

Esta imagem “logo.svg” foi feita no CorelDraw, basta configurar a unidade de medida para pixel e o tamanho da página para o tamanho desejado, neste caso 150 x 150 pixels, criar o logo e salvar como SVG, sim, o Corel nos fornece este formato.

Pronto, agora nossa página está mais leve para ser carregada, já que trocamos duas imagens por SVG.



Conteúdo editável

São muito banacas as novas possibilidades com HTML5, e agora podemos permitir que o conteúdo de uma <div> por exemplo seja editável, basta inserir o parâmetro contenteditable="true", veja o código a seguir.

```

<div contenteditable="true">
    Canal Fessor Bruno
</div>

```

Este <div> é mostrada com o texto “Canal Fessor Bruno” mas você pode mudar este texto, obviamente que a alteração do texto não será gravada.

Podemos usar contenteditable="true" em tags <p> <h1>...<h6>, enfim, todas as tags que contém textos.

Revisão ortográfica

Outra grande inclusão no HTML5 é a possibilidade de habilitar revisão ortográfica para elementos com contém textos, elementos de formulário ou que possuem o parâmetro contenteditable="true", para habilitar a revisão ortográfica basta configurar o parâmetro spellcheck em true, como mostra o código a seguir.

```

<div contenteditable="true" spellcheck="true" style="border:1px solid #000">
    Canal Fessor Bruno
</div>

```

Na imagem de exemplo a seguir podemos ver como o corretor destaca a palavra errada.

Este carro é bunito demais

MathML – Criando equações

MathML é uma linguagem baseada em XML para traçarmos equação matemáticas diretamente pelo código HTML, a seguir vamos entender os elementos usados na linguagem MathML.

Infelizmente o Google Chrome nem o Internet Explorer renderizam MathML, para rodar MathML iremos precisar de plug-ins, ou algum browser que suporte a linguagem.

Vamos aos elementos MathML.

Variável

`<mi>` → MathML identifier, pode ser o nome de uma função, uma constante ou uma variável.

`<mi>x</mi>` → Neste caso usamos a letra X como um identificador de valor.

Números

`<mn>` → MathML number, este element representa um numeral literal, numerais inteiros ou reais.

`<mn>6</mn>` → Neste caso representamos o numeral 6

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <mn>6</mn>
  <mi>x</mi>
</math>
```

O código acima gera o resultado 6x.

Operadores

`<mo>` → Elemento para representar um operador =, +, ×, ÷, -

`<mo>+</mo>` → Mostra o operador soma +.

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <mn>6</mn>
  <mi>x</mi>
  <mo>-</mo>
  <mn>2</mn>
  <mi>y</mi>
</math>
```

O código acima gera a seguinte saída: 6x – 2y

Tabela de operadores.

Operador	Entidade	Hexadecimal	Descrição
–	−	−	Subtração
×	×	×	Multiplicação
÷	÷	÷	Divisão
≠	≠	≠	Diferente
≈	≈	≈	Aproximadamente
<	<	<	Menor
≤	≤	≤	Menor ou igual
>	>	>	Maior
≥	≥	≥	Maior ou igual
±	±	±	Mais ou menos

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <mn>6</mn>
  <mi>x</mi>
  <mo>-</mo>
  <mn>4</mn>
  <mi>y</mi>
  <mo>&ge;</mo>
  <mn>0</mn>
</math>
```

O código acima gera a seguinte saída: $6x - 4y \geq 0$

Exponenciação / Superscripts

```
<msup>
  <mi>x</mi>
  <mn>2</mn>
</msup>
```

O código acima irá gerar X elevado ao quadrado.

$$x^2$$

Basicamente <msup> necessita de duas outras tags <mi> que é a base e <mn> para representar o expoente.

Agrupar expressões sup

Algumas vezes precisamos elevar expressões mais elaboradas do que um simples numeral.

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <msup>
    <mi>e</mi>
    <mrow>
      <mn>2</mn>
      <mi>x</mi>
      <mo>+</mo>
      <mn>1</mn>
    </mrow>
  </msup>
</math>
```

O código acima irá gerar o resultado a seguir.

$$e^{2x+1}$$

Note que precisamos agrupar a expressão do expoente com a tag <mrow> isso porque o expoente neste caso não é um número simples e sim uma expressão mais complexa.

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <msup>
    <mrow>
      <mo>(</mo>
      <mrow>
        <mrow><mn>3</mn><mi>x</mi></mrow>
        <mo>-</mo>
        <mrow><mn>2</mn><mi>y</mi></mrow>
      </mrow>
      <mo>)</mo>
    </mrow>
    <mn>2</mn>
  </msup>
```

```
</msup>
<math>
```

Veja o resultado do código acima.

$$(3x - 2y)^2$$

Raizes

Existem duas tags para representar raízes <msqrt> e <mroot>, <msqrt> representa uma raiz quadrada, qualquer outra raiz, cúbica por exemplo, deve ser representada por <mroot>.

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <msqrt>
    <mn>4</mn>
  </msqrt>
  <mo>&times;</mo>
  <mroot>
    <mn>8</mn>
    <mn>3</mn>
  </mroot>
  <mo>=</mo>
  <mn>4</mn>
</math>
```

$$\sqrt{4} \times \sqrt[3]{8} = 4$$

Frações

Para representar frações usamos a tag <mfrac>, veja o exemplo.

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <mfrac>
    <mn>1</mn>
    <mi>x</mi>
  </mfrac>
</math>
```

$$\frac{1}{x}$$

Atributos para <mfrac>.

Attribute	Description
linethickness	Permite configurar a espessura da linha da fração, podemos usar px, pt, em, etc.
numalign	Alinhamento do operador. Pode ser left, right, ou center.
denomalign	Alinhamento do denominador. Pode ser left, right, ou center.
bevelled	Determina se a fração será na vertical ou inline. Pode ser true ou false.

Juntando tudo

```
<math xmlns='http://www.w3.org/1998/Math/MathML' display='block'>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mo>-</mo>
      <mi>b</mi>
      <mo>+<mo>sqrt<mo>
        <msup>
          <mi>b</mi>
          <mn>2</mn>
        </msup>
      <mo>-<mo>4<mo>a<mi>c</mi>
    </mrow>
    <mrow>
      <mn>2</mn>
      <mi>a</mi>
    </mrow>
  </mfrac>
</math>
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

meter

Meter é um elemento novo que mostra uma barra com a medida especificada, é como se fosse uma barra de gráfico, é um elemento simples mas bem interessante, a seguir mostro um exemplo com dois <meter>.

```
<p>Barras meter</p>
<meter value="5" min="0" max="10">5 de 10</meter><br>
<meter value="0.8">80%</meter>
```

Barras meter



<meter value="5" min="0" max="10">5 de 10</meter> → Value indica o valor atual da barra, min e max indicam os valores mínimo e máximo para a barra.

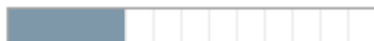
<meter value="0.8">80%</meter> → Neste <meter> configuramos a propriedade value em porcentagem, no caso em 80%.

progress

Progress é um elemento para barra de loading, pode representar por exemplo o progresso em uma determinada tarefa.

```
<p>Carregando</p>
<progress value="32" max="100"></progress>
```

Carregando



`<progress value="32" max="100"></progress>` → Valor do mostrador neste caso é 32 e o valor máximo configurado é 100.

Terceira página em HTML5

Nosso terceiro e último site do curso será sobre a agência de turismo, fictícia, kkkkk é claro, neste trabalho vamos fixar os principais conceitos e aplicar alguns recursos que não usamos nos sites anteriores, confira a seguir como ficará este site.



Vamos começar o projeto do site, o primeiro passo vai ser criar as páginas com todo seu conteúdo, a seguir vou disponibilizar todo código das páginas HTML e você vai copiar todo esse conteúdo com muito cuidado para não errar.

Uma observação importante é quanto aos arquivos anexos, todas as imagens “das cidades”, as imagens banner.png, logo.svg, fundo.svg, fbicone.svg e twticone.svg estão na pasta raiz “pg3”.

Outra observação importante, as imagens SVG foram feitas no CorelDraw, mas você pode tentar construí-las manualmente.

index.html

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="estilos.css">
  <title>CFB Turismo</title>
</head>
<body>

  <header id="cab">
    
```

```

        <div id="sociais">
            <a href="#" target="_blank"></a>
            <a href="#" target="_blank"></a>
        </div>
    </header>

    <nav id="menu">
        <ul>
            <li><a href="index.html" target="_self">Home</a></li>
            <li><a href="destinos.html" target="_self">Destinos</a></li>
            <li><a href="promoco.es.html" target="_self">Promoções</a></li>
            <li><a href="contato.html" target="_self">Contato</a></li>
        </ul>
    </nav>

    <br class="vazio">

    <section id="banner">
        
    </section>

    <aside id="anexo">
        <article>
            <h1>Telefones</h1>
            <p>(31) 3333-3333</p>
            <p>(31) 2222-2222</p>
        </article>
        <article>
            <h1>E-mails</h1>
            <p>comercial@cfbturismo.com.br</p>
            <p>sac@cfbturismo.com.br</p>
        </article>
    </aside>

    <section id="main">
        <article>
            <h1>Viaje bem viaje CFB Turismo</h1>
            <p>Pensou em viajar? Então consulte nossos pacotes nacionais e internacionais, consulte também nossas opções de cruzeiro.</p>
            <p>Compre pelo telefone sem sair de casa, basta se cadastrar em uma de nossas lojas uma única vez e comprar pelo telefone quantas vezes quiser.</p>
            <p>Então você já sabe, pensou em viajar? Pensou CFB Turismo.</p>
        </article>
        <article>
            <h1>Contate-nos</h1>
            <p>Tire todas suas dúvidas, consulte-nos sobre nossos pacotes, clique no menu Contato estamos prontos para atendê-lo na comodidade da sua casa.</p>
        </article>
    </section>

    <br class="vazio">

    <footer id="rodape">
        <p>CFB Turismo, viaje bem viaje CFB Turismo.
    </p>
    </footer>

</body>
</html>

```

destinos.html

```

<!DOCTYPE HTML>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="estilos.css">
    <title>CFB Turismo</title>
</head>
<body>

    <header id="cab">
        
        <div id="sociais">
            <a href="#" target="_blank"></a>
            <a href="#" target="_blank"></a>
        </div>
    </header>

```

```

        </div>
</header>

<nav id="menu">
    <ul>
        <li><a href="index.html" target="_self">Home</a></li>
        <li><a href="destinos.html" target="_self">Destinos</a></li>
        <li><a href="promoco.es.html" target="_self">Promoções</a></li>
        <li><a href="contato.html" target="_self">Contato</a></li>
    </ul>
</nav>

<br class="vazio">

<section id="destinos">
    <article id="nacionais">
        <details>
            <summary>Destinos Nacionais</summary>
            <p>Gramado</p>
            
            <br class="vazio">

            <p>Florianópolis</p>
            
            <br class="vazio">

            <p>Salvador</p>
            
            <br class="vazio">

            <p>Ouro Preto</p>
            
            <br class="vazio">

            <p>Rio de Janeiro</p>
            
            <br class="vazio">

            <p>Fernando de Noronha</p>
            
            <br class="vazio">

            <p>Porto de Galinhas</p>
            
            <br class="vazio">

            <p>Lençóis Maranhenses</p>
            
            <br class="vazio">

            <p>Vitória/Vila Velha</p>
            
            <br class="vazio">

            <p>Curitiba</p>
            
            <br class="vazio">

            <p>Mossoró</p>
            
            <br class="vazio">

            <p>Ubatuba</p>
            
            <br class="vazio">

            <p>São Luís</p>
            
            <br class="vazio">

        </details>
        <details>
            <summary>Destinos internacionais</summary>
            <p>New York</p>
            
            <br class="vazio">

```

```

                <p>Miami</p>
                
                <br class="vazio">

                <p>Las Vegas</p>
                
                <br class="vazio">
            </details>
        </details>
        <summary>Cruzeiros</summary>
        <p>Consulte-nos sobre nossos cruzeiros</p>
        
        <br class="vazio">
    </details>
</article>
</section>

<br class="vazio">

<footer id="rodape">
    <p>CFB Turismo, viaje bem viaje CFB Turismo.
</footer>

</body>
</html>

```

contato.html

```

<!DOCTYPE HTML>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="estilos.css">
    <title>CFB Turismo</title>
</head>
<body>

    <header id="cab">
        
        <div id="sociais">
            <a href="#" target="_blank"></a>
            <a href="#" target="_blank"></a>
        </div>
    </header>

    <nav id="menu">
        <ul>
            <li><a href="index.html" target="_self">Home</a></li>
            <li><a href="destinos.html" target="_self">Destinos</a></li>
            <li><a href="promoco.es.html" target="_self">Promoções</a></li>
            <li><a href="contato.html" target="_self">Contato</a></li>
        </ul>
    </nav>

    <br class="vazio">

    <section id="contato">
        <form id="fContato" name="fContato" method="post" action="#">
            <label>Nome</label><br>
            <input type="text" name="fNome" maxlength="40" size="50" title="Digite seu nome" required
autofocus><br><br>

            <label>Telefone</label><br>
            <input type="tel" maxlength="15" name="fFone" pattern="\([0-9]{2}\) [0-9]{4,6}-[0-9]{3,4}"
placeholder="(xx)xxxx-xxxx" title="Digite seu telefone" required><br><br>

            <label>Email</label><br>
            <input type="email" name="fMail" size="50" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-
z]{2,3}" title="Digite seu email" required><br><br>

            <label>Destino</label><br>
            <input type="text" name="fDestino" maxlength="40" size="50" title="Digite seu local de
destino" required><br><br>

            <label>Informe a data desejada</label><br>

```



```
<input type="date" maxlength="10" name="fData" pattern="[0-9]{2}\/[0-9]{2}\/[0-9]{4}$"
min="2012-01-01" max="2014-02-20" required><br><br>

<input type="submit" value="Enviar" id="btn">
<input type="reset" value="Limpar" id="btn">

</form>
</section>

<br class="vazio">

<footer id="rodape">
<p>CFB Turismo, viaje bem viaje CFB Turismo.
</footer>

</body>
</html>
```

promocoes.html

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="estilos.css">
<title>CFB Turismo</title>
</head>
<body>

<header id="cab">

<div id="sociais">
<a href="#" target="_blank"></a>
<a href="#" target="_blank"></a>
</div>
</header>

<nav id="menu">
<ul>
<li><a href="index.html" target="_self">Home</a></li>
<li><a href="destinos.html" target="_self">Destinos</a></li>
<li><a href="promocoes.html" target="_self">Promoções</a></li>
<li><a href="contato.html" target="_self">Contato</a></li>
</ul>
</nav>

<br class="vazio">

<section id="promocoes">
<article id="promo1">
<h1>Caldas Novas</h1>

<p>Caldas Novas é um município brasileiro do estado de Goiás.</p>
<p>O município é conhecido por ser a maior estância hidrotermal do mundo, possuindo águas
que brotam do chão em temperaturas que variam de 43° a 70°. Na alta temporada, a cidade chega a comportar
mais de 500 mil turistas.</p>
<p>A estrutura da cidade conta com hotéis, pousadas, chalés, clubes, boates e bares. Uma
outra grande atração de Caldas Novas é o ecoturismo, vez que a cidade encontra-se às margens do lago da
represa de Corumbá e ao lado da Serra de Caldas.</p>
<label>Diversão</label><br><meter value="0.8"></meter><label>80%</label></meter><br>
<label>Custo</label><br><meter value="0.5"></meter><label>50%</label></meter><br>
</article>
<br class="vazio">
<article id="promo2">
<h1>Campos do Jordão</h1>

<p>Campos do Jordão é um município brasileiro localizado no interior do estado de São
Paulo, mais precisamente na Serra da Mantiqueira; faz parte da recém-criada Região Metropolitana do Vale do
Paraíba e Litoral Norte, sub-região 2 de Taubaté.</p>
<p>A cidade fica à altitude de 1 628 metros, sendo portanto, o mais alto município
brasileiro, considerando a altitude da sede. Sua população estimada, em 2014, era de 50 541 habitantes. Dista
173 km da cidade de São Paulo, 350 km do Rio de Janeiro e 500 km de Belo Horizonte. Sua principal via de
acesso é a Rodovia Floriano Rodrigues Pinheiro.</p>
<label>Diversão</label><br><meter value="0.5"></meter><label>50%</label></meter><br>
<label>Custo</label><br><meter value="0.7"></meter><label>70%</label></meter><br>
</article>
```

```
</section>

<br class="vazio">

<footer id="rodape">
    <p>CFB Turismo, viaje bem viaje CFB Turismo.
</footer>

</body>
</html>
```

Pronto, com todo conteúdo digitado precisamos formatar, então crie o arquivo “estilos.css” e vamos digitar todo o código de formatação, a seguir vou explicar somente os comandos novos.

```
body{
    margin: 0 auto;
    width: 960px;
    background-image:url(fundo.svg);
    background-repeat: repeat-x;
    background-color: #DDD;
    padding: 0px;
    overflow-y: scroll;
}

.vazio{
    clear: both;
}

#menu{
    float:left;
    width: 960px;
}

#menu a:hover {
    color:#000;
}

#menu ul {
    list-style: none;
    padding: 0px;
    margin: 0px;
}

#menu li {
    height: 50px;
    display: table-cell;
    text-align: center;
    vertical-align: middle;
    width:240px;
}

#menu ul li a{
    text-decoration: none;
    font-size:20px;
    color: #FFF;
}

#logo{
    float: left;
}

#sociais{
    float: right;
    margin-top: 95px;
}

#main{
    background: #EEE;
    border-radius: 20px;
    padding:20px;
    float: left;
    width: 660px;
}

#destinos, #contato, #promocoas{
```

```
        background: #EEE;
        border-radius: 20px;
        padding:20px;
        width: 960px;
    }

    #main h1, #destinos{
        font-family: Arial;
        font-size: 20pt;
    }

    #main p, #destinos{
        font-size: 1.3em;
        text-align: justify;
    }

    #destinos img{
        float: left;
        padding: 0px 10px 0px 0px;
    }

    #anexo{
        background: #EEE;
        border-radius: 20px;
        padding:20px;
        width: 200px;
        float: right;
        margin-left: 10px;
    }

    #btn{
        color: #FFF;
        background-color: #3C5D95;
        font-size:1em;
        width: 150px;
        height: 40px;
        border-radius: 15px;
        border: 0px;
        cursor:pointer;
    }

    #btn:hover{
        background-color: #953C5D;
    }

    summary{
        font-size: 15pt;
        color: #FFF;
        background-color: #3C5D95;
        padding: 5px 0px 5px 10px;
        margin: 10px 0px 10px 0px;
        cursor:pointer;
    }

    details[open] summary{
        background-color: #953C5D;
    }

    input{
        font-size: 1.3em;
    }

    input:hover{
        border-color:#F00;
    }

    input:focus{
        background-color:#FDD;
    }

    footer{
        padding: 10px 0px 0px 0px;
    }
```




#destinos, #contato, #promocoes{ → Nesta linha aproveitamos os códigos de formatação para todos os três IDs é uma forma de reaproveitamento de código.

#btn: hover{ → Este código faz referência de como o botão será apresentado quando o mouse estiver sobre ele, ou seja, é a formatação do evento hover do botão.

input: focus{ → Aqui formatamos o evento focus dos input, ou seja, quando a caixa de texto estiver selecionada “focus” será formatada de acordo com o código especificado no bloco.

Resumo das tags HTML

Neste capítulo do curso vou descrever as tags existentes em HTML.

Novas tags HTML 5

TAG	Descrição
<article>	Representa um artigo, basicamente um bloco de conteúdo, texto, imagens, tabelas, vídeos, etc.
<aside>	Representa uma “caixa” com conteúdo anexo à página, com qualquer elemento.
<audio>	Inserir arquivo de áudio.
<bdi>	Texto para formatação bidirecional.
<canvas>	Área utilizada para desenhar gráficos em javascript.
<command>	Comando personalizado, pode ser chamado pelo usuário.
<datalist>	Lista do tipo “auto complete” para elementos de formulário.
<datatemplate>	Bloco para templates de dados.
<embed>	Inserir conteúdo externo como SWF, vídeos, áudios.
<figcaption>	Texto caption para um elemento do tipo <figure>.
<footer>	Área do rodapé da página.
<header>	Área do cabeçalho da página.
<keygen>	Gerar pares de chaves em um formulário.
<mark>	Destacar/marcar partes de um texto.
<meter>	Controle gráfico que mostra uma barra métrica que indica graficamente um valor.
<nav>	Seção de navegação principal, contém os links principais da página.
<output>	Usada para mostrar o resultado de um cálculo, não suportada no I.E.
<progress>	Barra de progresso, tipo loading para mostrar o progresso de uma determinada tarefa.
<ruby>	Anotação ruby.
<rp>	Texto a ser mostrado caso o browser não suporte anotações ruby.
<rt>	Texto de anotação ruby.
<section>	Bloco de seção do documento.



<source>	Apontador para os arquivos de mídia das tags <audio> e <video>.
<summary>	Cabeçalho para a tag <details>.
<time>	Representa uma data o hora.
<vídeo>	Tag para inserir vídeos na página.
<wbr>	Especifica uma quebra de linha para textos longos, especificamos onde o texto será quebrado.

Tags já existentes em HTML 4 que foram continuadas

TAG	Descrição
<!-- -->	Comentário.
<!doctype>	Definição do tipo de documento.
<a>	Âncoras, links.
<abbr>	Marca um texto que será configurado como abreviação.
<address>	Configura um texto como sendo um endereço.
<area>	Criação de mapas em imagens.
	Bold/Negrito.
<base>	Configuração da URL base para os links da página.
<bdo>	Altera a direção do texto.
<blockquote>	Texto de citação.
<body>	Corpo da página, onde estão os elementos visíveis.
 	Quebra de linha.
<button>	Botão padrão, sem ação pré-configurada.
<caption>	Cabeçalho para tabelas.
<cite>	Insere uma citação no meio de uma frase.
<code>	Configura um texto como um código.
<col>	Atributos para colunas de tabelas.
<colgroup>	Grupo de colunas de tabelas.
<dd>	Descrição de itens de listas de definição.
	Configura o texto como deletado.
<div>	Caixa/container genérico para conteúdo.
<dfn>	Texto de definição para ser inserido no meio de frases.
<dl>	Criar listas de definição.
	Configuração de texto em ênfase/destacado.
<fieldset>	Agrupa elementos.



<form>	Formulário.
<h1> a <h6>	Títulos.
<head>	Cabeçalho do documento, com tags de “configuração” da página.
<hr>	Horizontal rule, traça uma barra (linha) horizontal na página.
<html>	Tag principal da página, marca o início e o fim do código.
<i>	Itálico.
<iframe>	Quadro dinâmico, carrega uma página dentro dele e pode ser posicionado na página principal.
	Inserir imagens.
<input>	Elementos de formulário para entrada de dados.
<ins>	Formatação de texto inserido, ao contrário da tag .
<kbd>	Formatação para indicar que o texto deve ser digitado.
<label>	Etiqueta para elementos de formulário.
<legend>	Título do elemento <fieldset>.
	Itens de uma lista.
<link>	Referência para arquivos externos tipo, CSS, JS.
<map>	Definição mapas em imagens.
<meta>	Metadados definidos para o documento.
<noscript>	Texto mostrado quando o browser não suporta script.
<object>	Objeto inserido na página, áudio, vídeo, etc.
	Lista ordenada.
<optgroup>	Grupo de elementos <option>
<p>	Parágrafo.
<param>	Parâmetros para a tag <object>.
<pre>	Texto pré-formatado.
<q>	Texto de citação, insere aspas no início e o fim.
<samp>	Saída de um código/programa.
<script>	Bloco de scripts a serem executados na página, javascript por exemplo.
<select>	Lista para seleção de itens, formulário.
	Bloco/container genérico, semelhante ao <div>, porém não quebra linha.
	Formatação tipo negrito.
<style>	Definição de estilos CSS.
<sub>	Texto sobescrito.



<sup>	Texto Sobrescrito.
<table>	Tabela.
<tbody>	Corpo da tabela.
<td>	Table division, coluna da tabela.
<tr>	Table row, linha da tabela.
<th>	Coluna de cabeçalho da tabela.
<tfoot>	Rodapé da tabela.
<textarea>	Campo de texto multilinha, formulário.
<title>	Título da página.
	Lista não ordenada.
<var>	Texto que representa variável.

Tags descontinuadas, não devem ser usadas

TAG	Descrição
<acronym>	Acrônimo.
<applet>	Applet imbutido na página.
<basefont>	Configurações padrões da fonte.
<bgsound>	Som de fundo na página.
<big>	Aumenta o tamanho da fonte.
<center>	Centraliza conteúdo.
<frame>	Quadro de um <frameset> para conteúdo na página.
	Formatação de fonte.
<fn>	Nota de rodapé.
<frameset>	Conjunto de quadros.
<hgroup>	Grupos de tags <h1> a <h6>, foi criada em HTML5 e já foi descontinuada.
<isindex>	Campo de texto simples.
<dir>	Lista de diretórios.
<noembed>	Texto a ser exibido se o elemento embutido não for suportado.
<noframe>	Texto a ser exibido se frames não forem suportados.
<s>	Texto riscado.
<strike>	Texto riscado.
<tt>	Texto mono espaçado.
<u>	Underline/sublinhado.



<xmp>

Texto onde as tags HTML são ignoradas.