

DESENVOLVIMENTO WEB

Prof Alann Perini

Briefing do projeto que será desenvolvido

A empresa Artech Ar Condicionados nos procurou para que desenvolvêssemos um site de divulgação de seus produtos e para que os seus clientes pudessem entrar em contato com a mesma.

Desenvolveremos então nessa trilha o front-end de uma aplicação web, que consiste em 03 páginas:

- Página inicial - Conterá informações gerais da empresa
- Produtos - Exibe os produtos disponíveis
- Fale conosco - Permite aos clientes que entrem em contato com a empresa

As imagens a seguir ilustram essas 03 páginas.



Página 2





Artech Ar Condicionados

[Início](#)[Produtos](#)[Fale Conosco](#)



Fale Conosco!

Ficou com alguma dúvida? Preencha este formulário e entraremos em contato!

Dados Pessoais:

Nome:

Telefone:

Email:

Data Nascimento:
dd/mm/aaaa

Motivo do Contato:

Motivo:

Escolha

Comentário:

Copyright © Escola Sistêmica 2019

Como pode observar, todas as páginas contém o logo da empresa, menu e rodapé. Na “Página inicial”, a primeira página que o usuário verá ao acessar o site da Artech Ar Condicionados, além da estrutura já mencionada, conterão informações sobre a empresa. Já na página de “Produtos”, tabelas com os ar condicionados e aquecedores que a empresa vende serão mostrados. Por fim, na página “Fale conosco”, um formulário para que o usuário entre em contato com a empresa será disponibilizado.

Alguns conteúdos dessas páginas serão carregados dinamicamente com JavaScript porém inicialmente desenvolveremos apenas os códigos HTML “fixos” de cada página. Além disso, a estilização das páginas será feita com CSS. No entanto, mostraremos a você as imagens de como as páginas deverão ser apresentadas após o desenvolvimento do código, momento esse que será sempre o ideal para que faça seus testes.

Criação do arquivo inicial e sua codificação HTML

Agora, criaremos a estrutura da nossa primeira página, que chamaremos de **index.html**. Reparou que o arquivo já vem com as tags básicas? Porém faltam algumas coisas, como a linguagem de nosso site e as tags complementares que utilizaremos em todas as páginas: as tags meta. Deixe seu código como o da imagem a seguir para atendê-las.

```

1  <!DOCTYPE html>
2  <html lang="PT-BR">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="description" content="Site Artech Ar Condicionados">
6  <meta name="keywords" content="Ar-condicionado,Refrigeração, Inverter, Janela,
7  Split,Aquecedores,Artech">
8  <meta name="author" content="Artech">
9  <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 <title>Artech Ar Condicionados</title>
11 <link rel="stylesheet" type="text/css" href="site.css">
12
13 </head>
14

```

Agora vamos fazer o código de dentro da **tag body da index.html**, começando pelo cabeçalho. Veja a imagem abaixo e copie seu código:

```

15 <body>
16   <header>
17     
18     <h1>Artech Ar Condicionados</h1>
19

```

Repare que focamos na **tag body**, mas é óbvio que o restante não deve ser excluído, certo? Vamos falar agora sobre as tags usadas:

- **header** - Define nosso cabeçalho da página. Semanticamente correta e necessária no nosso caso;
- **img** - Insere uma imagem. Precisa do caminho da mesma e de um texto alternativo caso não seja carregada por algum motivo (como problema na conexão). Esse texto alternativo também é lido por leitores de tela criados para a acessibilidade. É um novo atributo da HTML5 que entra na vertente acessibilidade dessa nova versão ;
- **h1** - cria um título do maior tamanho possível.

```

22   <nav>
23     <a href="index.html"><div class="opcao">Inicio</div></a>
24     <a href="produtos.html"><div class="opcao">Produtos</div></a>
25     <a href="faleconosco.html"><div class="opcao">Fale Conosco</div></a>
26   </nav>

```

As novas tags:

- **nav** - Cria área para navegação (como menus e afins). Semanticamente correto;
- **a** - Cria um hiperlink para outro arquivo (local), outro site (externo) ou algum local da própria página (interno);
- **div** - Cria uma divisão de conteúdo. Nesse caso, será para que ao clicar nela, e não apenas no texto, o usuário seja redirecionado.

Seguimos agora com a última parte do código, novamente destacando na imagem onde o código novo deve ficar:

```

27
28     <section>
29         <div id="topo">
30             
32             <div id="titulos">
33                 <h2>BEM VINDO AO SITE DA ARTECH<br>AR CONDICIONADOS</h2>
34                 <h3>A maior Loja de Ar Condicionado da região !!!</h3>
35             </div>
36             
38         </div>
39         <br>
40         <p>Artech Ar Condicionados é a maior empresa de revenda de
41         Ar Condicionados e afins da região. Desde 2007</p>
42         <p>Sua Ar Condicionados estragou? Está precisando de um maior ou com mais
43         potência? Quer mais conforto e praticidade? Já pensou em fazer um consórcio
44         de Ar Condicionados?</p>
45     </section>
46     <footer>Copyright &copy; Artech Ar Condicionados 2019 </footer>
47 </body>
48 </html>

```

Mais uma vez, vamos às tags novas:

- **section** - Define uma seção. No nosso caso, definirá a área para os conteúdos específicos de cada página;
- **Div** - é um container genérico para conteúdo de fluxo, usado para agrupar elementos para fins de estilos (usando class ou id), ou porque eles compartilham valores de atributos.
- **footer** - Define um rodapé. Mais uma vez, semanticamente é o correto a se fazer.
- **P** – indica que está sendo utilizado um parágrafo.

Criação dos demais arquivos e edição do necessário

Agora que já temos o arquivo base, podemos criar os demais.

Crie mais 2 páginas html e 1 página css aonde vamos trabalhar os estilos das páginas, a “**produtos.html**”, crie a “**faleconosco.html**” e crie a página “**site.css**”.

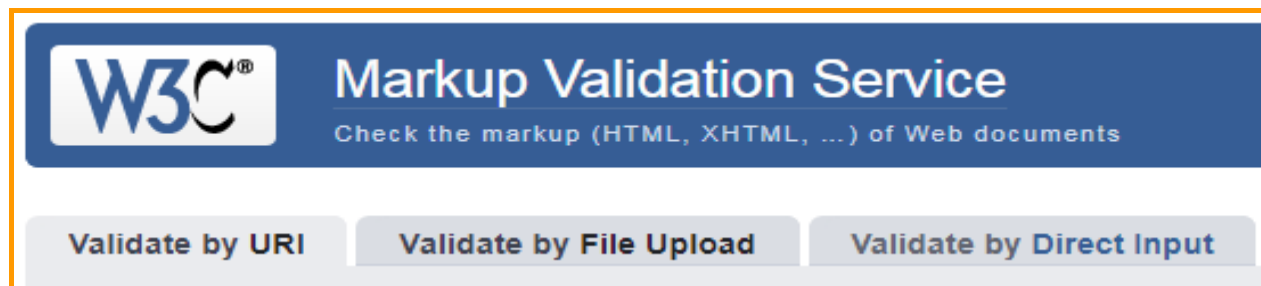
Tendo esses 2 arquivos, podemos testar se os links do menu estão OK. Para isso, abra a **index.html** no navegador (ou atualize-a) e clique no link para a página “**Produtos**”:

Validador HTML

Bem, agora que os códigos foram terminados, precisamos nos certificar que ele foi escrito corretamente. Para isso, existem sites validadores do nosso código em HTML. Com ele, podemos ver se nosso site está dentro das boas práticas e se tudo foi feito dentro dos padrões, como colocar e fechar as tags no local certo, usar os atributos corretos, entre outras coisas. Isso é muito importante, já que os navegadores são feitos para “corrigir” automaticamente nossos erros ao interpretar uma página HTML, mas como isso é feito por ele, ele pode fazê-lo em um local errado. Além disso, o CSS e o JavaScript não vão aceitar isso de bom grado: um dos maiores problemas de quem faz CSS é um HTML mal feito!

Sendo assim, vamos usar o [Validator W3C](https://validator.w3.org/) para verificar se tudo está nos conformes.

Veja o topo de sua página inicial:

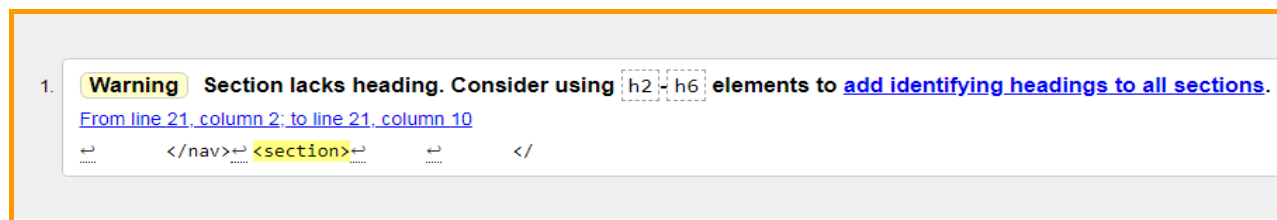


Nele, temos 3 opções de validação:

- URI: Para validar sites online;
- File Upload: para você fazer upload dos arquivos que quiser validar;
- Direct Input: para digitar ou colar o código que deseja testar.

Vamos sugerir aqui o Direct Input, pois já estamos com o código aberto, mas se quiser fazer por File Upload, fique a vontade.

Clique em Direct Input, copie primeiramente todo seu código da index, e cole na área. Para verificar, clique em Check. Se seu código estiver igual ao nosso, receberá apenas um Warning (aviso):



Ele acusa que nossa **tag section** deveria ter um elemento de título, porém como ainda não preenchemos nossa **section**, vamos ignorar esse aviso por enquanto.

Repita o procedimento com as demais páginas. Por motivos óbvios, o resultado deve ser o mesmo Warning que recebemos antes. Se não, reveja os códigos parte que indica a codificação que causou Warning/Error e corrija-a em todos os arquivos.

Assim que as 3 páginas estiverem acusando apenas o Warning da section, sua missão está cumprida!

Introdução ao CSS

Abaixo criamos a página **site.css** aonde colocaremos as configurações de estilos da nossa página.

```
1  @charset "UTF-8";
2
26 /*****
27  TELA INICIAL
28  *****/
29
30  body{
31      margin: 0 auto;
32      font-family: "Verdana", sans-serif;
33  }
34
35  header{
36      height: 170px;
37      background-color: #008eb6;
38  }
39  header img{
40      float: left;
41      height: 150px;
42      width: 150px;
43      margin: 10px;
44  }
45
46  header h1{
47      float: left;
48      margin: 50px 0 0 30px;
49      font-size: 60px;
50      color: #ffffff;
51  }
52  nav{
53      clear: both;
54      width: 80%;
55      margin: 0px auto;
56      height: 55px;
57  }
58
59
60  nav a{
61      color: white;
62      text-decoration: none;
63      text-align: center;
64  }
```

Repare que temos 3 seletores agora. Vamos explicá-los brevemente:

- **header** - muda a tag homônima, alterando sua altura e cor de fundo;
- **header img** - altera as tags img que estiverem dentro da tag header (ou seja, a logo...), fazendo-a flutuar para a esquerda (float:left), e alterando sua altura, largura e margem;
- **header h1** - modifica as tags h1 que estiverem dentro da header, fazendo-a flutuar para a esquerda, e alterando sua margem, tamanho e cor da fonte.
- **Float**- Por padrão, todos os elementos HTML ficam um em cima do outro, na ordem em que foram declarados. Assim, usamos o float para que eles vão se “ajeitando” um ao lado do outro.

ATENÇÃO! Float não é o único jeito de colocar elementos HTML lado a lado. Outros exemplos são vistos aplicando a propriedade CSS display, com valores como flex e inline-block.

Css dos Menus

```
52
53  /* Menu */
54
55  nav{
56      clear:both;
57      width: 80%;
58      margin: 0px auto;
59      height: 55px;
60  }
61
62  nav a{
63      color: white;
64      text-decoration: none;
65      text-align: center;
66  }
67  nav .opcao{
68      float: left;
69      width: 31.3%;
70      background-color: #b1ccf5;
71      padding: 10px 0;
72      margin: 0 1%;
73      font-size: 26px;
74  }
75
76  nav .opcao:hover{
77      background-color: #83afd3
78  }
```

Estilização do menu da página

Começaremos a formatação do menu por uma pequena mudança na página index.html. Nela, apenas adicionaremos uma classe nas divs dos opções do menu. (Se vc copiou o código essa parte já foi feita class = “opcao”).

Com o seletor nav, mudamos a tag homônima. Primeiro, usamos o “**clear:both**”, depois configuramos largura, margem e altura.

Aprofundando o “**clear:both**”, usamos ele como uma boa prática após um elemento que contenha algo com float (como nesse caso, onde o elemento antes da nav, o header, possui float em seus filhos), para evitar que os elementos continuem levando esse float em consideração.

```
53  /* Menu */
54
55  nav{
56      clear:both;
57      width: 80%;
58      margin: 0px auto;
59      height: 55px;
60  }
61
```

Assim, alinhamos as divs que possuem essa classe uma ao lado da outra, definimos largura, cor de fundo, espaçamento do texto internamente em relação à div, margem e tamanho da fonte.

Outro ponto relevante a se destacar nessa imagem é o uso das porcentagens. Para explicá-las, vamos recorrer a um pouco de matemática. Primeiro, você já sabe que 100% de alguma coisa representa sempre a totalidade daquilo (como “100% dos votos” representa “todos os votos”). Queremos preencher 100% da largura da tag nav usando os 3 botões, assim, devemos dividir esses 100% em 3, o que gera uma dízima periódica de 33,3%. Porém, porque usamos então apenas 31,3% na largura? Repare na margem e verá que usamos 1% de margem de cada lado. Desse modo, 31,3% (de largura da div) + 1% (margem esquerda) + 1% (margem direita) é igual aos 33,3% necessários para que os 3 botões preencham totalmente a nav (na verdade, 99,9%, mas é suficiente).

Por fim, vamos fazer mais uma configuração em nosso menu:

```
75
76  nav .opcao:hover{
77      background-color: #83afd3
78  }
```

Aqui, vemos um tipo de seletor diferente: o **hover**, que permite-nos estilizar algo quando o usuário **passar o mouse** em cima de algo na página. Usando-o dessa maneira, fazemos com que ao se passar o mouse em um elemento com a classe opção que estiver dentro de uma tag nav (ou seja, nas divs que serão nossos links), a cor de fundo desse elemento será alterada.

Estilização da área central da página

Todas as nossas páginas terão conteúdos diferenciados na tag section, então a configuração dela será simplificada. Insira o código da imagem logo abaixo do anterior no arquivo CSS:

```
section{
    clear:both;
    min-height: 360px;
    width: 100%;
    margin-top: 0px;
}

section #topo{
    width: 1050px;
    margin: 0px auto;
    height: 270px;
}
```

Com isso, alteramos a configuração da tag section para novamente desconsiderar os float anteriores (clear:both), decretar altura mínima (para que se o conteúdo que posteriormente colocaremos dentro dela for mais alto do que 360px, ela se expanda conforme necessário for), largura e margem apenas para o topo.


```

93 .arcondicionado{
94     height: 200px;
95     width: 200px;
96 }
97 .direito{
98     float: right;
99 }
100 .esquerdo{
101     float: left;
102 }
103 .topo2{
104     height: 250px;
105     width: 800px;
106 }
107 }
108

```

```

109 section #titulos{
110     float: left;
111     height: 250px;
112     width: 650px;
113     margin: 0px auto;
114 }
115
116 h2{
117     text-align: center;
118     margin-top: 0px;
119     margin-bottom: 20px;
120     font-size: 40px;
121 }
122 h3{
123     text-align: center;
124     margin-top: 50px;
125     font-size: 20px;
126 }
127
128 section p{
129     clear: both;
130     text-align: center;
131     margin: 10px 80px;
132 }
133

```

Estilização do rodapé da página

Agora só falta o rodapé na nossa estilização geral! Vamos ao código dele, que deve ir abaixo dos anteriores:

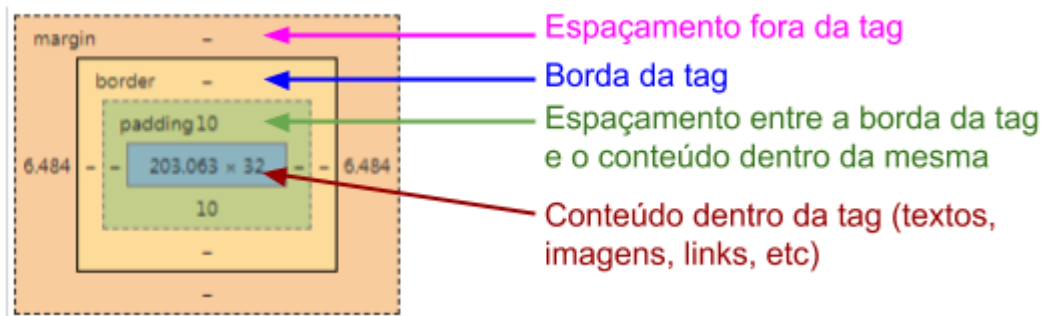
```

135
136 /* RODAPÉ */
137
138 footer{
139     background-color: #005f89;
140     color: #fff;
141     padding: 10px;
142     text-align: center;
143 }
144
145

```

Padding ou Margin?

Nesse documento, usamos essas 2 propriedades que tem a ver com espaçamento. Muitos confundem o papel de cada uma, por isso faremos um breve adendo falando de suas diferenças.



É importante destacar que, mesmo que não seja visível, considera-se a existência de bordas em todos os elementos, para simples referência de tamanho e posição. Pensando desse modo, é mais fácil decidir entre qual dos dois usar: se você achar que precisa afastar algo que está dentro de um elemento da sua borda (como um texto que está muito grudado na borda), deve usar o **padding**, já se sua ideia é separar dois elementos distintos, deve usar o **margin**. Alguns elementos não possuem uma borda interna, como um elemento `` por exemplo, então não faria sentido utilizar um padding nesse caso.

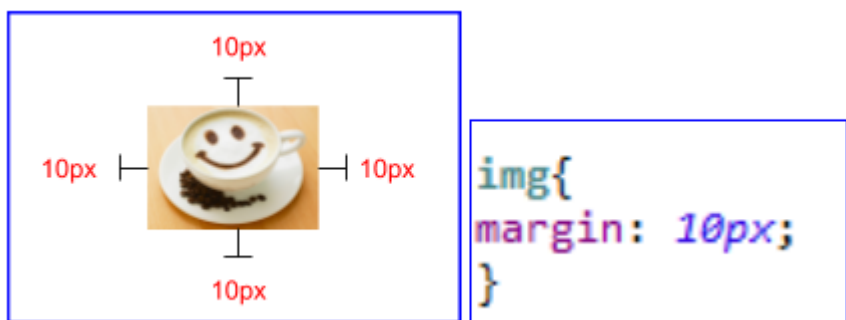
Formatos de escrita de algumas regras

Você deve ter notado que no decorrer do desenvolvimento dessas atividades que utilizamos a propriedade `margin` de maneiras diferentes, ora atribuindo um valor, ora atribuindo dois valores e ora atribuindo quatro valores a ela (é possível também defini-la atribuindo 3 valores) como nas imagens que seguem:

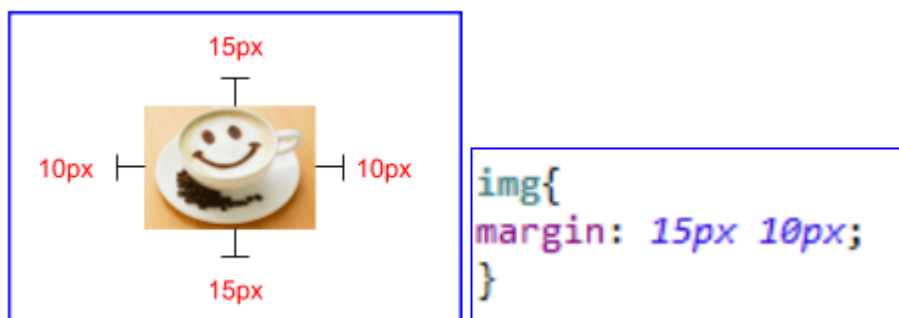
```
header h1{
  float: left;
  margin: 50px 0 0 30px;
  font-size: 60px;
  color: #ffffff;
}

nav .opcao{
  float: left;
  width: 31.3%;
  background-color: #b1ccf5;
  padding: 10px 0;
  margin: 0 1%;
  font-size: 26px;
}
```

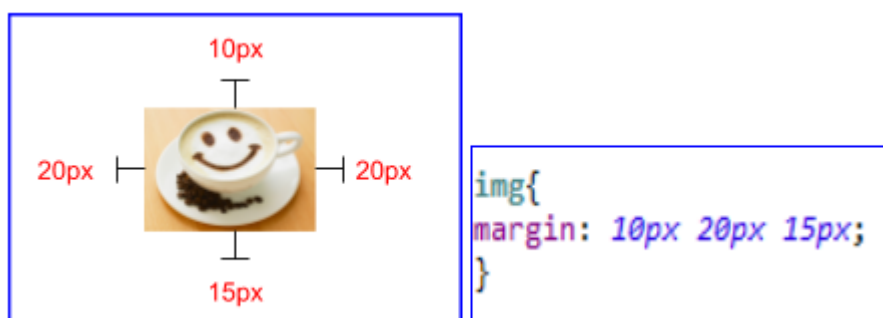
Quando declaramos uma propriedade com apenas um valor, entende-se que esse valor vale para as quatro margens: **superior (top)**, **inferior (bottom)**, **direita (right)** e **esquerda (left)**:



Já quando declaramos dois valores: o primeiro valor refere-se à margem superior e inferior e o segundo valor refere-se à margem **esquerda e direita**.



É possível também declarar a propriedade margin com três valores: o primeiro valor refere-se à margem **superior**, o segundo às margens **esquerda e direita** e o último valor à margem **inferior**.



Você ainda pode escrever as regras especificando a margem desejada, basta usar a propriedade margin-top, por exemplo, para a margem superior. A propriedade padding e border também podem ser escritas utilizando as shorthands, nome dado a essas “facilidades”.

Validador CSS

Assim como fizemos no HTML, podemos verificar se nosso CSS está corretamente codificado. Para isso, é só acessar o [validador CSS do W3C](#). O visual e as opções são iguais às do HTML:

Captura de tela do Validador CSS do W3C. O cabeçalho mostra o logo W3C e o título "CSS Validation Service". Abaixo, há uma barra de navegação com opções: "Por URI", "Por upload de arquivo" (selecionada) e "Por entrada direta". O formulário principal tem o título "Validar por URI" e um campo de texto para "Endereço:". Abaixo do campo, há um botão "Verificar". No rodapé, há uma seção de promoção para o "Professional Certificate Program" e uma nota sobre a validação de CSS incorporado a um documento (X)HTML.

Sugerimos, novamente, que vá em “Por entrada direta” e copie o código do arquivo CSS para testá-lo.

*** Continuaremos no próximo material nosso site**