# BootCamp HTML5 e CSS3

## HTML

### Elemento

Em 1991 Tim Berners-Lee criou essa linguagem de marcação para melhorar a comunicação entre ele e seus colegas de trabalho no CERN, desde então já surgiram 5 versões e o HTML se tornou a base da web.

Com o HTML definimos o significado e a estrutura do conteúdo da web e, além de texto, nossas páginas precisam de imagens, vídeos e vários outros formatos e para isso temos os elementos HTML.

Um elemento HTML é formado pela tag de abertura e seus atributos, o conteúdo e uma tag de fechamento. E mais a frente veremos que existem elementos que não tem tag de fechamento.

Com esses elementos podemos agrupar tipos de conteúdo, alterar tamanho e forma de fontes e adicionar diferentes mídias ao nossa página na web.

E agora podemos ver como é a estrutura básica de um arquivo HTML.

A primeira linha do documento deve ser o <!DOCTYPE html>, apesar de parecer um elemento HTML ela apenas diz ao navegador que ele está lidando com um arquivo do tipo HTML5. Os elementos HTML vem logo abaixo.

**<html>**

A tag html é a raiz do seu documento, todos os elementos HTML devem estar dentro dela. E nela nós informamos ao navegador qual é o idioma desse nosso documento, através do atributo lang, para o português brasileiro usamos pt-BR.

**<head>**

A tag head contém elementos que serão lidos pelo navegador, como os metadados - um exemplo é o charset, que é a codificação de caracteres e a mais comum é a UTF-8, o JavaScript com a tag script, o CSS através das tags style e link - veremos a diferença quando falarmos sobre CSS - e o título da página com a tag title.

**<body>**

E dentro da tag body colocamos todo o conteúdo visível ao usuário: textos, imagens, vídeos.

### Estrutura Básica

<!DOCTYPE html>

<html>

<head>

Aqui possuem metas informações que o navegador necessita.

<meta>

<title></title>

</head>

<body>

Onde estará o conteúdo da página

</body>

</html>

### Semântica

Durante muitos anos o elemento padrão no HTML era a div, construíamos nosso conteúdo todo baseado nela, e assim nascia a sopa de divs.

Mas em 2014 saiu a quinta versão do HTML, e com ela vieram vários mudanças importantes, como performance e acessibilidade, mas nesse curso introdutório vamos focar na semântica.

A semântica nos permite descrever mais precisamente o nosso conteúdo, agora um bloco de texto não é apenas uma div, agora é um article e tem mais significado assim. E temos vários elementos para ressignificar as divs:

<section>

Representa uma seção genérica de conteúdo quando não houver um elemento mais específico para isso.

<header>

É o cabeçalho da página ou de uma seção da página e normalmente contém logotipos, menus, campos de busca.

<article>

Representa um conteúdo independente e de maior relevância dentro de uma página, como um post de blog, uma notícia em uma barra lateral ou um bloco de comentários. Um article pode conter outros elementos, como header, cabeçalhos, parágrafos e imagens.

<aside>

É uma seção que engloba conteúdos relacionados ao conteúdo principal, como artigos relacionados, biografia do autor e publicidade. Normalmente são representadas como barras laterais.

<footer>

Esse elemento representa o rodapé do conteúdo ou de parte dele, pois ele é aceito dentro de vários elementos, como article e section e até do body. Exemplos de conteúdo de um <footer> são informações de autor e links relacionados.

<h1>-<h6>

Eles não foram criados na versão 5 do HTML e nem são específicos para semântica, mas servem para esse propósito. São utilizados para marcar a importância dos títulos, sendo <h1> o mais importante e <h6> o menos. Uma dica: use apenas um <h1> por página, pois ele representa o objetivo da sua página.

### Textos e Links

A criação do HTML foi motivada pela necessidade de compartilhar textos e documentos, e mesmo depois de quase 30 anos, com toda a evolução da web, isso ainda representa uma boa parte do conteúdo da web.

Já falamos anteriormente sobre os elementos h1-h6 e, eles são essenciais para nos indicar visualmente a importância e localização de seções de texto na página, mas para textos maiores e mais densos usamos o elemento p.

O <p> representa um parágrafo, mas ele não suporta apenas texto, podemos adicionar imagens, código, vídeos e vários outros tipos de conteúdo dentro dele.

Um outro elemento interessante e extremamente necessário na web é o <a> - que significa anchor/âncora, ele representa um hyperlink, é ele que interliga vários conteúdos e páginas na web.

O elemento a tem vários atributos, mas vamos focar em dois, o href e o target.

O href representa o hyperlink para onde sua âncora aponta, pode ser uma página do seu ou de outro site, um e-mail e até mesmo um telefone, os dois últimos precisam dos prefixos “mailto:” e “tel:”, respectivamente.

O target neste momento vai servir para nos ajudar a abrir nossos links em outra aba do navegador usando o valor “\_blank”.

### Imagens

A web também é feita de imagens e para representá-las temos o elemento <img>, ele é um daqueles elementos sem tag de fechamento.

O elemento img é bem simples, contendo apenas 2 atributos próprios, o src e o alt.

O src é obrigatório e **guarda o caminho para a imagem** que você quer mostrar na página.

<img src=”img/avatar.jpg”>

O alt não é obrigatório, mas é altamente recomendado por melhorar a acessibilidade, ele **mostra a descrição da imagem caso ela não carregue** e leitores de tela usam esse atributo para descrever a imagem para o usuário saber o que ela significa.

<img alt=”Foto de qualquer coisa”>

### Listas

Os últimos elementos que veremos neste módulo são os relacionados a lista: <ul>, <ol> e <li>.

Listas servem para agrupar uma coleção de itens, como uma lista de ingredientes ou, como será no nosso caso, uma lista com contatos.

O elemento ul cria uma lista não ordenada, onde a ordem dos elementos não é importante, e é representada com pontos, círculos ou quadrados.

O <ol> serve para criar lista ordenadas, nessas a ordem importa, portanto elas são representadas com números, algarismos romanos ou letras.

E o elemento li é um item dentro de uma dessas listas. Um <li> pode conter vários tipos de conteúdo, como parágrafos, imagens e até outras listas.

## CSS

### Definição

Após a criação do HTML a necessidade de formatar as páginas ficou evidente, assim, em 1996, foi criada a linguagem de estilo que conhecemos por CSS.

A sintaxe é bem simples e pode ser explicada com a frase "você cria regras de estilo para elementos ou grupos de elementos".

Vamos usar um elemento HTML que vimos anteriormente, a âncora <a>, para exemplificar.

Uma regra CSS é representada por um seletor ou um grupo de seletores, no nosso caso é o <a>, então dentro de um par de chaves adicionamos as declarações, no exemplo acima estamos alterando cor e tamanho da fonte dessa âncora, as declarações são formadas por uma propriedade e um valor.

*#title*, *.subtitle*, *.post\_title*{

    color: blue;

}

*.post\_title*{

    font-size: 16px;

    font-style: italic;

}

Percebam que podemos colocar vários seletores em uma regra separando-os por vírgula.

*#title*, *.subtitle*, *.post\_title*

E há um último detalhe nesse exemplo: a pseudo-classe. Elementos HTML sofrem alterações causadas pela interação do usuário, como mover o mouse por cima ou clicar nesse elemento.

O a:hover do exemplo significa que a âncora também terá essa aparência quando o usuário passar o mouse por cima de um hyperlink.

### ID x Classe

No exemplo anterior criamos uma regra que altera um elemento HTML diretamente, mas isso significa que todos os elementos <a> ficarão com aquela aparência, e normalmente temos sites mais complexos que precisam de várias regras diferentes para elementos iguais.

Para ficar mais tangível vamos relembrar um pouco o site que começamos a fazer no módulo passado, ele tinha vários elementos header, mas não vamos querer que o header principal tenha a mesma formatação que o header de uma postagem, é aí que entram os IDs e Classes.

O seletor que vimos no primeiro exemplo é um seletor de tipo, pois ele representa um elemento HTML, e com IDs e Classes podemos representar qualquer tipo de elemento, mas há algumas diferenças entre eles:

ID: é representado pelo símbolo # (hash) seguido de um nome para esse ID.

<h1 *id*="title">Leonardo Mello</h1>

*#title*

Classe: a classe é representada de forma parecida do ID, mas é precedida por um ponto em vez do hash. = .hash

<h2 *class*="subtitle">Posts</h2>

<h3 *class*="post\_title">

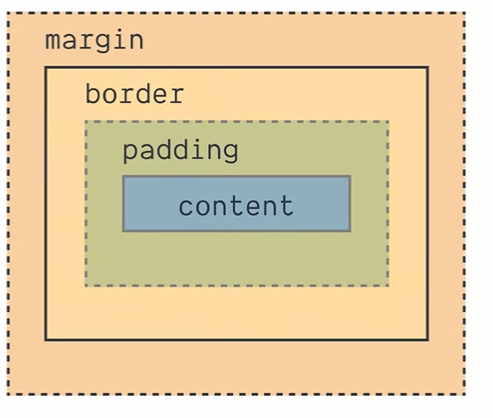
*.subtitle*, *.post\_title*

E a diferença mais importante entre eles é a forma como devem ser usados: **o ID só pode ser usado uma vez em uma página HTML** enquanto a classe não tem restrições.

### Box-Model

Quando estamos criando o layout de um site o navegador representa cada elemento HTML como uma caixa retangular, isso é o box-model. E com CSS nós alteramos a aparência dessa caixa (largura, altura, cor de fundo, etc.). Essa caixa é composta por 4 áreas: o conteúdo, o padding, a borda e a margem.

* As margens (margin) são espaçamentos entre elementos;
* As bordas (border) ;
* O padding é um espaçamento entre as bordas e o conteúdo, a diferença para as margens é que declarações de imagem de fundo funcionam nele;
* O conteúdo (content) é o que o seu bloco representa, um texto, uma imagem, um vídeo;



### Estilizando elementos

#### Padding e Margin

Anteriormente usamos o padding e o margin da forma mais básica, com apenas um valor, mas eles são mais poderosos que isso. Se quisermos atribuir tamanhos diferentes para cada lado do box nós podemos, e vamos ver três formas de fazer isso.

A primeira é colocando um valor para as partes superior e inferior e depois para os lados esquerdo e direito.

O valor de 10 pixels se refere ao eixo Y, ou partes superior e inferior, e os 5 pixels se referem aos lados esquerdo e direito.

A segunda forma é dando valores para cada lado do box.

Então começamos pelo topo com 15 pixels, passamos o lado direito com 10 pixels, depois para a parte inferior com 5 pixels e por último o lado esquerdo com 0, e sempre nessa ordem.(topo, direita, baixo, esquerda).

Uma boa dica também é que quando o valor for 0 não precisamos não precisamos colocar a unidade.

A terceira forma é com as propriedades específicas para cada lado, até agora tínhamos visto atalhos para essas propriedades.

Padding-top;

Padding-right;

Padding-botom;

Padding-left;

Essa opção é mais usada quando temos o mesmo valor para 3 lados, e o quarto precisa ter um valor diferente, então usamos o padding com apenas um valor e uma dessas opções para representar o lado diferente.

#### Background

A propriedade background também é um atalho para várias propriedades, mas isso vocês podem absorver aos poucos, e uma boa opção de leitura é a documentação do MDN.

Por enquanto veremos apenas como mudar a cor de fundo.

E aqui temos 3 formas de colocar uma cor de fundo, e ainda existem outras.

Backgroud- color;

Background-image;

Backgroud-postion;

A primeira é pelo nome da cor em inglês, a segunda é pelo código hexadecimal e a terceira é usando apenas o atalho background.

#### Border

Vimos que a propriedade border pode ter 3 valores: a largura, a cor e o estilo, mas existem algumas particularidades nisso.

A largura pode ser usada com várias unidades, como px, em e mm. A cor pode ser atribuída pelo nome ou por um código hexadecimal, assim como fizemos com o background, e o estilo é representada por palavras-chave, vamos ver algumas delas:

Também aceita estilização para cada lado.(top, right, bottom, left)

* solid: mostra uma borda simples e reta;
* dotted: são bolinhas com um pequeno espaçamento entre elas
* dashed: forma uma linha tracejada.

E aproveitando que mostrei esse código temos que falar sobre como separar a estilização dos lados de uma borda.

E se você não quiser usar a propriedade border existem as propriedade específicas para cada aspecto de uma borda, são elas border-width para a largura, border-color para a cor e border-style para o estilo.

Aqui temos o mesmo código anterior de duas formas diferentes, a primeira com o atalho border e a segunda com cada propriedade específica.

E depois disso podemos juntar os lados com os aspectos de uma borda e criar uma regra mais específica ainda.

Exemplo: Border-top-width..

##### Border-radius

E a última propriedade é o border-radius, ele permite arredondar os cantos de um elemento. Podemos usar várias unidades, mas as mais comuns são os pixels e a porcentagem.

Colocando apenas um valor mudamos todos os cantos do elemento, mas seguindo aquela mesma ordem que vimos no padding e margin - topo, direita, inferior e esquerda - conseguimos alterar cada canto separadamente.

### Estilizando Textos

#### Font-Family

Com o font-family podemos alterar a fonte dos nossos textos, como uma fonte da internet ou uma que esteja instalada no nosso computador, mas vamos nos ater às fontes seguras, chamadas de web safe fonts.

Essas fontes são chamadas assim pois são encontradas em quase todos os sistemas e podem ser usadas sem preocupação.

Exemplo:

font-family: Verdana, Geneva, Tahoma, sans-serif;

Aqui existe uma ressalva, caso você queira uma fonte backup, é só colocar o nome de outra fonte separado por vírgula, pois caso a primeira fonte não possa ser escolhida, a segunda entra em funcionamento.

#### Font-Size

O font-size nos ajuda a mudar o tamanho do texto, existem algumas unidades de medida para ele, mas por enquanto os pixels são suficientes para nós.

font-size: large;

#### Font-style

Usamos o font-style para tornar-se um texto itálico, na maioria das vezes você usará apenas o valor italic para ele, mas se precisar tirar o itálico de um texto você pode usar o valor normal.

 font-style: italic;

#### Font-weight

Altera o peso do texto, podendo deixar o texto em negrito.

Exemplo de texto em negrito:

font-weight: bold;

#### Text-Transform

Muda o texto de maiúscula para minúscula.

text-transform: capitalize;

    text-transform: lowercase;

    text-transform: uppercase;

Capitalize: deixa as primeiras letras em maiúscula;

Lowercase: deixa tudo em minúscula;

Upeercase: deixa tudo em maiúscula;

#### Text-Decoration

Muito usado para incrementar o visual do texto, pois adiciona linhas ao texto:

text-decoration: underline;

    text-decoration: overline;

    text-decoration: line-through;

Underline: Linha abaixo do texto;

Overline: Linha acima do texto;

Line-through: Linha ao centro, cortando o texto ;

### Estilo de Lista

#### List-Style-Type

Altera o símbolo ou a numeração das referências para as listas criadas.

ul{

    list-style-type: square;

}

ol{

    list-style-type: upper-roman;

}

list-style-image: url("xxxxxxxx.png");

### Dimensão e Alinhamento

Width: ajuste de largura (em pixel ou em porcentagem)

Height: ajuste de altura (em pixel ou em porcentagem)

Max-width: Largura máxima permitida

Max-height: Altura máxima permitida

Margin: Alinhamento automatico

Text align: alinhamento de texto (aceita valores como left, right, center e justify)