HTML5 && CSS3 – Alura

Parte 1 – A primeira página da Web

1. **Aula 1 – Marcação do Primeiro Texto:**
   1. Uma página web tem seu conteúdo em HTML: “Hyper text marcation language” – linguagem de marcação de texto.
   2. O formato de uma tag é *<tag>*
   3. Para utilizar uma tag, é necessário indicarmos onde se iniciam e onde terminam. Para indicar a finalização da tag, usamos ela novamente, mas somada a uma “/”, ou seja, *<h1>Título</h1>*.
   4. As tags que usaremos nesse texto base da primeira aula são:
      1. <h> (heading): Título do conteúdo – Possui 6 níveis:
         1. <h1>: Para o primeiro título da página;
         2. <h2>:
         3. <h3>:
         4. <h4>:
         5. <h5>:
         6. <h6>:
      2. <p> (paragraph): Usado para colocar parágrafos de textos.
      3. <strong> (forte/negrito): É uma tag semântica usada para deixar as palavras ou frases dentro delas com uma tonalidade mais forte, dando destaque.
      4. <em> (ênfase): Usada para dar ênfase em um parágrafo u palavra, ao utilizar essa tag, você determina que tudo o que estiver dentro está em itálico.
2. **Aula 2 – Separando o Conteúdo e Informações:**
   1. Para que o navegador renderize a nossa página da forma correta, não basta que coloquemos .html na hora de salvar o arquivo, existem outras tags de marcação que precisamos acrescentar no nosso código para que isso se torne explícito.
   2. Nós precisamos deixar claro para o navegador que estamos trabalhando cm HTML5, portanto, temos tags específicas para ter certeza que o navegador não irá interpretar a nossa página como HTML3.
   3. As tags que utilizaremos nessa aula são:
      1. <!DOCTYPE html>: Esta tag declara para o navegador que estamos utilizando a última versão do html, dessa forma ele irá interpretar a página corretamente. Não precisa ser fechada como as outras tags por não ser uma tag de conteúdo.
      2. <html>: Tag estrutural que serve para marcar tudo o que é o html que vai ser renderizado no navegador a partir dessa página. Como as outras tags de conteúdo, precisamos fechar essa tag, colocando </html> no final do documento, literalmente depois de todo o resto que está no meio da página.
         1. Dentro desta tag colocamos um parâmetro para identificar em qual idioma essa página se encontra. O parâmetro que utilizamos é o *lang=”pt-br”*. Parâmetros são colocados dentro dos <> junto com as tags correspondentes.
      3. <meta>: Tag usada para indicar qual o idioma e acentuações estamos utilizando nessa página, dessa forma, o navegador pode exibi-la corretamente independentemente do navegador e de onde a página for aberta.
         1. O parâmetro que vamos utilazar para que as acentuações fiquem corretas na página é o *charset=”UTF -8”*- este é o dicionário que estamos utilizando para a nossa página, dicionário utf-8 especificado pelo parâmetro charset.
         2. Essa tag no geral, independentemente do parâmetro que esteja junto com ela, será sempre colocada abaixo da tag <html>.
         3. Essa não é uma tag de conteúdo, portando, não precisa ser fechada em algum momento.
      4. <title>: Tag usada para determinar o título de uma página, o que fica na guia, lá no topo do navegador.
      5. <head>: Tag usada para classificar e colocar todas as tags informativas para a página, como a tag <meta>, <title>, dentre outras.
      6. <body>: Tag usada para colocar todo o conteúdo que será exibido na página, como o <h1>, <p>, dentre outras tags e conteúdos escritos.
3. **Aula 3 – Trabalhando com CSS:**
   1. Taduzindo: Folha de estilo em cascata.
   2. Para adicionar uma estilização inline, ou seja, aquela que se coloca dentro da tag específica e que será aplicada somente a ela, é necessário colocar o parâmetro *style=””;* dentro da tag desejada, como um parágrafo (<p>), por exemplo.
   3. Dentro desse parâmetro, podemos colocar a estilização desejada, como alterar o tamanho da fonte com *font-size=: ;*, por exemplo. O tamanho padrão da fonte que o navegar aplica automaticamente é 16px (pixels) que equivale a 1em (tamanho medido a partir da altura da letra “m” minúscula de uma fonte específica) ou 1ex (igual ao “em”, mas se refere à letra “x”) ou 12pt (points – usado no word), portanto, essas alturas/tamanhos podem ser alterados utilizando esse parâmetro e comando de CSS3 mostrados.
   4. Utilizando o parâmetro *text-align: ;* podemos ajustar o alinhamento de um texto. No caso de um título, podemos colocar dentro da tag <h1> o style=”text-align: center;”, ficando assim: <h1 style=”text-align: center;”>. Utilizando esse alinhamento, o título passa a ficar no centro da página, independente do tamanho da janela do navegador.
   5. Existem 3 formas de configurar o CSS:
      1. Inline: Já falado acima, onde a estilização é colocada especificamente em cada tag desejada. A técnica menos utilizada e que demanda mais trabalho para alteração de vários documentos. Definitivamente não é recomendada a sua utilização. Caso não possua mais de 1 página, ou mesmo se tiver apenas uma, utilize sempre os métodos abaixo. Deixa o código mais limpo e melhor para ler, além de facilitar qualquer tipo de alteração futura que queira fazer na sua estilização, sem grandes dores de cabeça por ter que alterar tag por tag até a finalização.
      2. Local: Uma tag <style></style> que é colcoada dentro do <head> de um <html> e lá iremos configurar cada tag do html em conjunto, ou seja, se colocarmos <style>p{text-align: center; color: blue; font-size: 20px}</style>, a alteração especificada irá acontecer em todos os parágrafos que tenham a tag <p>. Método bastante utilizado e até que recomendado caso não possua muitas páginas em um site.
      3. Externo: Toda a estilização é feita em um documento separado cuja a extensão é .css e é linkado ao documento .html pela tag <link>. Dessa forma, podemos fazer toda a estilização de um site com apenas 1 documento. Como geralmente utilizamos o mesmo padrão de estilização para todo o conjunto de páginas de um site, esse é o método mais utilizado e mais recomendado, uma vez que, caso haja a necessidade de se fazer uma alteração na estilização, não será necessário abrir o documento de todas as páginas e alterar um por um até terminar, basta alterar aquele documento .css externo que as alterações se aplicarão a todas as suas páginas web imediatamente após o salvamento.
         1. Dentro da tag <link> precisamos colocar alguns parâmetros para fazer o link com o doc .css:

1. rel=”stylesheet” (relação)

2. href=”style.css” (endereço de referência – esse endereço é o padrão, caso não esteja na mesma pasta que a página é necessário alterar. Para alterar, basta apagar tudo dentre os “” e apertar ctrl+space. Ele vai abrir uma janela onde você pode navegar e escolher a folha de estilo que deseja usar para aquele site).

* + 1. Utilizando o comando *bakcground-color: ;* nós podemos alterar a cor do fundo de qualquer coisa, desde um parágrafo/frase/título até o fundo ba página toda.
    2. Podemos alterar qualquer estilo de uma tag específica utilizando prioridades de busca em CSS. Por exemplo, se vc quiser alterar a cor de uma frase específica que está dentro de uma tag <strong> que está dentro de uma tag <em>, assim: *<em>Nossa missão é: <strong>”Proporcionar auto-estima e qualidade de vida aos clientes”</strong></em>*. Nós podemos fazer a seguinte classificação no CSS: *em strong{color: red}*. Desse modo, ele irá buscar e alterar a cor de todos os strong’s que estejam dentro dos em’s, se tiver algum fora, continuará do mesmo jeito que antes.

1. **Aula 4 - Esilizando Imagens:**
   1. Nós podemos adicionar identificadores (id=”nome”) para qualquer tag especificamente, para selecionar qualquer uma que queira e alterar seu estilo num arquivo .css separado sem modificar as outras, vamos supor que vc tenha um parágrafo e coloque um id=”missao”, quando for estilizar ele no css, basta referenciar como #missão{font-size: 20px;}, e ele será alterado para o que quiser.
   2. Para adicionar uma imagem nós utilizamos a tag <img>.
   3. Como a imagem é um arquivo externo, nós precisamos dizer a onde ela está, dessa forma, adicionando o parâmetro src=”caminho”
   4. Além do caminho, também colocamos o parâmetro alt=”descrição”, para dizer o que é essa foto, se é a foto de um banner, perfil, dentre outras coisas. Isso é importante para quando der algum problema e não for possível carregar a imagem, pois é o texto que foi colocado lá que irá aparecer no lugar.
   5. Para fazer alterações em css de uma imagem, utilizamos uma id=”nome”, colocamos #nome{} e então podemos alterar como quisermos. Obs.: O símbolo “#” é chamado de “tralha”, não jogo da velha e nem hashtag.
   6. Podemos colocar largura de uma altura em 100% para ela ocupar sempre a largura toda da tela, independentemente do tamanho da janela do navegador. Usamos *width: 100%;* pra isso.
   7. Podemos criar uma borda em qualquer elemento usando a marcação *border: tamanho tipo cor*, exemplo: border: 10px solid black. A borda é a finalização do elemento.
   8. O espaçamento interno, que fica entre a bora e o conteúdo do elemento (seja uma foto, ou texto) é chamado de *padding* e podemos criar vários tipos de espaçamentos, somente em cima, somente em baixo, na direita ou esquerda, além é claro, de em todos os lados ao mesmo tempo. Por exemplo: padding: 20px;. Com essa marcação, o nosso elemento teria um respiro de 20 pixels entre a borda e o texto digitado em todas as direções. Para criar em direções específicas basta colocar *-local*. Exemplo: padding-top: 20px; - cria um espaçamento entre o conteúdo e a borda de 20 pixels na parte superior da box. O mesmo ocorre se colocar -bottom, -left e -right.
   9. O espaçamento inter, que fica entre a box e o resto do conteúdo da página, é chamada de *margin*, literalmente uma margem entre a box e o conteúdo. O esquema de direcionamento e tamanho funciona igual ao *padding*, usando top, left, 20px, 60px, etc...
2. **Aula 5 – Listas e divisões de Conteúdo:**
   1. Existem duas tags para criação de listas:
      1. <ol> (ordered list): Listas ordenadas, ou seja, numeradas, ou alfabéticas ou por números romanos todos em letras maiúsculas ou não.
      2. <ul> (unordered list): Listas não ordenadas, onde não tem importância de numeração e qual item vem depois, sendo marcadas por bolinhas, discos e quadrados.
   2. Independente de qual lista usar, a tag precisa ser fechada.
   3. Para criar itens dentro das listas, independentemente de qual esteja usando, precisa utilizar a tag <li> (list iten) que pode ou não ser fechada ao final do item ao qual se refere, nesse caso, o fechamento da tag fica ao seu critério.
   4. Para criar uma estilização para as listas, nós precisamos classificar os itens. É como colocar um id=”nome”, mas funciona de melhor maneira para listas. O parâmetro (ou propriedade) que utilizamos é *class=”nome”*. Exatamente igual ao *id=”nome”*, só muda o parâmetro. No caso de listas precisamos colocar o mesmo nome para todos os itens, dessa forma, podemos alterar todos de uma mesma maneira.
   5. Ao contrário do identificador, onde no css usamos a tralha (#), para nos referirmos ao item, no caso das classes, utilizaremos um ponto (.) e o nome dado para a classe. Ex.: .itens {} (referenciando uma classe em css) e #itens {} (referenciando um identificador em css).
   6. Para deixar um elemento (ou uma classe deles) em itálico, podemos utilizar o marcador *font-style> italic;*.
   7. Nós podemos trabalhar com divisões em uma página para podermos classificar diferentes conteúdos, para isso utilizamos a tag <div>
      1. Elas não interferem no visual ou no conteúdo da página, elas servem apenas para marcar o conteúdo e a partir do css fazer os efeitos desejados.
      2. Como todas as div’s possuem o mesmo nome, podemos diferenciá-las criando id ou class’s para cada uma especificamente.
      3. Identificador serve para itens únicos e as classes para itens que vão ser repetidos. Porém, se quisermos criar um layout robusto com todas as técnicas e quisermos replicar esse conteúdo no futuro, o ideal é criarmos classes.
      4. Sempre que quisermos marcar um conteúdo visualmente, vamos criar uma classe para ele.
   8. Conteúdo block, conteúdo inline e conteúdo inline block:
      1. Block: Quando um texto ou elemento ocupa 100% do espaço da página, não permintindo que nada mais fique na sua lateral. É o que ocorre com as listas, parágrafos dentre outras tags.
      2. Inline: É o que acontece com imagens. Elas não bloqueiam o resto da página, permitindo que mais conteúdo possa aparecer e ser exibido na mesma linha que ela está, porém, não permite que eu altere espaçamento lateral, interno, externo dentre outros.
      3. Inline block: Quando a tag possui as duas características, ela vai bloquear tudo na sua lateral, mas é de um tamanho fixo, ou seja, eu posso alterar o tamanho do block, espaçamento interno e exerto, dentre outros, deixando assim, que mais conteúdo caiba na sua lateral.
      4. Todas essas são características da marcação *display: ;*, desse modo, podendo ser alterado em css. Pode-se fazer uma marcação de display para as <ul> e colocar ele como qualquer uma das características anteriores, ou seja: display: block/inline/inline-block;
   9. Alinhamento vertical:
      1. Em css o alinhamento vertical sempre é na parte inferior do conteúdo, para alterar isso, basta colocar o *vertical-align: top;*, assim, o conteúdo vai ser alinhado na parte superior.
3. **Finalizando a Página:**
   1. O cabeçalho de uma página possui a própria tag, que nada mais é que uma div, mas possuí sua semântica de cabeçalho. A tag é <header></header>.
   2. O ideal é utilizarmos class para todas as estilizações, porque se por acaso quisermos alterar um título e substituir por outro, vai acabar com o código, ficar uma bagunça. Portanto, sempre que for fazer uma estilização, coloque class específica para aquilo.
   3. É interessante colocar um padding-left em títulos que não forem centralizados na página, dessa forma, existe um respiro para ficar visualmente bonito.
   4. O título é sempre a coisa de mais destaque e que aparece primeiro na página.