

HTML e CSS

Conceitos e fundamentos do desenvolvimento web

Por Heloísa Martins

Composição Front-End

As muitas partes do todo

Quando falamos em *front-end*, estamos nos referindo à parte do produto que interage com o usuário: é a fachada, ou melhor, a página de apresentação daquilo que queremos exibir - dados, informações, links, operações, etc.

O foco do desenvolvedor, quando trabalha no front-end, é criar uma página **organizada**, que promova **clareza das informações apresentadas**. Além disso, a página precisa estar **apresentável**, com um **visual agradável e de acordo com os padrões do produto**. Por fim, deve ser **funcional** e ter um **propósito no contexto da solução**.

Organização, apresentação e funcionalidade são três partes do nosso todo. Cada uma é construída com uma ferramenta diferente, mas são unidas com o propósito de compor a nossa página web.



HTML



JavaScript



CSS

HTML: A estrutura e o esqueleto

HTML - Hyper-Text Markup Language

HTML é a linguagem de marcação de hipertexto utilizada para dar estrutura ao conteúdo apresentado numa página. É o esqueleto que sustenta tudo aquilo que nós vemos e interagimos.

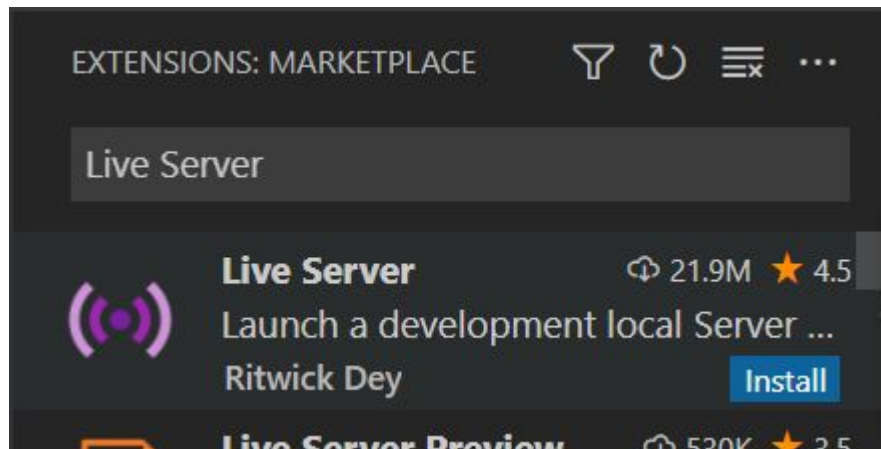
Não é uma linguagem de programação: quando trabalhamos com HTML não há uma construção de lógica, não são realizadas operações de cálculo explícito e nem há armazenamento de dados brutos. O propósito do HTML é servir como fundação para aquilo que queremos mostrar na página!

Para conhecermos como uma página HTML é basicamente construída, vamos testar na prática!

- Abra o Visual Studio Code;
- Barra superior “File” -> “Open Folder...”;
- Crie um diretório para as nossas aulas;
- Depois, no explorador do VS Code, clique em “New File...”;
- Crie um arquivo chamado “index.html”.

O arquivo da sua primeira página web está pronto para ser editado e visualizado no navegador que estiver utilizando!

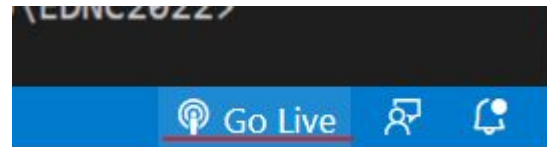
Dica: adicione a extensão “Live Server” para facilitar a visualização da página!



No VS Code, ao lado direito, abra o explorador de extensões e pesquise por “**Live Server**”. Clique em “**install**” para que seja instalado no seu Visual Studio Code.

Essa extensão permite a simulação de um “servidor” em tempo real que facilita a visualização das edições feitas no documento html. Sempre que você salvar as alterações com “ctrl + s” ou outro atalho, a página será atualizada automaticamente, sem precisar recarregar na mão!

Para ativar o live server, você precisa estar em um documento HTML com a estrutura básica já montada. Nessas condições, clique em “Go Live” na barra inferior do Visual Studio Code para abrir a visualização.



A estrutura básica de um documento HTML

Quando trabalhamos com uma página nova, ou um documento *fresco*, é sempre interessante escrever as definições do documento para que o navegador possa tratar corretamente de algumas características.

Ao lado, podemos ver como é a “base” dessa definição. Temos a marcação principal do documento, com a tag `<!DOCTYPE>` que descreve que estamos trabalhando com um arquivo html.

Em seguida, a tag `<html>`, que é a raiz de todo o nosso documento. Todos os elementos devem ser descendentes desta tag para fazerem parte do mesmo documento! Além disso, ela pode trazer algumas informações através de atributos, como a “`lang`” que descreve o idioma geral do documento.

O nosso documento é dividido, geralmente, em duas partes principais: `head` e `body`.

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <title>Titulo da Página</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

A estrutura básica de um documento HTML

Head é a área que contém informações adicionais sobre o documento, não sendo visíveis diretamente na página web. Estas informações serão lidas pelo navegador e irão modificar características como o título da página na web, o conjunto de caracteres utilizado, realizar links com folhas de estilos, definir scripts e outros metadados customizáveis.

<meta charset="UTF-8"> : elemento que representa um metadado. A propriedade charset faz a definição do padrão de caracteres utilizado neste documento (UTF-8 é o conjunto padrão, capaz de representar todos os símbolos Unicode).

<title>Sample Text</title> : elemento que faz com que o título da página web, no navegador, seja exibido de modo customizado.

Body é a área que contém as informações a serem exibidas na página web. É aqui que vamos incluir as nossas regiões, textos, botões, imagens, áreas de entrada, etc. É a parte onde trabalharemos a maior parte do tempo, realizando a inclusão e a manutenção constante dos elementos.

Elementos HTML: Estruturais

Elementos HTML: Estruturais

Podemos chamar de “elementos estruturais” as tags html que definem áreas e escopos da página. Um cabeçalho, um rodapé, uma sessão, uma coluna, a região de um texto, dentre outras áreas de uma página, são divididas dentro do documento por estes elementos estruturais.

Antes do HTML5 não havia a variedade de tags estruturais que temos hoje; por este motivo, muitas soluções e páginas que vemos na web são organizadas majoritariamente por um único tipo de tag estrutural chamada `<div>`.

Elementos HTML: Estruturais

<header></header> - Representa a região de um cabeçalho. Não confunda com a tag <head> utilizada no topo do documento html.

<footer></footer> - Representa a região de rodapé da página.

<nav></nav> - Utilizado para representar a área de barra de navegação.

<main></main> - Utilizado para representar uma área principal da página (como o container dos demais conteúdos) ou para representar uma área principal dentro de algum outro escopo (como o corpo de texto em um article, por exemplo)

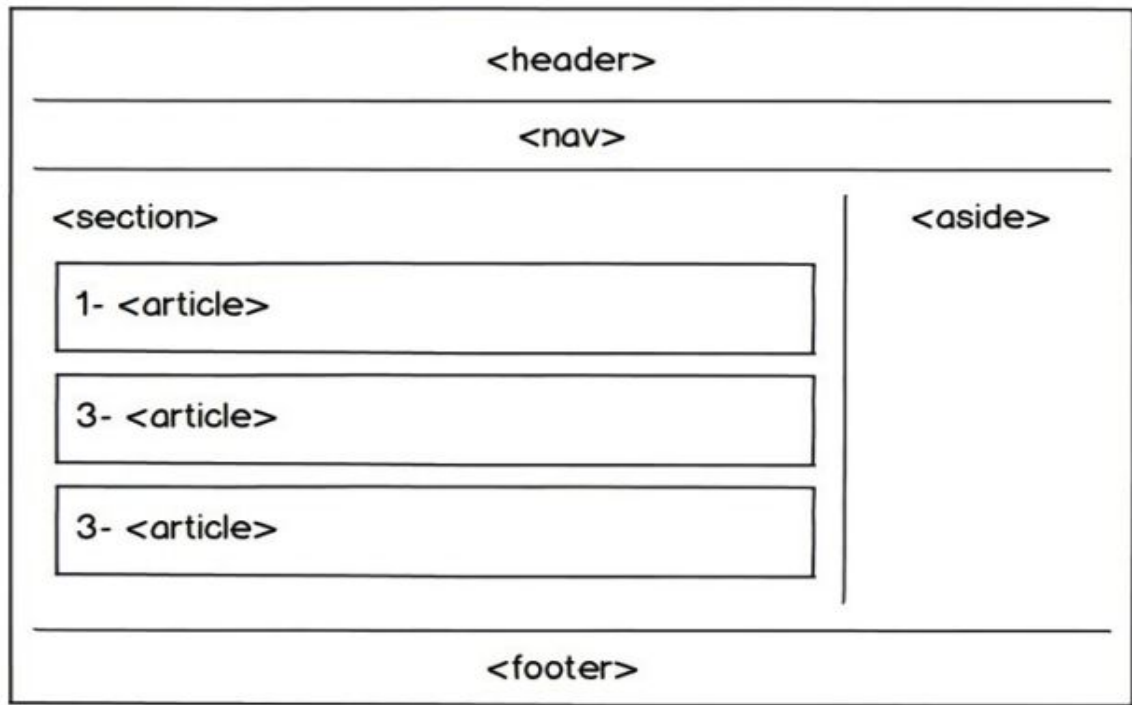
<section></section> - Representação de uma sessão da página.

<article></article> - Representa um trecho único dentro de uma sessão da página (por exemplo um banner de manchete individual na sessão de notícias)

<aside></aside> - Representa uma área de conteúdo relacionado à página mas que possui tangência suficiente para ser separado.

<div></div> - Estrutura primária de divisão de áreas. Pode ser utilizada com qualquer propósito organizacional.

Elementos HTML: Estruturais



Com a chegada do HTML5 as páginas se tornaram muito mais organizadas, tanto para a leitura do navegador quanto para os desenvolvedores que trabalham sobre seu código. Apesar da organização se tornar mais fácil e ser uma tendência, não há mal nenhum em utilizar apenas a tag `<div>` para realizar a construção dessas áreas. Tudo depende das convenções utilizadas no projeto em que você trabalha e na disposição da equipe em aplicar novas convenções.

Elementos HTML: Conteúdo

Elementos HTML: Conteúdo

`<h1></h1>` à `<h6></h6>` - Representam um conteúdo de título, de maior hierarquia até a menor hierarquia. É recomendado que a página possua apenas um `<h1>`, pois define um título principal utilizado pelos mecanismos de busca para identificar o contexto daquela página ou site.

`<p></p>` - Representação de um parágrafo/conteúdo de texto extenso. Muito utilizado para grandes blocos de informação.

`` - Possui a mesma funcionalidade do parágrafo, porém, é utilizado geralmente para textos pequenos, legendas, informações de tooltip, etc.

`<pre></pre>` - Conteúdo pré-formatado. Segue exatamente a indentação que for feita no conteúdo do arquivo html. Muito utilizado para inserir trechos de código.

`` ou `*` - Faz com que o conteúdo fique em negrito.

`<i></i>` ou `*` - Faz com que o conteúdo fique em itálico.

`
` - Tag não precisa de um fechamento. Insere uma quebra de linha.

`<hr>` - Tag não precisa de um fechamento. Insere uma linha horizontal.

Atividade de Teste 1

Reproduzindo um
esqueleto simples

Atividade Teste 1: Reproduzindo um esqueleto simples

<code><header></code>
<code><nav></code>
<code><section></code>
1- <code><article></code>
2 - <code><article></code>
3- <code><article></code>
<code><footer></code>

Reproduza o protótipo ao lado e gere uma página de testes para aplicar o que foi visto até agora!

Dica: Por enquanto, utilize a propriedade `style="text-align:center"` para centralizar os elementos que você deseja.

Se precisar de texto para preencher as regiões da página, acesse <https://www.lipsum.com/> para obter o texto exemplo necessário.

Página de Testes

[Link exemplo 1] - [Link exemplo 2] - [Link exemplo 3]

Sessão de Notícias mais recentes

Titulo da primeira manchete

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus dapibus elit at metus efficitur, vel tempor urna pharetra. Suspendisse quis ante ultricies, vestibulum lorem ut, finibus nibh. Duis egestas mi sit amet libero scelerisque, sit amet congue libero scelerisque. Morbi et nulla tincidunt, laoreet lectus quis, pulvinar mi.

Titulo da segunda manchete

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus dapibus elit at metus efficitur, vel tempor urna pharetra. Suspendisse quis ante ultricies, vestibulum lorem ut, finibus nibh. Duis egestas mi sit amet libero scelerisque, sit amet congue libero scelerisque. Morbi et nulla tincidunt, laoreet lectus quis, pulvinar mi.

Titulo da terceira manchete

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus dapibus elit at metus efficitur, vel tempor urna pharetra. Suspendisse quis ante ultricies, vestibulum lorem ut, finibus nibh. Duis egestas mi sit amet libero scelerisque, sit amet congue libero scelerisque. Morbi et nulla tincidunt, laoreet lectus quis, pulvinar mi.

Acesse as nossas redes sociais!
Instagram - Facebook - Twitter

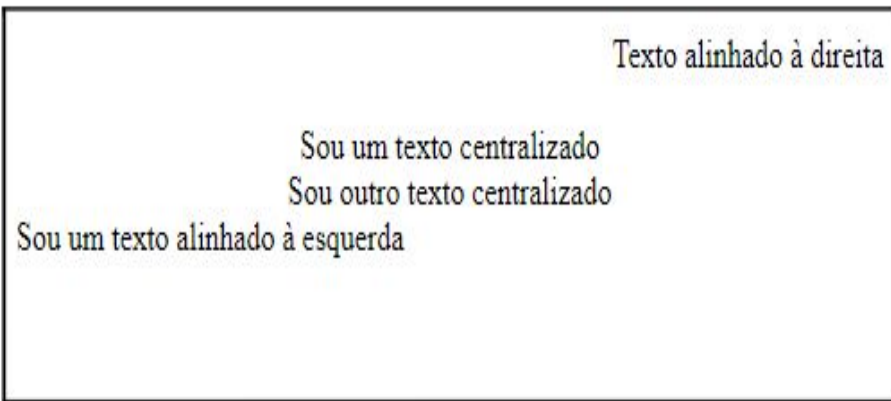
E se usarmos
apenas `<div>` ao
invés de usar
`<header>`, `<article>`,
`<section>`... ?

Um adendo sobre hierarquias e propriedades!

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Titulo da Página</title>
6  </head>
7  <body>
8  |   <section style="text-align:right">
9  | |   Texto alinhado à direita
10 | </section>
11 | <br>
12 | <section style="text-align:center">
13 | |   <article>
14 | | |   Sou um texto centralizado
15 | | </article>
16 | |   <article>
17 | | |   Sou outro texto centralizado
18 | | </article>
19 | |   <article style="text-align:left">
20 | | |   Sou um texto alinhado à esquerda
21 | | </article>
22 | </section>
23 </body>
24 </html>
```

Elementos podem - e devem - ser aninhados em uma hierarquia que faça sentido. Existem elementos html específicos para definir estruturas, áreas e pseudo-elementos num documento.

Quando estes elementos possuem características definidas, seus “filhos” poderão herdar estes mesmos valores, desde que não possuam um valor diferente definido individualmente.



Resultado do código à esquerda

Elementos HTML: Listas e Tabelas

Elementos HTML: Listas e Tabelas

Listas podem ser construídas de maneira ordenada ou não ordenada, através das tags `` e `` respectivamente. Estas tags definem que será gerada uma lista com seus elementos filhos.

Cada item da lista deve ser representado por um elemento ``, independente do tipo da lista sendo utilizada.

O navegador, por padrão, trata as duas listas de maneiras diferentes em questão do visual e da leitura para construção da página. Experimente o exemplo ao lado e veja como fica!

```
<> listastabelas.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Listas e Tabelas</title>
6  </head>
7  <body>
8  |   <div>
9  |       Lista Ordenada
10 |       <ol>
11 |           <li>Item um</li>
12 |           <li>Item dois</li>
13 |           <li>Item três</li>
14 |           <li>Item quatro</li>
15 |       </ol>
16 |   </div>
17 |   <div>
18 |       Lista não Ordenada
19 |       <ul>
20 |           <li>Item um</li>
21 |           <li>Item dois</li>
22 |           <li>Item três</li>
23 |           <li>Item quatro</li>
24 |       </ul>
25 |   </div>
26 </body>
27 </html>
```

Elementos HTML: Listas e Tabelas

Tabelas são construídas a partir do elemento principal `<table></table>`, que vai definir o grupo de informações a seguir como conteúdo de uma tabela. Existem três elementos html que existem com o propósito de estruturar e agrupar os dados corretamente, entre conteúdo do cabeçalho, das linhas e do rodapé: são, respectivamente, as tags `<thead></thead>`, `<tbody></tbody>` e `<tfoot></tfoot>`.

Dentro desses elementos de grupo nós realizamos a definição das linhas e do conteúdo de suas colunas, através das seguintes tags:

`<tr></tr>` - “Table row”, define o conjunto de informações que compõem uma única linha;

`<th></th>` - “Table header cell”, define o conteúdo de uma única célula do cabeçalho da tabela. Deve ser utilizado, de preferência dentro de `<thead></thead>`;

`<td></td>` - “Table data”, define o conteúdo de uma única célula dentro de uma linha da tabela.

Elementos HTML: Listas e Tabelas

Lembre-se que tanto tabelas como listas são elementos que possuem uma hierarquia direta entre seus elementos filhos. O que for aplicado no elemento superior poderá cascatear para os elementos mais abaixo na hierarquia!

```
27 <div>
28     Exemplo de Tabela
29     <table>
30         <thead>
31             <tr>
32                 <th>Nome</th>
33                 <th>Idade</th>
34                 <th>Equipe</th>
35             </tr>
36         </thead>
37         <tbody style="text-align:center">
38             <tr>
39                 <td>Gustavo</td>
40                 <td>23</td>
41                 <td>Dev. Mobile</td>
42             </tr>
43             <tr>
44                 <td>Ana Clara</td>
45                 <td>21</td>
46                 <td>Dev. e-Commerce</td>
47             </tr>
48             <tr>
49                 <td>Jessica</td>
50                 <td>21</td>
51                 <td>Testes e-Commerce</td>
52             </tr>
53         </tbody>
54     </table>
55 </div>
```

Elementos HTML: Input

Elementos HTML: Input

Provavelmente o elemento html mais importante para soluções e projetos que envolvem interação com o usuário; o elemento `<input></input>` permite a criação de um “campo” de formulário que recebe dados em diversos formatos. Este elemento possui uma propriedade chamada “type” que pode modificar suas características visuais, bem como o modo de interação com o usuário.

Existem 22 tipos diferentes de input que podem ser utilizados, sendo “text” o tipo padrão. A documentação para estes inputs pode ser encontrada em https://www.w3schools.com/tags/tag_input.asp .

Elementos HTML: Multimídia e Links

Elementos HTML: Multimídia e Links

Outro elemento muito importante na construção de páginas web é o elemento `<a>`, conhecido também por “anchor”. Como o próprio nome já diz, é uma “âncora” para um link ou uma referência, capaz de redirecionar a página para outro domínio. Podemos fazer isso atualizando a página atual ou abrindo em uma nova aba! É possível, com um pouco de javascript, fazer com que o link seja aberto em uma nova janela.

A tag `<a>` possui a propriedade `href`, que permite com que seja definido o link relacionado a ela. Além disso, a propriedade `target` nos permite decidir se o link será aberto em uma nova aba, definindo seu valor para “`_blank`”

Elementos HTML: Multimídia e Links

Podemos realizar a inserção de imagens, sons e até mesmo vídeos na página através de alguns elementos.

`` é a tag que define uma imagem na página. Através da propriedade `src` definimos o diretório dessa imagem que seja inserida na tela. O diretório e o nome do arquivo devem ser escritos corretamente para que a imagem seja exibida.

Com as propriedades `width` e `height` podemos ajustar o tamanho da imagem. Se não forem definidas estas propriedades, a imagem será inserida em seu tamanho nativo.

Com a propriedade `alt` nós definimos um texto que será lido por mecanismos de acessibilidade, além de proporcionar um “placeholder” para caso a imagem não seja carregada de maneira adequada.

Elementos HTML: Multimídia e Links

Os elementos `<audio></audio>` e `<video></video>` podem ser utilizados para realizar a inserção de conteúdos de... áudio e vídeo! Quando escrevemos estas tags temos a opção de informar o atributo “controls” para que um player seja exibido na página. Quando omitido, nenhum conteúdo é apresentado, porém, o recurso fica disponível para outros usos.

A fonte deve ser inserida num elemento chamado `<source>`, com as propriedades `src` e `type`, definindo respectivamente o diretório do recurso e o tipo do arquivo.

```
<> multimidia.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <title>Multimida e links</title>
6  </head>
7  <body>
8      <audio controls>
9          <source src="audio.mp3" type="audio/mpeg">
10     </audio>
11     <video controls>
12         <source src="video.mp4" type="video/mp4">
13         <source src="video.mp4" type="video/ogg">
14     </video>
15
16     <!-- O audio abaixo não aparece na tela com o player -->
17     <audio>
18         <source src="audio0culto.mp3" type="audio/mpeg">
19     </audio>
20 </body>
21 </html>
```

Atividade de Teste 2

O concurso

CSS: Folhas de estilo em cascata

CSS - Cascading Style Sheets

Enquanto usamos HTML para definir a estrutura das nossas páginas podemos utilizar CSS para dar cor, forma, estilo e um toque a mais de organização para nossos elementos! As folhas de estilo em cascata são utilizadas através de arquivos secundários de extensão .css e são “invocados” através da tag `<link>` nos metadados do arquivo html: a propriedade `href` indica o diretório e o nome do arquivo da folha de estilos e a propriedade `rel` indica o tipo de relacionamento (“*stylesheet*” é a definição para um relacionamento de folha de estilos).

```
3 <head>
4   <meta charset="UTF-8">
5   <title>Titulo da Página</title>
6   <link href="style.css" rel="stylesheet" >
7 </head>
```


Propriedade “class”: aplicando estilos

Podemos informar, nos elementos do nosso arquivo html, a propriedade “class”, que indica o nome da classe que será utilizada para estilizar este mesmo elemento. Um mesmo elemento pode receber várias classes de css para alterar suas características.

Uma classe css é um conjunto de propriedades de estilo, escritas no arquivo .css com o prefixo de um ponto. Segue-se com o nome da classe (ou das várias classes que irão receber aquelas propriedades) e as suas respectivas características dentro de chaves. As propriedades são separadas por ponto e vírgula.

```
9      <div class="minhaClasse">
10      |   <p>Olá mundo!</p>
11      </div>
```

```
# style.css > ...
1      .minhaClasse{
2      |   border-width: 1px;
3      |   border: solid;
4      |   border-color: blue;
5      |   width: 200px;
6      | }
```



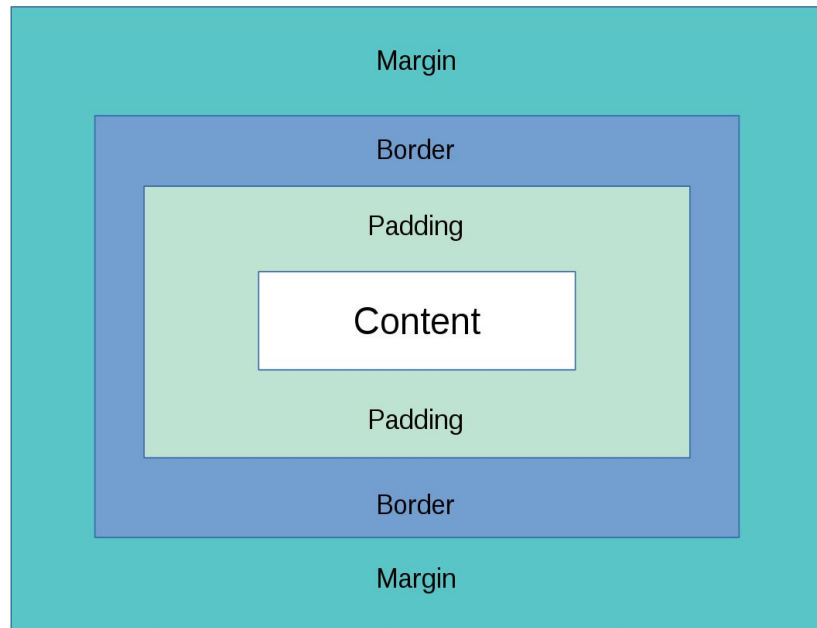
Olá mundo!

Box model: como os elementos se comportam

Podemos entender por “box model” o conceito de que todo elemento html está inserido dentro de uma espécie de *caixa*. Esta caixa possui áreas que definem espaçamentos como margem, preenchimento e conteúdo.

A propriedade **margin** define um espaço que é tomado antes de inserir o elemento e, dependendo de como for valorada, o espaço após este elemento. Cria um espaço vazio entre o elemento inserido e os outros elementos ao redor.

A propriedade **padding** define um espaçamento interno do elemento. Enquanto a margin coloca um espaço na parte externa do elemento, o padding coloca esse espaço para dentro, entre o conteúdo e a borda.



Quantas propriedades existem no CSS?

É muito difícil, no começo do desenvolvimento, saber quais propriedades existem disponíveis, quais devemos usar, quais nos ajudam em determinada situação, etc. A quantidade de propriedades que podem ser aplicadas, atualmente, é enorme: são **228** propriedades listadas pela **W3Schools**.

Portanto, torna-se quase impossível decorar todas elas logo de começo.

Mas não desanime! Existem algumas propriedades que você verá constantemente e, com o tempo, você consegue aprender quais são as mais relevantes, quais técnicas aplicar e como lidar com situações no dia-a-dia de trabalho. **A documentação é a sua melhor amiga** - basta ser persistente e ter foco no que é necessário no momento!

Margins e Paddings

As propriedades de **margin** e **padding** podem ser escritas de várias formas - como se elas possuíssem um tipo de sobrecarga. Qualquer um dos quatro padrões abaixo podem ser utilizados para realizar a definição de espaçamento das margens ou dos paddings:

1 único valor: A propriedade é aplicada para os 4 lados do elemento.

```
margin: 10px;  
margin: 0;  
padding: 5%;
```

2 valores: Aplica respectivamente para o eixo x (topo e inferior) e para o eixo y (esquerda e direita).

```
margin: 10px 15px;  
padding: 0 5%;
```

3 valores: Aplica a propriedade respectivamente para o topo, para o eixo y e para a inferior.

```
margin: 10px 5px 10px;  
padding: 2% 0 2%;
```

4 valores: Aplica os valores no sentido horário, começando pelo topo (topo, direita, inferior e esquerda).

```
margin: 10px 5px 10px 0  
padding: 2% 5px 2% 10px
```

A propriedade “color” e derivadas

Afeta a cor dos textos inseridos no elemento e no texto de seus elementos filhos - parágrafos, span, títulos e texto livre são afetados por esta propriedade. Pode ser escrita, também, em diversos padrões:

Através do nome literal das cores padrões:

```
color: red;  
color: lightgray;  
color: gold;
```

Pelo código hexadecimal da cor:

```
color: #FF0000;  
color: #D3D3D3;;  
color: #FFD700;
```

Pelo código RGB ou RGBA da cor:

```
color: rgb(255, 0, 0)  
color: rgb(211, 211, 211))  
color: rgb(255, 215, 0)  
color: rgba(255, 0, 0, 30%)  
color: rgba(255, 0, 0, 0.3)
```

Através do código HSL ou HSLA da cor:

```
color: hsl(0, 100%, 50%)  
color: hsl(0, 0%, 83%)  
color: hsl(51, 100%, 50%)  
color: hsl(0, 100%, 50%, 30%)  
color: hsl(0, 100%, 50%, 0.3)
```

Dica: O Visual Studio Code facilita a conversão de cores de um valor para o outro através da própria interface! Basta clicar no nome que fica sobre a paleta de cores, quando estiver com o mouse sobre o nome da cor.

A propriedade “color” e derivadas

Outras “sub-propriedades” do CSS também interagem com cores, e podem utilizar das mesmas formas de escrita da propriedade color. Abaixo alguns exemplos de propriedades que interagem com mudança de cor:

background-color: rgba(31, 18, 212, 0.3);

caret-color: hls(190, 90%, 35%);

outline-color: #000000;

border-color: red;

Tudo em apenas uma linha

Algumas propriedades são “compostas” por pequenas partes - por exemplo, podemos dividir a propriedade `margin` em quatro outras propriedades que definem cada um de seus lados: `margin-top`, `margin-bottom`, `margin-left` e `margin-right`. Outras propriedades permitem a edição de suas várias partes em apenas uma linha, não apenas em questão de orientação. O nome dessa característica, de definir tudo em apenas uma linha é chamada de “shorthand”:

`border: [border-width] [border-style (obrigatório)] [border-color];`

`border: 5px dashed #0000ff;`

`border: 3px solid;`

`border: 1px solid red;`

Atividade de Teste 3

Aplicando os
primeiros estilos
para o concurso