Sumário

[Colocar um texto em negrito: 3](#_Toc95034786)

[Colocar um texto em itálico: 3](#_Toc95034787)

[Diminuir o texto 3](#_Toc95034788)

[Sublinhando o texto 3](#_Toc95034789)

[Quebra de linhas 3](#_Toc95034790)

[Listas 3](#_Toc95034791)

[Listas não enumeradas 3](#_Toc95034792)

[Listas numeradas 4](#_Toc95034793)

[Lista de definições 4](#_Toc95034794)

[Uso da imagem 5](#_Toc95034795)

[Favicon 6](#_Toc95034796)

[Código 6](#_Toc95034797)

[Links 7](#_Toc95034798)

[Modificando a fonte, cor e tamanho 9](#_Toc95034799)

[Tabelas 9](#_Toc95034800)

[Div e span 11](#_Toc95034801)

[Criando páginas com quadros 12](#_Toc95034802)

[Formulário 13](#_Toc95034803)

[Lista suspensa 13](#_Toc95034804)

[Caixa de texto 13](#_Toc95034805)

[Pequena caixa de texto 13](#_Toc95034806)

[Abreviações 15](#_Toc95034807)

[Header, section and footer 15](#_Toc95034808)

[Vídeos/Áudios 15](#_Toc95034809)

[CSS 16](#_Toc95034810)

[Alterar fundo e cor da letra 17](#_Toc95034811)

[Distanciamento de palavras/letras/linhas e alinhamento de texto 19](#_Toc95034812)

[Imagem de fundo 20](#_Toc95034813)

[Bordas 20](#_Toc95034814)

[Espaçamento e margem 21](#_Toc95034815)

[Dimensões 21](#_Toc95034816)

[Posição: Estático, relativo e absoluto 21](#_Toc95034817)

[Flex-container 23](#_Toc95034818)

[Flex-display 25](#_Toc95034819)

[Flex justify contente 28](#_Toc95034820)

Alguns comandos que estou aprendendo em HTML:

# Colocar um texto em negrito:

A tag que deve ser atualizada para negrito é a tag <Strong>.

# Colocar um texto em itálico:

A tag que deve ser atualizada para itálico é a tag <em>.

# Diminuir o texto

A tag para diminuir o texto é a tag <small>

# Sublinhando o texto

Para sublinhar uma parte do texto usando HTML, devemos somente inserir o texto correspondente dentro de um par de tags tipo <u>.

   <em>Italico</em>

    <br>

    <strong> Negrito</strong>

    <br>

    <small>Pequeno</small>

    <br>

    <u>Sublinhado</u>

# Quebra de linhas

Temos a quebra de linha como já conheço, o <br>, o livro recomenda utilizar <br/>. Utilizando o comando <hr/>, podemos ter uma quebra de linha, e o desenho de uma linha na web.

# Listas

## Listas não enumeradas

Para descrever os dois tipos de tags que devem ser utilizadas na construção de uma lista veremos – e não podia ser de outra forma - a seguinte lista:

Para delimitar o começo e o final de uma lista, isto é, a área compreendida entre o inicio do primeiro dos elementos e o final do último elemento, são utilizadas as tags <ul> e </ul>.

É denominada conforme o seu nome em inglês: unordered list (lista não enumerada).

Para delimitar cada elemento da lista de forma individual, são utilizadas as tags <li> e </li>. O nome desta outra tag proveem de “list ítem” (elemento de lista).

Para exemplificar, é utilizado a tag <ul> e o seu final </ul> para marcar a área que será feito a lista. E depois, cada item que será da lista, é colocado da forma <li> e fechado com </li>. Podemos colocar o type, são 3 tipos disc - padrão. Uma bola preta totalmente pintada circle - Uma bola com uma borda preta e sem preenchimento square - Um pequeno quadrado preto totalmente pintado. Exemplo:

<ul>

<li> item 1 </li>

<li> item 2 </li>

</ul>

## Listas numeradas

Para descrever os dois tipos de tags que devem ser utilizadas na construção de uma lista veremos – e não podia ser de outra forma - a seguinte lista:

Para delimitar o começo e o final de uma lista, isto é, a área compreendida entre o inicio do primeiro dos elementos e o final do último elemento, são utilizadas as tags <ol> e </ol>.

É denominada conforme o seu nome em inglês: unordered list (lista não enumerada).

Para delimitar cada elemento da lista de forma individual, são utilizadas as tags <li> e </li>. O nome desta outra tag proveem de “list ítem” (elemento de lista).

Para exemplificar, é utilizado a tag <ol> e o seu final </ol> para marcar a área que será feito a lista. E depois, cada item que será da lista, é colocado da forma <li> e fechado com </li>(Não é obrigadtorio fechar). Exemplo:

<ol>

<li> item 1 </li>

<li> item 2 </li>

</ol>

O podemos colocar características no ol, seriam type e start. Type é o tipo de lista que estou criando e start é o número que quero que comece.

<ol type =”I” start=”5”>

## Lista de definições

A diferença desta para as anteriores consiste em que cada um dos seus elementos tem um título próprio com a explicação (ou definição) do mesmo na sequência.

As listas de definições podem ser construídas mediante a combinação de três pares distintos de tags, que são:

Para delimitar o começo e o final de uma lista, isto é, a área compreendida entre o inicio do primeiro dos elementos e o final do último elemento, se utilizam as tags <dl> e </dl>. O nome desta tag está relacionado com seu nome em idioma inglês: definition list (lista de definições).

Para identificar cada um dos títulos da lista, devem ser usadas as tags <dt> e </dt>, cujos nomes proveem do inglés definition term (termo definido).

Logo após cada termo definido com as tags anteriores, se deve colocar sua respectiva explicação usando as tags <dd> e </dd>, as quais proveem do termo definition description (descrição da definição).

Essa lista é mais complicada que as anteriores, para delimitar a área que a lista vai ocupar vamos utilizar a tag <dl>, para nomear os títulos devemos usar a tag <dt> e o conteúdo da lista deve utilizar a tag <dd>. Exemplo:

<dl>

<dt> Titulo 1 </dt>

<dd> Conteudo </dd>

</dl>

# Uso da imagem

<img src = 'boku.png' border = '2' height="1000"/>

O uso de imagem, deve ser utilizado a tag img, além disso, o scr que é para a imagem aparecer e posso modificar da forma que eu quero utilizando o:

border: Indica a grossura (largura) da borda a ser mostrada ao redor da imagem.

width: Indica o largura em pixels em que se deve exibir a imagem.

height: indica a altura em pixels da imagem a ser mostrada.

Alt : Para descrever a imagem, bom para pessoas que tem deficiência visual, para que possam entender através da descrição.

Pode ser utilizado três tipos de comandos para inserir uma imagem.

A primeira a imagem está na mesma pasta que o programa está sendo salvo.

A segunda está em outra pasta.

E a última, é o link do site que onde se encontra a imagem.

<p> Se a imagem estiver na mesma pasta</p>

    <img src="boku.jpg" alt="Imagem do Anime Boku">

    <p> Se a imagem estiver dentro de uma pasta</p>

    <img src="Imagem/boku.jpg" alt="Imagem do Anime Boku">

    <p> Imagem externa</p>

    <img src="https://images-na.ssl-images-amazon.com/images/I/715BlNAXJfL.jpg" alt="Imagem do anime Boku">

No html5, foi criado mais duas tags que podem ser usadas com a tag de imagem, é a tag <figure> e a tag <figcaption>. A tag <figure> delimita a área que a tag <img> vai estar, e a tag <figcaption> é a legenda que a figura deve ter.

    <figure>

        <img src = 'boku.jpg' border = '2' height="1000"/>

        <figcaption>Imagem do anime boku no hero</figcaption>

    </figure>

Utilizando com a tag <figure> a figura tem uma quebra de espaço.

Outro método para inserir imagem é a imagem dinâmica, em que podemos utilizar a imagem para alterar a resolução, de acordo com o tamanho da tela.

<picture>

    <source media="(max-width: 750px)" srcset="Imagens/Clarencio\_maca.jpg" type="image/jpg">

    <source media="(max-width: 1050px)" srcset="Imagens/Clarencio\_sorrindo.jpg" type="image/jpg">

    <img src="Imagens/Clarencio\_serio.jpg" alt="Clarencio">

</picture>

A tag <Picture> é a área da imagem, “media” é qual imagem vai aparecer de acordo com a resolução da janela.

# Favicon

<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">

Favicon é para o emoticon na guia, ele é colocado na parte de head do programa.

Href: é a localização do emoticon no computador.

# Código

A área que será colocado o código, iniamos com a tag <pre>, depois inicia a <code>.

<pre>

<code>

    <código>

</code>

</pre>

# Links

A tag para atribuir links tanto para imagens quanto para frases é a <a>.

Exemplo:

    <a href="https://web.facebook.com/?\_rdc=1&\_rdr">Direcionar para o facebook</a>

Além disso, pode ser usado um comando chamado ‘target’, em que podemos selecionar se o link que será aberto é para a página anterior ou a página atual. Para atribuir para uma nova guia, devemos usar os atributos:

\_blank vai abrir o link em uma nova janela em branco

\_self vai abrir o link na janela ou frame atual (padrão)

\_top vai desfazer todos os frames e abrir o destino no navegador completo

\_parent similar ao uso do \_top em uma referência à janela mãe nome-do-frame caso esteja usando frames, indicar o nome da janela a abrir.

<h1> Link do Github</h1>

    <a href="https://github.com/Josuerodrigojr">Link do Github</a>

    <br>

    <a href="https://web.facebook.com/josue.rodrigo.9">Link do Facebook</a>

    <h2> Link dentro do site</h2>

    <a href="Exemplo 5.html" target="\_self">Exemplo 5</a>

    <h2>Link em uma nova aba</h2>

    <a href="https://github.com/Josuerodrigojr" target="\_blank">Link do Github</a>

    <h2>Usando o 'top'</h2>

    <a href="https://github.com/Josuerodrigojr" target="\_top">Link do Github</a>

    <h2>usando o parent</h2>

    <a href="https://github.com/Josuerodrigojr" target="\_parent">Link do Github</a>

Além disso, pode indicar o destino do link usando o *rel.*

‘next’ indica que o link é para a próxima parte do documento atual

‘Prev’ indica que o link é para a parte anterior do documento atual

‘Author’ indica que é um link para o site do autor do artigo atual

‘External’ indica que é um link para outro site que não faz parte do site

‘Nofollow’ indica que é um link para um site não endossado, como um link pago Exemplo:

<h2>Link dentro do site</h2>

<a href="Exemplo 7.html" target ="\_self" rel="next">Exemplo de listas descritivas</a>

<h2>Link github</h2>

<a href="https://github.com/Josuerodrigojr" target="\_blank" rel="external">Meu Github</a>

<h2>Link dentro do site</h2>

<a href="Exemplo 7.html" target ="\_self" rel="prev">Exemplo de listas descritivas</a>

<h2>Link github</h2>

<a href="https://github.com/Josuerodrigojr" target="\_blank" rel="author">Meu Github</a>

<h2>Link dentro do site</h2>

<a href="Exemplo 7.html" target ="\_self" rel="nofollow">Exemplo de listas descritivas</a>

Podemos também criar algo para ser baixado, utilizando o ‘download’ e descrevendo o arquivo que vai ser baixado em ‘type’. No download é o nome que vai ser baixado. Exemplo:

 <h2>Baixar arquivo pdf</h2>

    <a href="Arquivos/Curriculo - Josué.pdf" download="Meu\_curriculo.pdf" type = "application/pdf">Meu curriculo</a>

    <h2>Baixar imagem</h2>

    <a href="Arquivos/background\_zoom.png" download = "Fundo de tela da Lets.png" type ="application/png"> Imagem</a>

Exemplos de type:

application/zip

text/html

text/css

text/Javascript

video/mp4

video/H264

video/JPEG

audio/aac

audio/mpeg

font/ttf

image/jpeg

image/png

# Modificando a fonte, cor e tamanho

Um jeito de alterar tanto a fonte, a cor e o tamanho da palavra, está dentro da tag <font>. É recomendado o uso cauteloso dessa ferramenta, pois, pode sobrecarregar a página. Para o uso dessa tag, devemos informar as características, e logo após colocar o texto que vai ser alterado dessa forma.

Dentro da tag fonte, o atributo face é para o tipo de fonte que quero no texto, color é para indicar a cor que eu quero no texto e size o tamanho que quero do texto.

Exemplo:

    <font face="Arial" color = "#1C1C1C" size = "12">

        Texto com fonte arial, cor azul e tamanho 12

    </font>

</br>

    <font face = "Courier" color = 'red' size = '30'>

        Texto com fonte courier, cor vermelha e tamanho 30

    </font>

        <font face="Arial" color = "(131,111,255)" size = "12">

        Texto com fonte arial, cor azul e tamanho 12

    </font>

Tipos de fontes:

* Arial
* Brush Script MT
* Comic Sans
* Courier New
* Garamond
* Georgia
* Helvetica
* Impact
* Lucida Console
* Palatino
* Tahoma
* Times New Roman
* Trebuchet MS
* Verdana

Observação, na parte de cor, eu posso utilizar tanto o nome em inglês das cores, quanto o comando hexadecimal ou as cores na tabela RGB.

# Tabelas

Para criar a tabela, devemos utilizar três tags.

* <table></table> que indica a região que será a tabela. Na criação dessa tag, para a tabela não aparecer sem linhas ou colunas, deve usar o comando border.
* <tr></tr> indica a região que será as linhas.
* <td></td> indica as colunas e os conteúdos de cada coluna.

Exemplo:

    <p> Uma tabela simples</p>

    <table border="1">

        <tr>

            <td>A11</td>

            <td>A12</td>

        </tr>

        <tr>

            <td>A21</td>

            <td>A22</td>

        </tr>

    </table>

Dois atributos importantes na criação de tabelas são os atributos *rowspan* e *colspan.* O rowspan é para um elemento ocupar mais de uma linha, e o colspan é para um elemento ocupar mais de uma coluna.

Antes de definir a linha, podemos usar duas tags para a definição de tabela, <thead> e <tbody>, em <thead> utilizamos th para definir as colunas e em <tbody> utilizamos o <td> para definir as colunas:

    <table>

        <thead>

            <tr>

                <th>X</th>

                <th>Y</th>

                <th>Z</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td>X</td>

                <td>Y</td>

                <td>Z</td>

            </tr>

        </tbody>

    </table>

Exemplo:

<p>Tabela utilizando os atributos rowspan e colspan</p>

<table border="1">

    <tr>

        <td rowspan = "2">A11 e A21</td>

        <td>A12</td>

        <td>A13</td>

    </tr>

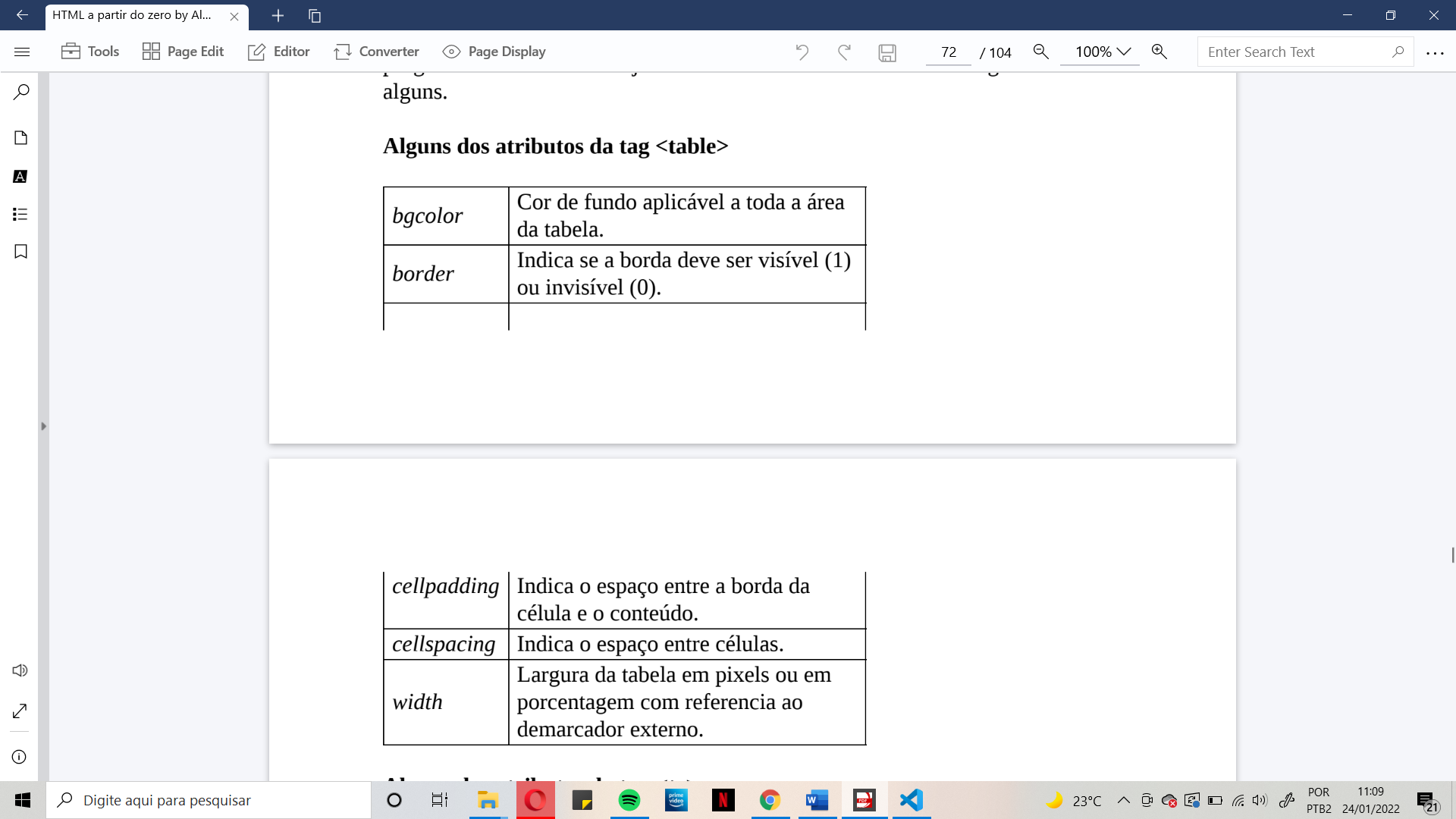
    <tr>

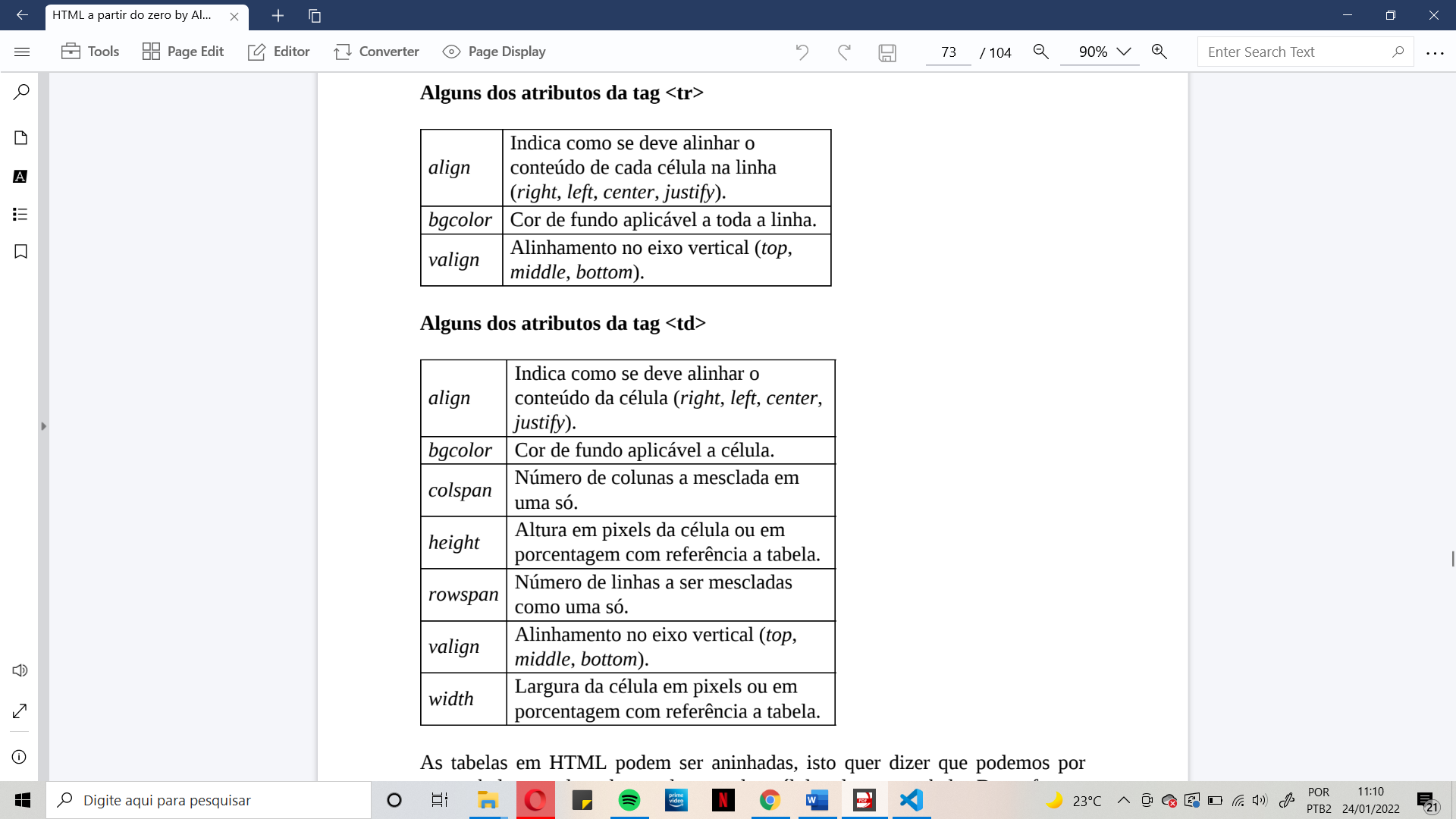
        <td colspan = "2">A22 e A23</td>

    </tr>

</table>

Abaixo temos alguns exemplos que pode ser utilizado na tag <table>, <tr> e <td>.





# Div e span

Para uma melhor organização, temos as duas tags, dev e span. A tag dev é utilizando como uma quebra de linha, enquanto a tag span é utilizado como uma quebra de espaço.

    <div> Utilizando o div</div>

    <div> Utilizando o div</div>

    <span>Utilizando o span</span>

    <span>Utilizando o span</span>

# Criando páginas com quadros

A definição de criar uma página com quadros, é o uso de mais de uma informação na tela. Para entender a definição, é utilizado três conceitos:

1. A tag <frameset> é a tela completa do computador. Nessa tag, posso utilizar o atributo rows ou cols, em que decido se quero definir meus quadros em linhas ou colunas. Além disso, devo indicar qual é a medição de cada um. Posso utilizar tanto valores com medições, ou asteriscos. Quando utilizo dois astericos, quero separar as telas na mesma dimensão. Um valor e um asterisco, significa que uma página deve ter aquela medicação e o restante é da segunda página.
2. A tag <frame> onde eu posso definir os quadros que irão aparecer na minha janela que é o frameset.
3. A tag <noframes>, é a mensagem que deve aparecer para o usuário, caso o navegador dele não suporte.
4. </head>
5. <frameset rows = '\*,\*'>
6. <frame src ="div, span.html">
7. <frame src = "Fontes, cores e tamanho.html">
8. <noframes>
9. <body>
10. <p>Não suporta</p>
11. </body>
12. </noframes>
13. </frameset>

Detalhe importante: A tag frameset e frame, fica antes do body e depois do head. Já o noframes, fica entre o body.

Além disso, podemos separar os frames em quadros também.

<frameset cols = '80,\*,50'>

    <frame src = "Listas.html">

        <frameset rows='\*,\*'>

            <frame src = 'Links.html'>

            <frame src = "Fontes, cores e tamanho.html">

        </frameset>

        <frame src = "Negrito, italico e sublinado.html">

<body>

</body>

</frameset>

# Formulário

## Lista suspensa

Para criar a lista suspensa no formulário, utilizamos 3 tags.

* A tag <form>, indica a área que vai ficar o formulário.
* A tag <select> é onde ficará o nome da lista suspensa que estamos criando.
* A tag <option> é as alternativas que o usuário pode selecionar.

É como se fosse a criação de uma tabela, o form seria a tabela, select seria a linha e a tag option seria as colunas.

Exemplo:

        <form name=’formulário’ >

        Indique sua profissão:

        <select name=’profissão’ >

        <option value=’0’ selected>Nenhuma</option>

        <option value=’1’>Programador</option>

        <option value=’2’>Especialista em redes</option>

        <option value=’3’>Designer gráfico</option>

        <option value=’4’>Adm. de banco de

        dados</option>

        </select>

        </form>

## Caixa de texto

A caixa de texto servirá para o usuário digitar algum texto em uma área designada. Devemos abrir com a tag <form>, logo depois utilizar a tag <textarea> informando o tamanho da caixa de texto que o website deverá ter.

    <form name='Caixa de Texto'>

        Endereço: </br>

        <textarea name="Teste" cols="50" rows="20">

        </textarea>

    </form>

## Pequena caixa de texto

Essa tag é utilizada para termos uma pequena caixa de texto ao lado de alguma informação que o usuário deve digitar. Diferente do exemplo anterior, não é necessário informar as características e o tamanho da caixa de texto. De forma semelhante, utilizando novamente a tag <form> para construir a área designada como formulário, nas áreas que quero esse pequeno retângulo, colocamos <input>.

    <form>

        Nome:

        <input type='text' name='nome'/>

    </form>

Quando utilizamos o type “password”, é a ocultação de caracteres, o usuário quando digitar, será ocultado por asteriscos ou pontos. Já quando se é utilizando o type hidden, temos a caixa de texto oculta. Podemos utilizar o type ‘radio’ que é colocado opções em que usuário pode selecionar um ou mais. Temos também o tipo ‘checkbox’, utilizando para criar opções que podem selecionar e desmarcar. Para termos um botão na página, usamos o type ‘botton’. Utilizamos o type ‘submit’, que é semelhante ao botão, a diferença é que a função dele seja somente enviar o formulário que está sendo preenchido. E o último tipo, que também é um botão é o type =’cancel’, que serve para cancelar o envio do formulário e preenche as lacunas novamente. (O cancelar não foi quando testei)

No exemplo abaixo, tem todos os tipos de types mencionados.

    <form>

        Nome:

        <input type='text' name='nome'/>

    </br>

    Senha:

    <input type ='password' name='senha'/>

</br>

    Invisivel:

    <input type ="hidden" name = 'Invisivel'/>

    <br>

    Sexo:

    <br>

    <input type = 'radio' name ='M'> Masculino <br>

    <input type="radio" name = 'F'> Feminino <br>

    Sexo:

    <br>

    <input type = 'checkbox' name ='M'> Masculino <br>

    <input type="checkbox" name = 'F'> Feminino <br>

    <input type ='button' value='Botão'>

    <br>

    <input type = 'cancel' name ='Cancelar'>

    </form>

    Invisivel:

    <input type ="hidden" name = 'Invisivel'/>

    </form>

## Abreviações

A tag para abreviação é <abbr>.

    <p>

        Biden é o presidente dos <abbr title="Estados Unidos da América">EUA</abbr>

    </p>

## Header, section and footer

Na construção de um web site, é importante ter organização, essas tags servem para que possamos organizar de forma que podemos montar o cabeçalho, o corpo e o rodapé.

    <header>

        Nota: Este artigo é sobre a série da Netflix. Para a série polonesa, veja Wiedźmin (série de televisão). Para o jogo eletrônico, veja The Witcher (série de jogos eletrônicos).

    </header>

    <section>

        The Witcher (em polonês, Wiedźmin) é uma série de televisão via streaming estadunidense-polonesa de drama e fantasia, criada por Lauren Schmidt Hissrich, baseada na série de livros do escritor polonês Andrzej Sapkowski. Situado em uma terra fictícia de inspiração medieval conhecida como "o continente", The Witcher explora a lenda de Geralt de Rivia e da Princesa Cirila, que estão ligados um ao outro pelo destino.[7] A série é estrelada por Henry Cavill, Freya Allan e Anya Chalotra.

    </section>

    <footer>

        A primeira temporada consistiu em oito episódios e foi lançada na Netflix em 20 de dezembro de 2019. Foi baseada em The Last Wish e Sword of Destiny, que são coleções de contos que precedem a saga principal de Wiedźmin. A segunda temporada, composta por oito episódios, foi lançada em 17 de dezembro de 2021.[9] Em setembro de 2021, a Netflix renovou a série para uma terceira temporada.[10] Um filme de animação, The Witcher: Nightmare of the Wolf, foi lançado em 23 de agosto de 2021,[11] enquanto uma minissérie prequela, The Witcher: Blood Origin, será lançada em 2022.[12]

    </footer>

# Vídeos/Áudios

Para inserir vídeos e áudios, temos que utilizar vários formatos do arquivo. Para a introdução do áudio, abrimos a tag áudio. Colocamos os comandos “autoplay controls loop” para que tenha controle no áudio. Colocamos mais de um tipo de áudio que é para o computador verificar se é compatível com o navegador. E no final, uma mensagem caso não seja compatível com nenhum dos tipos.

Já em vídeo, colocamos o tamanho do vídeo que queremos na janela, o poster que tem que aparecer antes de dar o play, o esquema restante é o mesmo.

Quando for colocar por algum link, por youtube por exemplo, temos a opção no site, na parte de compartilhamento.

<body>

    <h1>Reproduzindo Àudio</h1>

    <p>Vamos aprender a reproduzir áudios em HTML5</p>

    <audio preload ="metadata" autoplay controls loop>

        <source src="Arquivos/musica.mp3" type="audio/mp3">

        <source src="Arquivos/musica.ogg" type="audio/ogg">

        <source src="Arquivos/musica.wav" type="audio/wav">

        <p>Infelizmente seu navegador não consegue reproduzir o áudio</p>

        <a href="Arquivos/musica.mp3"> Clique aqui para baixar o arquivo MP3</a>

    </audio>

    <h1>Reproduzindo Videos</h1>

    <p>Vamos aprender a reproduzir vídeos em HTML5</p>

    <video width="560" poster="Arquivos/background\_zoom.png" autoplay controls loop>

        <source src="Arquivos/video.mp4" type="audio/mp4">

        <source src="Arquivos/musica.m4v" type="audio/m4v">

        <source src="Arquivos/musica.ogg" type="audio/ogg">

        <p>Infelizmente seu navegador não consegue reproduzir o vídeo</p>

        <a href="Arquivos/musica.mp3"> Clique aqui para baixar o arquivo MP3</a>

    </video>

    <h1>Inserindo vídeos do Youtube</h1>

    <iframe width="560" height="315" src="https://www.youtube.com/embed/bzvsfXIXV7o" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

</body>

# CSS

O CSS é tags que podemos utilizar para estilizar nossa página html. Existe duas maneiras de se fazer, utilizando a tag de css no arquivo html, ou, utilizando a tag css em um arquivo separado e orientando no arquivo do html.

## Alterar fundo e cor da letra

Utilizamos um comando css, a tag <style>, dentro da tag colocamos os atributos *backgroun-color* e *color*. Responsavel pelo fundo do texto e o texto.

Para que o programa entenda quais são os parágrafos que se devem ser feitas as alterações, é informando antes da abertura do parêntese.

    <style>

        em, strong, small {background-color:yellow;

        color: red;

    }

    </style>

Existe uma segunda forma de utilizar o css para modificar uma página, criando o arquivo css em um outro arquivo, e, referenciando no arquivo html com a tag <link>.

Arquivo css:

em, strong, small {

    background-color: chocolate; color: blueviolet;

}

Arquivo HTML:

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Testes</title>

<link rel="stylesheet" href="Css/Teste.css">

</head>

<body>

    <em>Italico</em>

    <br>

    <strong> Negrito</strong>

    <br>

    <small>Pequeno</small>

    <br>

    <u>Sublinhado</u>

</body>

</html>

No arquivo css, posso declarar que tipo de letra eu vou querer, além disso, posso selecionar o body, que seria a página completa a ser mostrada. Posso alternar a letra com a tag <font-family>. Podemos utilizar também a tag <font-style> para selecionar normal, italic (itálico) e oblique (oblíquia).

body{

    background-color: chocolate; color: blueviolet;

}

h1 {

    font-family: Arial; font-style: oblique;

}

h2

{font-family: 'Times New Roman';}

h3{

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

}

Mas, podemos aparecer com um problema mais complexo, queremos configuras alguns parágrafos, mas não todos, com isso, podemos utilizar o que chamamos de id. No arquivo CSS colocamos #exemplo e informamos como atributo nas tags que queremos as configurações propostas. Quando utilizamos esse método do id, não podemos repetir, ou seja, o id só pode aparecer em uma tag.

Arquivo CSS:

#cabecalho{

    color: white;

    text-align: justify;

    background-color: blueviolet;

}

Arquivo HTML:

    <p id="cabecalho">

        The Witcher (em polonês, Wiedźmin) é uma série de televisão via streaming estadunidense-polonesa de drama e fantasia, criada por Lauren Schmidt Hissrich, baseada na série de livros do escritor polonês Andrzej Sapkowski. Situado em uma terra fictícia de inspiração medieval conhecida como "o continente", The Witcher explora a lenda de Geralt de Rivia e da Princesa Cirila, que estão ligados um ao outro pelo destino.[7] A série é estrelada por Henry Cavill, Freya Allan e Anya Chalotra.

A primeira temporada consistiu em oito episódios e foi lançada na Netflix em 20 de dezembro de 2019. Foi baseada em The Last Wish e Sword of Destiny, que são coleções de contos que precedem a saga principal de Wiedźmin. A segunda temporada, composta por oito episódios, foi lançada em 17 de dezembro de 2021.[9] Em setembro de 2021, a Netflix renovou a série para uma terceira temporada.[10] Um filme de animação, The Witcher: Nightmare of the Wolf, foi lançado em 23 de agosto de 2021,[11] enquanto uma minissérie prequela, The Witcher: Blood Origin, será lançada em 2022.[12]

    </p>

Além do id, podemos utilizar o class, no arquivo css, usaríamos o . em vez do #, a diferença de ambos no uso de uma página, é que o class permite que eu possa associar mais de um atributo css nas tags, já o id, permite somente uma.

Arquivo CSS:

#cabecalho{

    color: white;

    text-align: justify;

    background-color: blueviolet;

}

.cab{

    color: chocolate;

}

.cab2{

    text-align: center;

}

Arquivo HTML:

    <h1> Usando o class</h1>

    <p class="cab cab2">

        The Witcher (em polonês, Wiedźmin) é uma série de televisão via streaming estadunidense-polonesa de drama e fantasia, criada por Lauren Schmidt Hissrich, baseada na série de livros do escritor polonês Andrzej Sapkowski. Situado em uma terra fictícia de inspiração medieval conhecida como "o continente", The Witcher explora a lenda de Geralt de Rivia e da Princesa Cirila, que estão ligados um ao outro pelo destino.[7] A série é estrelada por Henry Cavill, Freya Allan e Anya Chalotra.

A primeira temporada consistiu em oito episódios e foi lançada na Netflix em 20 de dezembro de 2019. Foi baseada em The Last Wish e Sword of Destiny, que são coleções de contos que precedem a saga principal de Wiedźmin. A segunda temporada, composta por oito episódios, foi lançada em 17 de dezembro de 2021.[9] Em setembro de 2021, a Netflix renovou a série para uma terceira temporada.[10] Um filme de animação, The Witcher: Nightmare of the Wolf, foi lançado em 23 de agosto de 2021,[11] enquanto uma minissérie prequela, The Witcher: Blood Origin, será lançada em 2022.[12]

    </p>

## Distanciamento de palavras/letras/linhas e alinhamento de texto

Através do arquivo css, podemos modificar o distanciamento que queremos no texto e seu alinhamento.

Para podermos alinhar, temos somente a função text-align, mas, podemos atribuir:

“Center” – Centralizar o texto.

“Left” – Colocar o texto na esquerda.

“Right” – Colocar o texto na direita.

“Justify” – Justificar o texto.

Já para termos a separação de palavras/distanciamento e/ou linhas, temos os comandos:

‘line-height’ – Tamanho da altura entre as linhas.

‘letter-spacing’ – Distância entre as letras.

‘word-spacing’ – Distância entre as palavras.

‘text-indent’ – Distância da margem da primeira linha.

h2{

    text-align: right;

}

h3{

    text-align: left;

}

h4{

    text-align: justify;

}

h5{

    text-align: center;

}

p{

    line-height: 15px;

    letter-spacing: 1px;

    word-spacing: 3px;

    text-indent: 40px;

}

## Imagem de fundo

Para colocar uma imagem de fundo, basta utilizar o comando background-image, e selecionar onde se encontra a imagem.

body{

    background-image: url(../boku.jpg);

}

## Bordas

Para adicionar as bordas, precisamos de 3 comandos.

‘border-color’ – Para definir a cor da borda.

‘border-style’ – O estilo da borda

‘border-width’ – O tamanho da borda.

h1, p{

    border-color:aqua;

    border-style: solid;

    border-width: 1px

}

## Espaçamento e margem

Em css, para podermos utilizar o espaçamento, é utilizado o comando *padding*. Podemos definir o *padding* pela esquerda, direita, topo ou baixo, mas, se usarmos somente o comando *padding*, é definido para as 4 direções.

Quando utilizamos o comando e somente um valor de medida, é utilizado nas 4 direções, se usarmos dois valores de medida, o primeiro valor é aplicado no topo e baixo, e o segundo valor é aplicado na esquerda e na direita. Se utilizarmos três valores, o primeiro valor é aplicado no topo, o segundo na direita e na esquerda e o último embaixo. Já se utilizarmos os 4 valores, é feito na direção horaria, o primeiro é topo, o segundo é direita, o terceiro é baixo e o último é esquerda.

Já a margem é o comando *margin*. Funciona de maneira semelhante a *padding*, mas, tem um comando especial, para verificar a disponibilidade da janela e colocar margens automáticas(Coloquei no exemplo abaixo).

body{

    padding: 10px 15px 20px 30px;

    margin: 0 auto;

}

## Dimensões

Podemos definir que cada paragrafo será armazenado em uma área e definir o tamanho e o fundo.

p{

    background-color: antiquewhite;

    height: 300px;

    width: 300px;

}

Cada parágrafo ficará em um quadrado antiquewhite.

## Posição: Estático, relativo e absoluto

No arquivo css, podemos utilizar o comando position, para determinar se será absoluto, relativo, static e fixed.

Utilizando o comando static, é posicionada automaticamente na página, relative é feita de maneira semelhante ao comando static, mas, podemos renderizar da maneira que queremos. Absoluto podemos configurar para o lugar que meu titulo vai pertencer e fixed, é utilizado para os elementos que mesmo abaixo a tela, o bloco ou objeto permanecerá ali.

Arquivo html:

<p class="quadrado">Titulo somente com a classe quadrado</p>

    <p class = "quadrado absoluto">Titulo com a classe quadrado e absoluto</p>

    <p class="quadrado relativo">Titulo com a classe quadrado e relativo</p>

    <p class="absoluto relativo">Titulo com as classes absoluto e relativo</p>

    <p class = "quadrado fixed absoluto"> Titulo com as classes absoluto, fixo e quadrado</p>

    <p class = "quadrado static">Titulo com as classes absoluto e estatico</p>

    <p>

Arquivo CSS:

.quadrado{

    background: green;

    height: 200px;

    width: 200px;

}

.absoluto{

    position: absolute;

    top: 20px;

    right: 30px;

}

.relativo{

    position: relative;

}

.fixed{

    position:fixed;

}

.static{

    position:static;

}

## Flex-container

O *flex container* serve para caso tenhamos uma lista, possamos orientar em como ela pode aparecer na janela web. Podemos alterar a posição por linha e colunas, alterando a ordem dos elementos dessa minha lista.

Para isso, utilizamos o atributo contêiner, como o pai, configurando para o flex ou block.

Logo depois, utilizamos o comando *flex-direction* informando a direção que a minha lista deve ter. O flex-direction, possui 4 principais atributos:

* Row: Deixa a lista na horizontal, em ordem crescente.
* Row-reverse: Deixa a lista na horizontal, em ordem decrescente.
* Column: Deixa a lista na vertical, em ordem crescente.
* Column-reverse: Deixa a lista na vertical, em ordem decrescente.

Podemos configurar cada tag da dev, utilizando o atributo filho item.

Arquivo CSS:

.conteiner{

    display: flex;

    max-width: 400px;

    margin: 0 auto;

    border: 1px solid black;

}

.row {

    flex-direction: row;

}

.row-reverse{

    flex-direction: row-reverse;

}

.column{

    flex-direction: column;

}

.column-reverse{

    flex-direction: column-reverse;

}

.item{

    flex: 1;

    margin: 5px;

    background-color: red;

    text-align: center;

    font-size: 1.5em;

}

h1{

    color:gold;

    font-size: 1.8em;

    font-family: 'Courier New', Courier, monospace;

    text-align: center;

}

Arquivo HTML:

<body>

    <h1>Utilizando conteiner row</h1>

    <div class="conteiner row">

        <div class="item">1</div>

        <div class="item">2</div>

        <div class="item">3</div>

    </div>

    <h1>Utilizando conteiner row-reverse</h1>

    <div class="conteiner row-reverse">

        <div class="item">1</div>

        <div class="item">2</div>

        <div class="item">3</div>

    </div>

    <h1>Utilizando conteiner column</h1>

    <div class="conteiner column">

        <div class="item">1</div>

        <div class="item">2</div>

        <div class="item">3</div>

    </div>

    <h1>Utilizando conteiner column-reverse</h1>

    <div class="conteiner column-reverse">

        <div class="item">1</div>

        <div class="item">2</div>

        <div class="item">3</div>

    </div>

</body>

## Flex-display

O flex display é para configurar a visualização dos produtos, podemos ordenar para que tenha uma quebra de linha, ou que todos os itens devam estar em uma linha só. Podemos usar os principais 3 tipos de itens:

* Wrap: Serve para que ocorra uma quebra de linha caso os itens ultrapassem a área delimitada;
* Wrap-reverse: Serve para que ocorra uma quebra de linha caso os itens ultrapassem a área delimitada, mas, a ordem é decrescente, de baixo para cima.
* Nowrap: Tenta encaixar todos os itens na mesma linha.

Arquivo HTML:

<body>

    <h1>Utilizando conteiner wrap</h1>

    <div class="wrap conteiner">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

        <div class="item">Item 6</div>

        <div class="item">Item 7</div>

        <div class="item">Item 8</div>

        <div class="item">Item 9</div>

    </div>

    <h1>Utilizando conteiner wrap-reverse</h1>

    <div class="conteiner wrap-reverse">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

        <div class="item">Item 6</div>

        <div class="item">Item 7</div>

        <div class="item">Item 8</div>

        <div class="item">Item 9</div>

    </div>

    <h1>Utilizando conteiner nowrap</h1>

    <div class="conteiner nowrap">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

        <div class="item">Item 6</div>

        <div class="item">Item 7</div>

        <div class="item">Item 8</div>

        <div class="item">Item 9</div>

    </div>

    <h1>Utilizando conteiner wrap</h1>

    <div class="wrap conteiner flex">

        <div class="item flex">Item 1</div>

        <div class="item flex">Item 2</div>

        <div class="item flex">Item 3</div>

        <div class="item flex">Item 4</div>

        <div class="item flex">Item 5</div>

        <div class="item flex">Item 6</div>

        <div class="item flex">Item 7</div>

        <div class="item flex">Item 8</div>

        <div class="item flex">Item 9</div>

    </div>

    <h1>Utilizando conteiner wrap-reverse</h1>

    <div class="conteiner wrap-reverse">

        <div class="item flex">Item 1</div>

        <div class="item flex">Item 2</div>

        <div class="item flex">Item 3</div>

        <div class="item flex">Item 4</div>

        <div class="item flex">Item 5</div>

        <div class="item flex">Item 6</div>

        <div class="item flex">Item 7</div>

        <div class="item flex">Item 8</div>

        <div class="item flex">Item 9</div>

    </div>

    <h1>Utilizando conteiner nowrap</h1>

    <div class="conteiner nowrap">

        <div class="item flex">Item 1</div>

        <div class="item flex">Item 2</div>

        <div class="item flex">Item 3</div>

        <div class="item flex">Item 4</div>

        <div class="item flex">Item 5</div>

        <div class="item flex">Item 6</div>

        <div class="item flex">Item 7</div>

        <div class="item flex">Item 8</div>

        <div class="item flex">Item 9</div>

    </div>

</body>

Arquivo CSS:

.conteiner{

    display: flex;

    max-width: 400px;

    margin: 0 auto;

    border: 1px solid black;

}

.wrap{

    flex-wrap: wrap;

}

.nowrap{

    flex-wrap: nowrap;

}

.wrap-reverse{

    flex-wrap: wrap-reverse;

}

.item{

    margin: 5px;

    background-color: #E1CD58;

    text-align: center;

    font-size: 1.5em;

    color: #5C5324;

}

h1{

    color:#5C4F08;

    font-size: 1.8em;

    font-family: 'Courier New', Courier, monospace;

    text-align: center;

}

.flex{

    flex: 1;

}

## Flex justify contente

Esse comando no css, serve para podermos alinhar de forma organizada nosso espaçamento dos elementos. Utilizamos o comando justify-content, e podemos atribuir as seguintes características para esse comando:

* flex-start (padrão): os ítens são alinhados junto à borda de início (start) de acordo com qual for a flex-direction do container.
* flex-end: os ítens são alinhados junto à borda final (end) de acordo com qual for a flex-direction do container.
* start: os ítens são alinhados junto à borda de início da direção do writing-mode (modo de escrita).
* end: os ítens são alinhados junto à borda final da direção do writing-mode (modo de escrita).
* left: os ítens são alinhados junto à borda esquerda do container, a não ser que isso não faça sentido com o flex-direction que estiver sendo utilizado. Nesse caso, se comporta como start.
* right: os ítens são alinhados junto à borda direita do container, a não ser que isso não faça sentido com o flex-direction que estiver sendo utilizado. Nesse caso, se comporta como start.
* center: os ítens são centralizados na linha.
* space-between: os ítens são distribuídos de forma igual ao longo da linha; o primeiro ítem junto à borda inicial da linha, o último junto à borda final da linha.
* space-around: os ítens são distribuídos na linha com o mesmo espaçamento entre eles. Note que, visualmente, o espaço pode não ser igual, uma vez que todos os ítens tem a mesma quantidade de espaço dos dois lados: o primeiro item vai ter somente uma unidade de espaço junto à borda do container, mas duas unidades de espaço entre ele e o próximo ítem, pois o próximo ítem também tem seu próprio espaçamento que está sendo aplicado.
* space-evenly: os ítens são distribuídos de forma que o espaçamento entre quaisquer dois itens da linha (incluindo entre os ítens e as bordas) seja igual.

O arquivo HTML:

    <h1>Utilizando flex-start</h1>

    <div class="start conteiner">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

    </div>

    <h1>Utilizando flex-end</h1>

    <div class="conteiner end">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

    </div>

    <h1>Utilizando center</h1>

    <div class="conteiner center">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

    </div>

    <h1>Utilizando space-around</h1>

    <div class="conteiner space-around">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

    </div>

    <h1>Utilizando space-between</h1>

    <div class="conteiner space-between">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

    </div>

    <h1>Utilizando space-evenly</h1>

    <div class="conteiner space-evenly">

        <div class="item">Item 1</div>

        <div class="item">Item 2</div>

        <div class="item">Item 3</div>

        <div class="item">Item 4</div>

        <div class="item">Item 5</div>

    </div>

O arquivo css:

.conteiner{

    display: flex;

    max-width: 400px;

    margin: 0 auto;

    border: 1px solid black;

}

.item{

    margin: 5px;

    background-color: #E1CD58;

    text-align: center;

    font-size: 1.5em;

    color: #5C5324;

}

h1{

    color:#5C4F08;

    font-size: 1.8em;

    font-family: 'Courier New', Courier, monospace;

    text-align: center;

}

.start{

    justify-content: flex-start;

}

.end{

    justify-content: flex-end;

}

.center{

    justify-content: center;

}

.space-between{

    justify-content: space-between

}

.space-around {

    justify-content: space-around;

}

.space-evenly{

    justify-content: space-evenly;

}

Podemos utilizar também o comando de colunas, misturando colunas com o justify-content.

## Grid

Outra forma de podermos dividir o layout de uma página Web, é utilizando o grid, diferente do *flex*, o grid serve para dividir a página, tanto na vertical, como na horizontal, utilizando o principio de coluna e linhas:

* grid-template-columns e grid-template-row: Para dividirmos a quantidade de colunas e linhas da página, para essa divisão, podemos usar o conceito de fração, de porcentagem, de pixel e etc. Utilizasse mais a fração e o da porcentagem.
* Grid-column-start e grid-column-end: Para selecionar a ocupação que a página vai ter na minha coluna, colocando o início e o fim. (Observação: podemos colocar o comando grid-column e separar por barra o inicio e o fim).
* Grid-row-start e grid-row-end: Para selecionar a ocupação que a página vai ter na minha linha, colocando o início e o fim. (Observação: podemos colocar o comando grid-row e separar por barra o início e o fim).
* Grid-column-grap: Esse comando é utilizado para delimitar o espaço para separar as colunas.
* Grid-row-grap: Esse comando é utilizado para delimitar o espaço para separar as linhas.
* (Observação, podemos utilizar o comando grid-gap, primeiro delimitando o espaço entre as colunas e separado por uma barra, o espaço entre as linhas.
* grid-auto-rows: minmax(400px, auto): Delimita que deve utilizar o tamanho da imagem para fazer uma linha.

Arquivo HTML:

 <div class="layout">

    <div class="noticia noticia-principal">

        <h1>Militar fica preso e deve responder por homicídio doloso por matar vizinho</h5>

        <ul>

            <li>Sargento da Marinha que atirou em vizinho no RJ foi candidato no MA</li>

        </ul>

    </div>

<div class="noticia">

        <img src="https://s2.glbimg.com/rJwXOVypws3BNgqGcM4b\_AOeAAc=/0x176:960x716/fit-in/540x304/middle/smart/filters:strip\_icc()/i.s3.glbimg.com/v1/AUTH\_bc8228b6673f488aa253bbcb03c80ec5/internal\_photos/bs/2022/m/X/vdh88ATVG7vVKwJxcfEA/whatsapp-image-2022-02-04-at-11.38.16.jpeg" />

        <h5>Fotos de prancheta e mais</h5>

        <h4>Vazamento de momentos privados do Ninho gera cobranças no Flamengo</h4>

        <ul>

            <li>Fla lista exemplos ao United e quer 'fatiar' compra de Andreas</li>

        </ul>

    </div>

    <div class="noticia">

        <img src="https://s2.glbimg.com/rJwXOVypws3BNgqGcM4b\_AOeAAc=/0x176:960x716/fit-in/540x304/middle/smart/filters:strip\_icc()/i.s3.glbimg.com/v1/AUTH\_bc8228b6673f488aa253bbcb03c80ec5/internal\_photos/bs/2022/m/X/vdh88ATVG7vVKwJxcfEA/whatsapp-image-2022-02-04-at-11.38.16.jpeg" />

        <h5>Fotos de prancheta e mais</h5>

        <h4>Vazamento de momentos privados do Ninho gera cobranças no Flamengo</h4>

        <ul>

            <li>Fla lista exemplos ao United e quer 'fatiar' compra de Andreas</li>

        </ul>

    </div>

    <div class="noticia">

        <img src="https://s2.glbimg.com/rJwXOVypws3BNgqGcM4b\_AOeAAc=/0x176:960x716/fit-in/540x304/middle/smart/filters:strip\_icc()/i.s3.glbimg.com/v1/AUTH\_bc8228b6673f488aa253bbcb03c80ec5/internal\_photos/bs/2022/m/X/vdh88ATVG7vVKwJxcfEA/whatsapp-image-2022-02-04-at-11.38.16.jpeg" />

        <h5>Fotos de prancheta e mais</h5>

        <h4>Vazamento de momentos privados do Ninho gera cobranças no Flamengo</h4>

        <ul>

            <li>Fla lista exemplos ao United e quer 'fatiar' compra de Andreas</li>

        </ul>

    </div>

    <div class="noticia">

        <img src="https://s2.glbimg.com/rJwXOVypws3BNgqGcM4b\_AOeAAc=/0x176:960x716/fit-in/540x304/middle/smart/filters:strip\_icc()/i.s3.glbimg.com/v1/AUTH\_bc8228b6673f488aa253bbcb03c80ec5/internal\_photos/bs/2022/m/X/vdh88ATVG7vVKwJxcfEA/whatsapp-image-2022-02-04-at-11.38.16.jpeg" />

        <h5>Fotos de prancheta e mais</h5>

        <h4>Vazamento de momentos privados do Ninho gera cobranças no Flamengo</h4>

        <ul>

            <li>Fla lista exemplos ao United e quer 'fatiar' compra de Andreas</li>

        </ul>

    </div>

</div>

Arquivo CSS:

.noticia img {

    width: 100%;

}

.noticia img{

    width: 100%;

}

.noticia-principal{

    grid-column-start: 1;

    grid-column-end: 3;

    /\*grid-column: 2/4\*/

    /\*grid-column: 2/span 2\*/

    grid-row-start: 2;

    grid-row-end: 1;

    /\*grid-row: 2/4\*/

    /\*grid-row: 2/span 2\*/

}

.layout{

    display: grid;

    grid-template-columns: 25% 25% 25% 25%;

    grid-template-rows: 50% 50%;

    grid-column-gap: 0.5em;

    grid-row-gap: 2em;

    /\*grid-gap: 0.5em 2em\*/

    /\* grid-auto-rows: 400px \*/

    /\* grid-auto-rows: minmax(400px, auto)\*/

}

Algumas observações:

@media (max-width="800px"){

    .grid{

        grid-template-columns: 30% 70%;

    }

}

Utilizamos o media, para definir a resolução do nosso template, depois de uma quantidade certa de pixels.

html, body{

    height: 100%;

}

body{

    margin:0%;

}

Colocamos a margin 0%, para que não tenha margens no site, e a altura 100%, para que aproveite a tela por completo.

main{

    background-color: #F7D44A;

    overflow-y: scroll;

    padding: 10px 20px;

}

Overflow-y, para indicar que temos uma rolagem na vertical, e padding para delimitarmos um espaço para a escrita.