Estrutura Básica HTML

Tag’s mais usadas/essências

<!DOCTYPEhtml> : Uma tag onde mostra que o site/pagina pode utilizar os recursos mais avançados e atualizados do HTML. (Uma tag especial, onde não a necessidade de seu fechamento)

<metacharset="UFT-8" /> : corrige a acentuação

<html>  </html> : Dentro da html será adicionado todos as informações do site.

<head>  </head> : (configurações do site) Não necessariamente oque irar aparecer no site.

<title> INFORMAÇÃO QUE DEVE APARECER NAS ABAS, LÁ EM CIMA </title>

<body> : Corpo do site onde todas as informações devem ser colocadas dentro dessa tag.

<p>  </p> : parágrafos, a cada linha colocar a tag <p> para que as linhas sejam embaixo uma das outras. (pois se não todas as informações inseridas fica na mesma linha.

<a href=”LINK”> : informação onde ao clicar deve ser direcionado </a>

<br>  <br> : pula uma linha

<button>  <button> : Cria um botão para ser clicado

Linha de Codigo:

<!DOCTYPE html>

<html>

     <head>

          <meta charset= "UFT-8"/>

          <title> título da página </title>

     </head>

     <body>

         <p>parágrafos, a cada linha colocar a tag <p>

     </body>

</html>

Entendendo CSS

Diagrama

Descrição gerada automaticamente

Css: Interno

Css: Inline

Css: Externo

Css Inline usamos da seguinte forma. Colocamos na linha de código usando style=’’informação’’ e pronto. (Colocamos dentro da tag) ex: <p style=’’inf’’>parágrafos, a cada linha colocar a tag <p>

Css interno: colocar as propriedades do css em dentro da aba head com a tag do css style Ex:

<!DOCTYPE html>

<html>

     <head>

           <style>

         p { color:blue; }

          <meta charset= "UFT-8"/>

          <title> título da página </title>

         <style/>

     </head>

     <body>

         <p>parágrafos, a cada linha colocar a tag <p>

     </body>  </html>

 Css externo: Criando outro arquivo na mesma pasta ex: XXXX.css

Para fazer que elas se comuniquem dentro da tag Head usamos a tag

<link rel="stylesheet" href="XXX.css" />

Colocamos dentro da Tag que desejamos mudar de cor o atributo: id ex:

<p id= “paragrafo”> parágrafos, a cada linha colocar a tag </p>

O id é exclusivamente apenas de um elemento, ou seja o id=“paragrafo”

É exclusivamente dessa linha ou seja, ao adicionar uma cor no css apenas o id parágrafo ficara dessa cor.

<!DOCTYPE html>

<html>

     <head>

          <link rel="stylesheet" href="XXX.css" />

          <meta charset= "UFT-8"/>

          <title> título da página </title>

     </head>

     <body>

         <p>parágrafos, a cada linha colocar a tag <p>

     </body>

</html>

Linha de Codigo CSS:

h1 {color:blue;

    font-size:15px;

}

button {color:aqua;

    font: size 60px;

}

Modelos de seleção:

1. id: atribuir algum id à uma tag específica. O id é único, pode ser atribuído a um único elemento.

No HTML

<tag id="titulo">texto qualquer</tag>

No CSS

#titulo {

    color:green;

}

2. class: é possível atribuir class a vários elementos, dando a mesma class a eles. É o modelo mais utilizado.

No HTML

<p class="texto">texto qualquer</p>

No CSS:

.texto {

    color:red;

}

3. Pelo próprio nome do elemento.

No HTML

<h1>texto qualquer</h1>

No CSS

h1 {

    color:red;

}

Como utilizar vários modelos de seleção de uma vez: basta separá-los por ","

#titulo, .texto {

    color:blue;

    font-size:25px;

}

Além disso, posso utilizar vários modelos de seleção, mas também especificar algo único a eles:

#titulo,

.texto {

    color:blue;

    font-size:25px;

}

#titulo {

     font-weight:normal;

}

.texto {

    font-weight:bold

}

Outro exemplo:

se eu utilizar um modelo de seleção pelo nome do elemento, por exemplo:

ol li {color:black};

será modificado o "li" apenas quando este for precedido por ele.

utilizando o "li" sozinho, ele será modificado, não importando quem o antecede, de é o "ul" ou "ol"

background-color: cor de fundo

color: cor da fonte

Existem 3 formas de definir cores no CSS:

1. Pelo próprio nome da cor em inglês (mais limitada, pois o HTML só contém 140 nomes de cores. É bom evitar utilizar a seleção de cor dessa forma)

Ex.

#titulo {

    background-color: red

}

2. Pela propriedade RGB (Vermelho-Verde-Azul). Nesse modelo é escolhido a proporção de cada cor primária, dando origem às cores queridas pelo usuário. O número de escolha varia de 0 a 255 - primeiro a proporção da cor vermelha, segundo a proporção da cor ver e por fim, a proporção da cor azul.

Ex.

#titulo {

    background-color:rgb(0,0,0); - fundo de cor preta

    color:rgb(255,0,0); - fonte de cor vermelha

}

Ex.

#titulo {

    background-color:rgb(255,255,255); - fundo cor branca

    color:rgb(0,0,0); - fonte de cor preta

    }

3. Método hexidacimal: Método mais utilizado e de melhor procedência

#titulo {

    background-color:#FF0000; -MÁXIMO DE VERMELHO E MÍNIMO DE VERDE E AZUL

}

#titulo {

   background-color:#FFFFFF; -MÁXIMO DE TODAS AS CORES, FORMA O FUNDO BRANCO

   color:#000000; -MÍNIMO DE TODAS AS CORES, FORMA O FUNDO PRETO

}

6 caracteres:

2 são vermelhos = 0123456789ABCDEF

2 são verdes = 0123456789ABCDEF

2 são azuis = 0123456789ABCDEF

Onde conseguir essas cores: em um software de layout

**Margins e Padding no CSS**

Todo elemento que por na tela, tera pelo menos 4 propriedades distintas:

-conteudo

-borda

-padding

-margins

mesmo que nao apareça, ela existe, so faltou ser delimitada, como por exemplo as bordas

Padding: é o espaçamento que fica entre a borda e o conteudo;

Margins: é o espaçamento externo a borda.

Padding/Margin especifico para os lados:

Padding/Margin-left:

Padding/Margin-rigth:

Padding/Margin-top:

Padding/Margin-bottom:

Padding/Margin especifico em um comando so:

Padding/Margin: 10px(cima) 20px(direita) 10px(baixo) 20px(esquerda) = nessa ordem, começando por cima, e segue sentido horário

Padding/Margin: 10px (cima/baixo) 20px(direita/esqueda)

Dicas sobre Width/Height

Essenciais

body {

margin: 0px;

}

\* {

box-sizing: border-box;

}

**Formatação de textos**

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8"/>

        <title>formatação de texto html</title>

        <link rel="stylesheet" href="formataçãotexto.css">

    </head>

    <body>

        1) <br> tag b = <b>NEGRITO</b> = sem semantica de importancia. <br>

        tag strong = <strong>NEGRITO</strong> = com sematica de importancia. <br><br><br><br>

        2) <br> tag i = <i>Italico</i> = sem semantica de importancia. <br>

        tag em = <em>talico</em> = com sematica de importancia. <br><br><br><br>

        3) <br> tag small = <small>torna texto ou palavra menor que o padrão</small>

        <br>

        tag del = <del>risca a palavra</del> <br>

        tag mark = <mark>marca a palavra</mark>

        <br>

        tag sup = <sup>sobe o texto</sup>

        <br>

        tag sub = <sub>desce o texto</sub>

        <br>

        <br><br><br>

    </body>

</html>

**TAG ESPECIAL**

<!--Informação-->

Essa tag é dada para a pessoa que for visualizar o codigo ter tal informação

**FORMATAÇÃO DE TEXTO**

Font-size: 30px;

Font-weight: bold; - Deixa todo o texto negrito

Font-family:helvetica; - Tipo da font do texto

Text-align:left; - Posição do texto

Text-decoration: none ; - Vai retirar as decoraçoes de texto

Text-decoration: line-through; - risca as palavras

Text-transform: uppercase; - todo o texto fica em caixa alta (maiusculo)

Text-transform: lowercase; - todo o texto fica em caixa alta (maiusculo)

Letter-spacing: 5px; - numero de espacamento entre letras

Letter-spacing: 5px; - numero de espacamento entre palavras

Line-height: 10px; - espacamento entre as linhas

Text-shadow: 2px 2px 2px black; - gera sombra atras das letras, o primeiro item a sombra vai para a direita,

O segundo a sombra vai para baixo, a terceira o nivel de embaçamento

a: houver {

Text-decoration: underline

}

**COMO IMPORTAR FONTES**

Lista ordenas são usadas a tag <ol> </ol>

<ul> </ul>

Usando tag <li> </li> dento de cada tag su perior no caso a ul e a ol

Para listas desorganizadas usamos o atribulo ul style=''list-style-type:XXXX'' para mudarmos os inets pequenos como ao invez de bolinhas serem quadrados bolinhas abertas etc...

Para o ol colocamos

Ol type=''X''

A para letras

I para algorismos romanos

1 para numeros etc... Caso loque um numero maio ex 4 a contagens começara a partir do 4 ao invez de type como atributo colocamos start

Para criar uma lista dentro de outra so add a lista dentro da que vc esta fazendo

No css fazemos o seguinte para aparecer cores ao passar o mouse em cima de determinado elemento

tbody tr:hover {

    background-color: rgb(6, 171, 0);

}

Ainda no css adicionamos para as escritas virem para a esquerda

thead {

    text-align:left ;

}

Aqui começamos os Formularios, base inical a base method significa o modo que a inf sera mandada, form indica que começamos o formulário action para direcionar para a pagina

 <form action="paginas/login.html" method="GET">

**INPUT**

Essa tag usamos dentro do formulário para abrirmos um campo de login para o usuário digitar a senha no lugar de password podemos inserir ‘’text’’ para o usuário poder escrever um texto.

<input type="password">

Essa é a inf para fazemos uma caixinha de marcar

<input type="checkbox">

Essa uma bolinha a ser marcada, sempre em mais de uma opção para ela funcionar corretamente, e com o mesmo nome para o grupo em si funcionar corretamente

    <input type="radio" name="op1">

<input type="radio" name="op1">

Submit é a tag para o botão de envio do formulário, em value colocamos o nome para aparecer no botão

<input type="submit"> value="enviar agora">

Abre um calendário de data , um lugar para colocar datas

<input type="date">

Assim como o date esse, a diferença é que a opção de por a data

<input type="datetime-local">

Assim como se auto explica, um espaço apenas destinado a números

<input type="number">

A opção de uma barra fixa e mover a tal tanto

<input type="range">

Uma variação to input text, porem mais arredondado

<input type="search">

Para urls (links)

<input type="url">

Semanas do ano e ano

<input type="week">

**SELECT**

Select é um ordenador de itens, ex: quando vc esta em um site de loja e quer filtar por mais vendidos mais baratos etc...

Um exemplo de como se montar

<select>

                <option>mais vendidos</option>

                <option>mais baratos</option>

</select>

Ex: Com name

<select name="ordenacao">

                <option value="best">mais vendidos</option>

                <option value="best2">mais baratos</option>

            </select>

Usando as opções value, na barra de tarefas irar aparecer as descrições do value, sem ele irar aparecer a descrição do option.

Ele basicamente envia a informação citada no value, porem para o usuário ira aparecer os do options

<select name="ordenacao">

                <option value="best">mais vendidos</option>

                <option value="best2" selected>mais baratos</option>

<select>

Usando selected na tag ela irar aparecer como primeira opção para o usuário

<select name="ordenacao">

                <option> </option>

                <option value="best2" selected  >mais baratos</option>

</select>

Não usando nada em option, a tag estará vazia

Taxtarea um campo grande para envio de formações geralmente colocado dentro da tag form

Ex:

 <textarea name="descricao"></textarea>

O label tem a função de indicar oque deve ser escrito na caixa de informação

O atributo for serve para ao clicar em cima da escrita seja aberta a caixa de informação

Ex:

<label for="1">AKI</label>

<input type="text" name="text" id="1">

Outro exemplo que adicionando o input dentro do label funcionara igual

 <label for="1"> AKI <br>

                <input type="text" name="text">

            </label>

Ex de como que fica:

Uma imagem contendo Retângulo

Descrição gerada automaticamente

**Tags Fieldset e Legend**

Fieldset gera uma linha, uma ‘’divisa’’ ente as forms selecionas

Legend gera um texto entre essa linha como algo para fazer a identificação de cada campo selecionado

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8"/>

    <title>Inicio</title>

    <link rel="stylesheet" href="estilo.css"/>

    <link href="https://fonts.googleapis.com/css2?family=Bungee+Spice&family=Noto+Sans+Mono:wght@100&family=Reem+Kufi+Ink&display=swap" rel="stylesheet">

<body>

    <form>

        <fieldset>

            <legend>Dados Pessoais</legend>

            <label>

            Primeiro nome: <br/>

            <input type="text" name="name"/>

             </label>

            <label>

            Sobre nome: <br/>

            <input type="text" name="lastname"/>

            </label>

        </fieldset>

    <fieldset>

        <legend>Dados Comerciais</legend>

        <label>

            nome da empresa: <br/>

            <input type="text" name="companyname"/>

        </label>

        <label>

            Cargo da empresa: <br/>

            <input type="text" name="companyjobit"/>

        </label>

    </fieldset>

        <input type="submit" value="cadastrar" />

    </form>

</body>

</html>

**Validações Nativas 1**

**Required**: O campo tem que ser obrigatoriamente preenchido ( Tem que estar dentro de um formulário)

<input type="text" name="name" required/>

**Minlength**: Significa O tanto de caracteres mínimas que tal campo deve ter

minlength="4"

**Maxlength**: Significa O tanto de caracteres maximas que tal campo deve ter

maxlength="7"

Obs: Para o max ou o min funcionar temos que ter o required tmb, pois sem ele é possível enviar o campo vazio.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<label>

                Idade: <br/>

                <input type="number" name="age"/>

                </label>

O **number** significa que apenas números serão colocados nakela área

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<label>

                Idade: <br/>

                <input type="number" name="age" step="2"/>

                </label>

**Step** significado que os números pularão de 2 em 2 ao clicar na flechinha para cima pulara de 4 para 6 de 6 para 8 e assim em diante

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 <label>

                Idade: <br/>

                <input type="number" name="age" min="18" max="100"/>

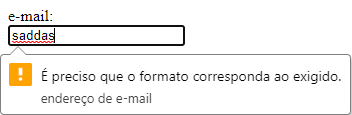
                </label>

Para os números e data o padrão muda para colocar um mínimo de caracteres será representado pela tag **min** e para caracteres máximas pela tag **max** (no lugar do number em type colocar date)

­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pattern** cria suas próprias padronizações sem colocando um **title** após ele para os usuários saberem qual o padrão que deve ser seguido

Usuário visualiza:



<label>

                    e-mail: <br/>

                    <input type="text" name="e-mail" pattern="(^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$)" title="endereço de e-mail"/>

                    </label>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Linha de Código:

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8"/>

    <title>Inicio</title>

    <link rel="stylesheet" href="estilo.css"/>

    <link href="https://fonts.googleapis.com/css2?family=Bungee+Spice&family=Noto+Sans+Mono:wght@100&family=Reem+Kufi+Ink&display=swap" rel="stylesheet">

<body>

    <form>

        <fieldset>

            <legend>Dados Pessoais</legend>

            <label>

            Primeiro nome: <br/>

            <input type="text" name="name" required minlength="4" maxlength="7"/>

             </label>

<label>

                Idade: <br/>

                <input type="number" name="age" step="2"/>

                </label>

            <label>

            Sobre nome: <br/>

            <input type="text" name="lastname"/>

            </label>

        </fieldset>

    <fieldset>

        <legend>Dados Comerciais</legend>

        <label>

            nome da empresa: <br/>

            <input type="text" name="companyname"/>

        </label>

        <label>

            Cargo da empresa: <br/>

            <input type="text" name="companyjobit"/>

        </label>

    </fieldset>

        <input type="submit" value="cadastrar" />

    </form>

</body>

</html>

***Input type Hidden***

***Input type file***

Sempre importante usar a propriedade enctype=multipart/form-data dentro do formulário

Ex código:

<form enctype="multipart/form-data">

                <label>

                    Foto: <br/>

                    <input type="file" name="pdf" accept="application/pdf" />

                </label>

    </form>

Para que podemos proporcionar que o usuário mande arquivos segue exemplo:

Em **type** colocamos **file** para representar o envio de um arquivo e em **name** para simbolizar qual arquivos queremos, agora para filtrar colocamos oque queremos/qual tipo de arquivo exemplo abaixo usamos **accept** para podermos selecionar qual tipo de arquivo queremos, exemplo abaixo queremos uma imagem assim só imagens poderão ser enviadas.

  <form>

                <label>

                    Foto: <br/>

                    <input type="file" name="photo" accept="image/\*"/>

                </label>

    </form>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Para outros arquivos exemplo:

<form>

                <label>

                    Foto: <br/>

                    <input type="file" name="pdf" accept="application/pdf" />

                </label>

    </form>

***Herança***

Algumas tags são herdadas outras não, as que são mais herdadas seriam as cores, fonts, tamanho de fonts etc...

Exemplos abaixo:

***Especificidade***

O CSS funciona de cima pra baixo, porem quando um ‘’argumento’’ e mais especificado ele vencera, se ambos tem o mesmo nível de especificação o ultimo vencera

***Unidades de medida absoluta***

***Unidades de medida relativas (fonte) 1***

***O inherit***

Inherit resumidamente significa forçar uma herença