**HTML – Linguagem de marcação de texto**

**<head> TAG DE ESTRUTURA do HTML informações que estamos passando (queremos passar) para o navegador, como <meta charset="UFT-8"> , <title>, <link>.**

**<body> TAG DE ESTRUTURA do HTML informações/conteúdo que queremos exibir na página, as tags de conteúdo ficam no body. Exemplo: <h1>, <p>, <img src="banner.jpg">, <title>, <link>, <header>.**

<h1> Título, vai do <h1> ao <h6>

<p> Parágrafo.

<strong> Negrito (destaque).

<em> Itálico (ênfase).

<!DOCTYPE html> **TAG DE ESTRUTURA do HTML** Para o navegador identificar que é um documento html e definimos qual versão do html estamos usando.

<html> **TAG DE ESTRUTURA do HTML** Marca todo HTML que vai ser renderizado, é uma TAG de conteúdo, logo precisa ser fechada com </>

<html lang="pt-br" > Fala qual é a linguagem que a página vai ser entendida, no caso português do Brasil.

<meta charset="UFT-8"> Dicionário(UFT-8) que tem as linguagens Unicode e ASIC2, logo é usado para que os acentos/caracteres funcionem corretamente. Possui todos os caracteres que são utilizados nas línguas da Europa, América Central, América do Sul, América do Note, etc.

<title> título da página

<!-- --> comentário em código HTML

<link rel="stylesheet" href="style.css"> faz a ligação do arquivo CSS e aponta (href)o local que o arquivo estar, é usado no <head>.

id="missao" = marcador de identificação, serve para identificar qual tag no HTML vai aplicar (referenciar) o CSS do arquivo externo CSS. Ex:

<p id="missao">

#missao {

    font-size:20px;

}

**Listas:**

*Não-Ordenadas <ul>:* não tem importância a ordem dos itens. E precisa marcar eles com a tag <li>

<ul>

            <li>Atendimento aos clientes</li>

            <li>Espaço diferenciado</li>

            <li>Localização</li>

            <li>Profissionais Qualificados</li>

        </ul>

*Ordenadas <ol>:* itens vem em ordem, a ordem tem importância como uma receita de bolo. E precisa marcar eles com a tag <li>

**CSS - Folha de Estilo em Cascata (propriedade estilo em cascata = o que faz no** **elemento pai reflete para o elemento filho)**

**CSS inline (primeira forma de apresentar o CSS)** na linha onde temos nossa tag, adicionamos a propriedade do CSS, dessa forma precisa repetir o código se for utilizar ele em mais linhas.

<p style="font-size: 20px"> muda tamanho da fonte.

<h1 style="text-align: center”>alinha o texto no centro.

<p style="font-size: 20px; text-align: center"> muda o tamanho da fonte e alinha no centro.

**CSS dentro do <head> (segunda forma de apresentamos o CSS)** é usado dentro do head e pode ser reutilizado o código, é aplicado na página inteira.

<style> </style> uma forma de fazer esse conteúdo ser repetível nesta página. É a segunda forma de apresentarmos o CSS. Exemplo: E está fazendo o parágrafo p ter o texto centralizado. Dentro dessa tag, podemos colocar marcações de CSS referentes aos elementos que temos no nosso HTML.

<style>

p {

text-align: center;

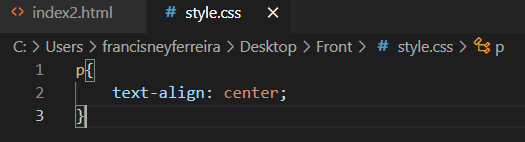
background: #CCCCCC;

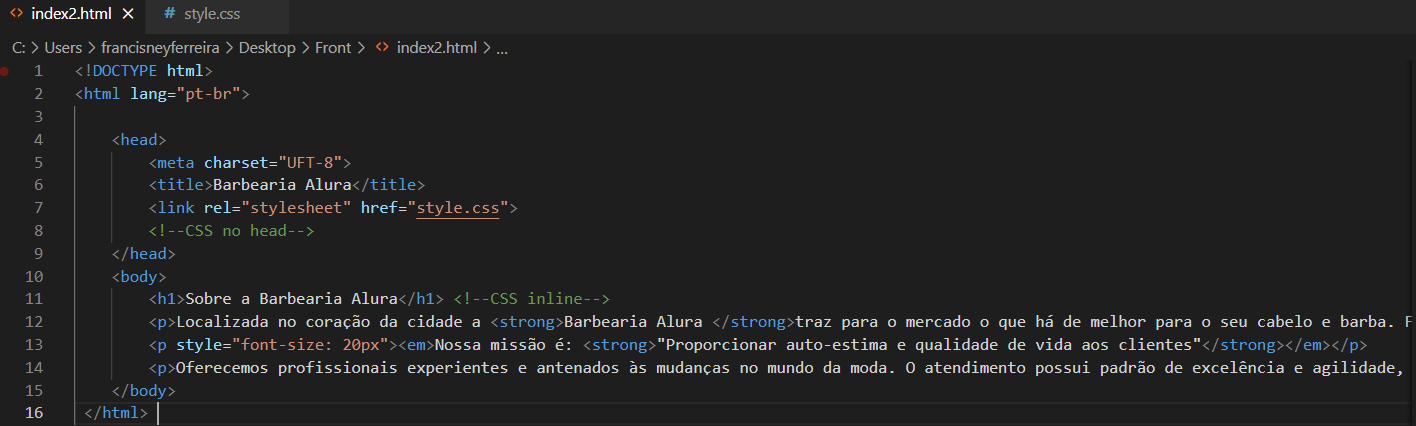
}

</style>

**CSS arquivo externo (terceira forma de apresentamos o CSS)** – cria um arquivo exteno que vai ter todo o CSS, e faz a ligação desse arquivo CSS com o arquivo html, é usado no <head>.

Essa é a forma mais comum, onde esse código pode ser aplicado em outras páginas fazendo-as a ter o mesmo estilo, bastando apenas fazer o link (referência) a esse arquivo externo.css:

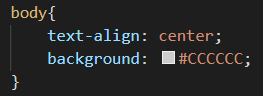




<link rel="stylesheet" href="style.css"> faz a ligação do arquivo CSS e aponta (href)o local que o arquivo estar, é usado no <head>.

<h1 style=" background: #CCCCCC”> muda a cor de fundo de um elemento (head, body) para cinza.

**Muda a cor de fundo do body para cinza.**



body {

text-align: center;

background: #CCCCCC;

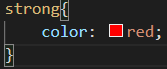
}

**Mudar a cor do texto para vermelho:**

strong{

color: red;

}

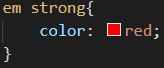


Mudar a cor de um parágrafo específico para vermelho, pode ser feito com CSS inline ou que mude o CSS (arquivo externo) para atender uma estrutura. Logo o navegador vai ler a estrutura e procurar uma estrutura específica. EX: em strong

em strong{

color: red;

}



**Funcionamento das cores no CSS:**

Hexadecimal: 0123456789ABCDEF

RGB = RED GREEN BLUE (três cores que derivam o restante das cores).

# \_ \_ \_ \_ \_ \_

#RRGGBB

0 = Ausência de cor

#000000 -> cor branca em hexadecimal

F= Máximo de cor

#FF0000 -> cor vermelha em hexadecimal

**Para representar as cores tem:**

Hexadecimal: #FF0000, #CCCCCC, #000000

Cores básicas: red, blue, yellow

RGB: Vai do 0 ao 255 para representar os níveis de cada cor. Sendo 0 ausência e 255 o máximo.

EX: Podemos representar a cor vermelha com o nome red, o hexadecimal #FF0000 e o RGB rgb(255,0,0).

**Representação de cor com RBG no CSS:**

rgb(255,255,255) -> branco

rgb(255,0,0) -> vermelho

rgb(0,0,255) -> azul

rgb(0,255,0) -> verde

**Marcador de identificação:**

id="missao" = marcador de identificação (#), serve para identificar qual tag no HTML vai aplicar (referenciar) o CSS do arquivo externo CSS. Ex:

<p id="missao">

#missao{

    font-size:20px;

}

- cria um identificador para marcar especificamente um elemento e faz referência a esse identificador no CSS.

**Adicionando imagem a página, a imagem é adiciona no body:**

<img src="banner.jpg">

<img src="banner.jpg">

Ou

<img src="beneficios.jpg">

Ou

<img src="beneficios.jpg" class="imagemBeneficios">

**Dimensões no CSS:**

Largura, altura, espaçamento interno de um elemento, espaçamento externo de um elemento.

**Navegador:**

F12 – Inspecionar: é uma ferramenta de DevTools de teste e inspeção que tem um layout mobile, mostrar os arquivos e deixa mexer no console, faz a separação do HTML, CSS, etc.

**Alterando altura de um elemento no CSS:**

height: 100 pix;

**Alterando largura de um elemento no CSS:**

width: 100%;

#banner{

    width: 100%;

}

**Alterando espaçamento de um elemento no CSS:**

**Espaçamento interno (padding): pode ser para todos lados ou só para uma direção. Esse espaçamento interno faz o elemento se comportar melhor naquele espaço que ele já têm. Logo serve para criar um respiro (espaço em branco) maior da borda para fora.**

padding: 20px; cria um espaçamento interno para todos lados.

padding-top: 20px; cria um espaçamento interno para cima, cria um quadro dentro para dá um respiro somente para cima.

*Qual o comportamento do elemento quando adiciono espaçamento interno, o padding?*

*O elemento aumenta o seu tamanho. Quando declaro que existe um espaçamento interno, estou aumentando o conteúdo.*

**Espaçamento externo (margin): pode ser para todos lados ou só para uma direção. Esse espaçamento interno faz o elemento se comportar melhor no espaço que já têm. Com isso a borda deixa de ser colada da lateral, se descola da lateral. Cria um espaço para fora do elemento, entre o elemento e a margem do navegador.**

Margin: 40px; cria um espaçamento externo para todos lados.

margin-right: 40px; cria um espaçamento externo para o lado direito.

margin-left: 40px; cria um espaçamento externo para o lado esquerdo.

**Criando borda em um elemento no CSS:**

border: 10px solid #000000;

Mexendo na borda consegue entender a largura e altura do elemento, a borda é a finalização do elemento.

**CRIANDO CLASSES:**

**Conceitos de Classes no CSS servem para marcar itens, e posteriormente colocar um estilo em cada um deles, e são repetíveis, logo pode marcar todos os itens com a mesma classe:**

<ul>

            <li class="itens">Atendimento aos clientes</li>

            <li class="itens">Espaço diferenciado</li>

            <li class="itens">Localização</li>

            <li class="itens">Profissionais Qualificados</li>

        </ul>

.itens{

    font-style: italic;

}

font-style: italic; todos os itens da lista que possuem a classe “itens” ficaram em itálicos.

**Criar Divisão na página – Separação entre as seções (blocos de conteúdo)**

Exemplo:

1 – Todo o texto de apresentação do site

2 – Benefícios apresentados para o cliente

<div> </div> Cria uma divisória/divisão na página, logo divide o conteúdo na página HTML em blocos. Por padrão as divisões não interferem na apresentação visual do conteúdo. Ela serve para marcar os conteúdos e a partir do CSS fazer os efeitos que quisemos.

<div class="beneficios">

            <h2>Benefícios</h2>

            <ul>

                <li class="itens">Atendimento aos clientes</li>

                <li class="itens">Espaço diferenciado</li>

                <li class="itens">Localização</li>

                <li class="itens">Profissionais Qualificados</li>

            </ul>

        </div>

Mudando a cor da classe benefícios da div no CSS:

.beneficios{

    background: #FFFFFF;

}

**Diferença identificador e classe:**

Identificador id (#id) – serve para itens únicos

Classe (.classe) – serve para itens que vão ser repetidos.

**Altera o tamanho de uma imagem no CSS:**

HTML:

<img src="beneficios.jpg" class="imagemBeneficios">

CSS:

.imagemBeneficios{

    width: 50%;

}

**Block e Inline:**

**BLOCK:** Quando uma TAG ocupa a largura inteira de uma página esse comportamento é chamado de block, ele bloqueia o conteúdo daquela linha. Exemplo: listas ordenadas e não ordenadas.

Logo o block sempre vai ocupa aquela linha mesmo que diminua o tamanho, largura do block, mesmo ele ocupando só metade. O conteúdo do block ocupa todo o tamanho da tela. Exemplo: imagem ocupa toda a tela.

**INLINE:** Deixa que tenha outros conteúdos na sua lateral (um do lado do outro). Logo o conteúdo da TAG é do tamanho do conteúdo que está ali (Ex: imagem tem o tamanho da imagem). Exemplo: imagens <img src=” “>

**Diferença entre block e inline:**

Mesmo que diminua o tamanho, ou seja, a largura do elemento block ele vai sempre ocupar aquela linha mesmo ele preenchendo só metade. O elemento inline por sua vez não deixa alterar o espaçamento interno e externo dele.

**Quando um elemento tem as duas condições, ele é inline e block:**

Ele bloqueia uma largura, mas essa largura é fixa, sou eu quem dou o tamanho da largura, e ele me permite alterar sua largura, seu espaçamento interno, seu espaçamento externo.

**Como se comporta um elemento que tem o seu display com o valor inline-block?**

O tamanho do seu conteúdo, mas regulável. Ou seja, o tamanho pode ser ajustado, tanto na largura, quanto na altura.

**Inline e block são características do display.**

display: inline; //um do lado do outro;

display: block;

display: inline-block; - ocupa só o tamanho do seu conteúdo, mas deixa mexer na largura e espaçamento.

ul{

    display: inline-block;

}

vertical-align: top; - conteúdo fica alinhado na vertical e no topo.

ul{

    display: inline-block;

    vertical-align: top;

    width: 20%;

    margin-right: 15%;

}

**Cabeçalho em HTML5:**

<header><\header> É o cabeçalho da página fica dentro do conteúdo/body, logo precisa ter mais destaque por isso usa o <h1>.

<header>

<h1>Barbearia Alura</h1>

</header>

No HTML5 dá para usar a TAG <div> para fazer o cabeçalho, contudo não é semanticamente correto, se for fazer um código correto deve-se a TAG correta, no caso o <header>.

**Fazer/criar um link na página HTML:**

<a href=> </a> pega você desse ponto e te leva para outra página. Serve para fazer um link que vai te levar de uma página para outra. Serve justamente como uma âncora, um link, que vai nos levar para outro ponto na internet

<li><a href="indexBarbearia.html">Home</a></li>

**Criando uma lista:**

<body>

        <header>

            <h1><img src="logo.png"></h1>

            <ul>

               <li><a href="indexBarbearia.html">Home</a></li>

               <li><a href="produtos.html">Produtos</a></li>

               <li><a href="http://www.google.com">Contato</a></li>

            </ul>

        </header>

    </body>

A lista pode ser ordenada: <ol></ol>

A lista pode ser desordenada: <ul></ul>

A lista possui itens que são marcados por: <li></li>

**Navegação (TAG menu de navegação)**

<nav> </nav> TAG de navegação do menu, pode ser usada com listas.

HTML:

<nav>

                <ul>

                    <li><a href="indexBarbearia.html">Home</a></li>

                    <li><a href="produtos.html">Produtos</a></li>

                    <li><a href="contato.html">Contato</a></li>

                </ul>

            </nav>

CSS:

nav li{

    display: inline;

}

**Deixar/transformar visualmente o texto/letras em maiúsculo:**

text-transform: uppercase; transforma o texto em maiúsculo.

text-transform: uppercase;

**Deixar a fonte mais forte/grossa (NEGRITO) na exibição:**

font-weight: bold;

font-weight: bold;

**Alterar tamanho da fonte:**

font-size: 22px;

font-size: 22px;

**Deixa o texto sem decoração nenhuma (tira sublinhado do texto):**

text-decoration: none; //remove a decoração do texto (sem sublinhado)

text-decoration: none;

**Aumentar espaçamento externo dos itens do link para a esquerda (1 top, 2 right, 3 bottom, 4 left):**

margin: 0 0 0 15px;

nav li{

    display: inline;

    margin: 0 0 0 15px;

}

**Mudar posição da barra de navegação/ ou outros elementos na página:**

position: static; a posição do elemento fica estática, está parado (padrão)

position: relative; a posição do elemento é relativa, pode ser alterada. Quando o elemento está deslocado, mas o ponto inicial dele continua o mesmo onde deveria estar.

position: absolute; posicionamento absoluto em relação a outra coisa, por exemplo a página toda, em relação ao cabeçalho, etc. Logo quando está com o position: absolute; o elemento automaticamente sai/desloca do cabeçalho. Portanto com o position absolute conseguimos posicionar o elemento onde quisemos na página, ele serve para mudar a posição do ponto inicial que ele (elemento) se encontra. O novo ponto inicial do elemento é o que selecionamos no navegador, contudo esse posicionamento absoluto é em relação a página.

top: 10px; cria um espaço do ponto onde ele deveria está para onde ele realmente está, logo ele fica 10px abaixo.

left: 10px; começa andar para direita pois está criando um espaço do ponto inicial e ele.

.caixa{

    position: relative;

    width: 940px;

}

nav{

    position: absolute;

    top: 120px;

    right: 0;

}

Menu = absoluto;

Caixa (div) = relatixa;

**Margem automática (espaçamento externo) da caixa/div (direita e esquerda) ou outro elemento:**

*Melhor técnica para alinhar uma div ao centro = usar o cálculo da margem automático.*

margin: 0 auto;

0 = para cima e para baixo. Auto = esquerda e direita se calculam automaticamente.

width: 940px; largura definida de forma estática.

.caixa{

    position: relative;

    width: 940px;

    margin: 0 auto;

}

**Padding espaçamento interno para cima e baixo no Header (cabeçalho aumenta um pouco):**

padding: 20px 0;

20 = para cima e para baixo. 0 = esquerda e direita.

Usou padding para que tudo dentro do cabeçalho esteja nesses limites e que nada dentro do cabeçalho chegue na margem. Dessa forma entre a margem e o conteúdo tem-se 20px para cima e para baixo.

header{

    background: #BBBBBB;

    padding: 20px 0;

}

**Divisão semântica na TAG <body>, onde fica o conteúdo principal**

<main></main> a tag main é onde fica o conteúdo principal, a coisa mais importante. Fica dentro do <body>

**Criando listas complexas (título, imagem, parágrafo)**

<ul class="produtos">

                <li>

                    <h2>Cabelo</h2>

                    <img src="cabelo.jpg">

                    <p class="produto-descricao">Na tesoura ou máquina, como o cliente preferir</p>

                    <p class="produto-preco">R$ 25,00</p>

                </li>

                <li>

                    <h2>Barba</h2>

                    <img src="barba.jpg">

                    <p class="produto-descricao">Corte e desenho profissional de barba</p>

                    <p class="produto-preco">R$ 18,00</p>

                </li>

                <li>

                    <h2>Cabelo + Barba</h2>

                    <img src="cabelo+barba.jpg">

                    <p class="produto-descricao">Pacote completo de cabelo e barba</p>

                    <p class="produto-preco">R$ 35,00</p>

                </li>

            </ul>

**Aplicando Bordas:**

A borda tem 3 aspectos fundamentais: tamanho, tipo/estilo, cor.

border-color: #000000;

    border-width: 2px;

    border-style: solid;

*Pode declarar também assim de forma mais simples (borda, tipo, cor):*

border: 2px solid #000000;

Tipo/estilo:

* solid linha \_\_\_\_\_\_
* dashed tracejado --------
* dotted pontilhado ........

Na estrutura de "caixas" do CSS, a borda fica posicionada entre o padding e a margin.

**Alterar ângulos da borda (arredondar o canto das bordas) / cantos arredondados:**

border-radius: 10 px; usado para arredondar as bordas, ou seja, arredonda todos os cantos da borda. Se quiser pode modificar o tamanho dos elementos da borda. Logo podemos utilizar para desenhar o canto arredondado dos elementos.

 border-radius: 10px;

border-radius-top-left o campo superior esquerdo.

border-radius-bottom-right campo inferior direito.

OBS: Se quiser pode modificar o tamanho dos elementos da borda.

 border-radius: 10px 20px 30px 40px;

**LISTA COMPLEXA NA PÁGINA FINALIZADA**

HTML

<ul class="produtos">

                <li>

                    <h2>Cabelo</h2>

                    <img src="cabelo.jpg">

                    <p class="produto-descricao">Na tesoura ou máquina, como o cliente preferir</p>

                    <p class="produto-preco">R$ 25,00</p>

                </li>

                <li>

                    <h2>Barba</h2>

                    <img src="barba.jpg">

                    <p class="produto-descricao">Corte e desenho profissional de barba</p>

                    <p class="produto-preco">R$ 18,00</p>

                </li>

                <li>

                    <h2>Cabelo + Barba</h2>

                    <img src="cabelo+barba.jpg">

                    <p class="produto-descricao">Pacote completo de cabelo e barba</p>

                    <p class="produto-preco">R$ 35,00</p>

                </li>

            </ul>

CSS

.produtos{

    width: 940px;

    margin: 0 auto;

    padding: 50px 0;

}

.produtos li{

    display: inline-block;

    text-align: center;

    width: 30%;

    vertical-align: top;

    margin: 0 1.5%;

    padding: 30px 20px;

    box-sizing: border-box;

    border: 2px solid #000000;

    border-radius: 10px;

}

.produtos h2{

    font-size: 30px;

    font-weight: bold;

}

.produto-descricao{

    font-size: 18px;

}

.produto-preco{

    font-size: 22px;

    font-weight: bold;

    margin-top: 10px;

}

**Fazer o CSS e o navegador ao invés de somar o percentual mais o pixels, mudar a forma como ele pensa:**

box-sizing: border-box;

Logo o box-sizing possibilita a modificação da forma que o tamanho do elemento é calculado no CSS. Ou seja, os valores de border e padding sejam incorporados às dimensões originais do elemento, mantendo-o dentro dos limites informados em width e height.

**Mouse passa o mouse cima do elemento (exemplo: menu) aplica um efeito/estilo -> Muda o comportamento do elemento:**

*hover* -> comportamento quando o mouse tiver por cima.

*nav a:hover* -> vai mapear o comportamento do link (a) que estão no menu de navegação. Logo quando mouse estiver por cima dos links dentro do menu de navegação ele vai ter uma ação/comportamento.

nav a:hover {

    color:blue;

    text-decoration: underline;

}

Nesse exemplo o mouse quando passa por cima dos links do menu, os links ficam azuis e sublinhados.

.produtos li:hover {

    border-color:blue;

}

**Texto sublinhado no CSS:**

text-decoration: underline;

**Quando o mouse está com o click pressionado muda o comportamento do elemento:**

.produtos li:active {

    border-color: green;

}

Nesse exemplo quando o click do mouse está pressionado as bordas dos itens de produto ficam verdes.

**Como alterar o tamanho de um elemento quando passar o mouse por cima, sendo que esse elemento está dentro de outro elemento (lista):**

.produtos li:hover h2 {

    font-size: 35px;

}

Nesse exemplo só alterar o tamanho do título quando passa o mouse por cima. Logo só altera o h2 (título) que está na lista (li) na classe .produtos

O comportamento serve para o elemento e todos os elementos que estão dentro dele. Ou seja, que são filhos do elemento. Logo o comportamento do CSS pode ser alterado com a interação do usuário.

*Quando podemos utilizar o comportamento do hover em um elemento?*

*No hover, mapeamos o comportamento do mouse por cima do elemento, e podemos usar isso para destacar visualmente o elemento em questão.*

***Pseudo-classes CSS***

* *hover, quando o usuário passa o cursor sobre o elemento*
* *active, quando um elemento está sendo ativado pelo usuário*

**Footer – Rodapé do HTML**

<footer>

            <img src="logo-branco.png">

            <p>Copyright Barbearia Alura - 2021</p>

        </footer>

Coloca a imagem de logo no rodapé e o texto.

footer {

    text-align: center;

    background: #CCCCCC;

}

Alinha o texto no CSS e coloca uma cor de fundo.

**Footer – Coloca/cria uma imagem no background - CSS:**

background: url("bg.jpg");

Coloca a url da imagem que queremos colocar. Ou seja, aponta para nossa imagem bg.jpg no pc.

footer {

    text-align: center;

    background: url("bg.jpg");

}

Alinha o texto e coloca/cria a imagem de fundo na página.

Quando se cria uma imagem de fundo no CSS, por padrão o CSS replica (copia e cola) essa imagem infinita vezes até ocupa todo o espaço do elemento. Usa a imagem pequena pois tem um tamanho pois o CSS vai replicando e com isso economiza banda de internet do usuário.

*Como adicionamos uma imagem de fundo em um elemento?*

*Com o valor url no background do elemento. No CSS, ao referenciarmos o endereço de uma imagem, usamos o valor url.*

*O CSS por padrão, copia e cola a imagem diversas vezes até ocupar todo o espaço do elemento.*

**Footer - Alterar espaçamento interno – CSS**

footer {

    text-align: center;

    background: url("bg.jpg");

    padding: 40px 0;

}

padding: 40px 0;

**Footer – Espaço entre os itens (logo e texto) - CSS:**

.copyright {

    color: #FFFFFF;

    font-size: 13px;

    margin: 20px 0 0;

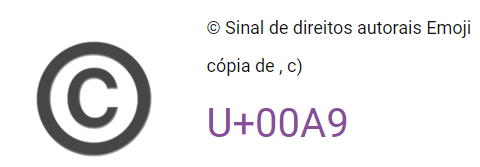
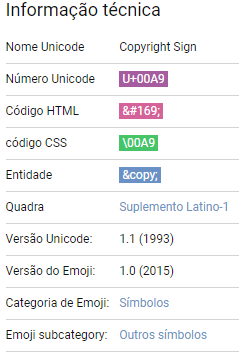
}

Cor do texto branco, fonte 13px, margem/espaço entre o texto e logo que está acima do texto.

**Footer - Usar caractere especial (UNICODE) para copyright – HTML**

Site para ver caracteres especiais (Unicode), possui tabela: https://unicode-table.com/pt/

https://unicode-table.com/en/

Usa a forma mais padrão que é a entidade:

Entidade: &copy;

<p class="copyright">&copy; Copyright Barbearia Alura - 2021</p>

<footer >

            <img src="logo-branco.png">

            <p class="copyright">&copy; Copyright Barbearia Alura - 2021</p>

        </footer>

**Formulários:**

O formulário é o responsável por pegar um conteúdo enviar para outra página ou lugar com esse conteúdo passado. Ou seja, ele passa o conteúdo para próxima página e esse conteúdo é tratado nessa próxima página. Pode também passar esse conteúdo para um servidor que vai enviar um e-mail, ou passar para um servidor que vai fazer o login. Logo sempre que estamos enviando informações que o usuário digitou de uma página para outra estamos usando formulário.

*Para decidir quando devemos usar um formulário nas nossas páginas, alguns motivos podem ser analisados:*

*1) Quando queremos enviar os dados para um servidor.*

*2) Quando queremos capturar informações que o usuário digitar.*

*3) Quando queremos enviar os dados para uma outra página.*

***TAG formulário, entrada com dados (input):***

<form></form>

 <input type="text">

Quando o usuário entra com algum dado ele está fazendo um input, toda tag de input tem um tipo. Logo usuário vai digitar um texto em um campo.

- tag <input>, para a entrada de dados do usuário.

- tag <label> etiqueta para o input.

- Conectar input com label: Coloca um id para o input e associamos esse id ao atributo for do label.

- label possui o display inline e o input possui display inline-block

Dentro de um formulário usa uma etiqueta para um input:

<label>Nome e sobrenome</label>

<input type="text">

label e input sempre andam junto (par de informação). Para cada label tem um input. O input tem um identificador, pois ele geralmente é único. A tag <label> é uma etiqueta para a entrada de dados, para o <input>.

<label for="nomesobrenome">Nome e sobrenome</label>

<input type="text" id="nomesobrenome">

E no label diz que aquela etiqueta é para um nome, logo elas são um par que se relacionam. Logo esse par (label e input) anda junto, e isso implica que quando clica no nome da label no navegador ela já vai para o input automaticamente, dessa forma facilitando a vida do usuário.

<main>

        <form>

                <label for="nomesobrenome">Nome e sobrenome</label>

                <input type="text" id="nomesobrenome">

        </form>

</main>

**Input de envio (cria um botão de envio):**

Envia os dados para o servidor, ou e-mail, ou para um próxima página.

<input type="submit">

<input type="submit" value="Enviar formulário">

Botão com o nome atribuído = Enviar formulário

**Formulário montado:**

<main>

        <form>

                <label for="nomesobrenome">Nome e sobrenome</label>

                <input type="text" id="nomesobrenome">

                <label for="email">E-mail</label>

                <input type="text" id="email">

                <label for="telefone">Telefone</label>

                <input type="text" id="telefone">

                <input type="submit" value="Enviar formulário">

        </form>

</main>



**Centralizando os dados da MAIN (formulário):**

main {

    width: 940px;

    margin: 0 auto;

}

**Colocando margem no formulário:**

form {

    margin: 40px 0;

}

Por padrão elementos do tipo parágrafo <p> são display do tipo block (ocupam 100% da largura (width da página). E o elemento inline ocupa só o tamanho do seu conteúdo.

**Quebrando estrutura do formulário (quebra de linha dos elementos) = linha do formulário fica um debaixo do outro.**

form label {

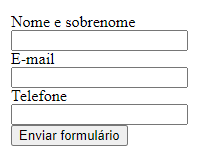
    display: block;

}

form input {

    display: block;

}



**Formulário – Aumentando tamanho da fonte e colocando margem:**

form label {

    display: block;

    font-size: 20px;

    margin: 0 0 10px;

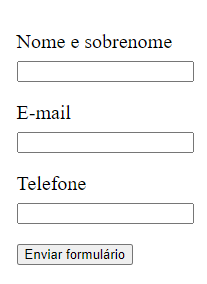
}

form input {

    display: block;

    margin: 0 0 20px;

}



**Formulário – Aumentando o tamanho da caixa de input, espaço interno:**

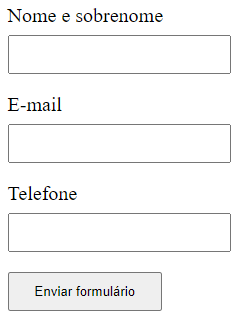
form input {

    display: block;

    margin: 0 0 20px;

    padding: 10px 25px;

}



*Qual propriedade devo alterar para que os meus <input> tenham um espaço entre a borda e o conteúdo?*

*A propriedade padding serve para alterarmos o espaçamento interno, entre o conteúdo e a borda.*

**Formulário – Aumentando a largura da caixa de input:**

form input {

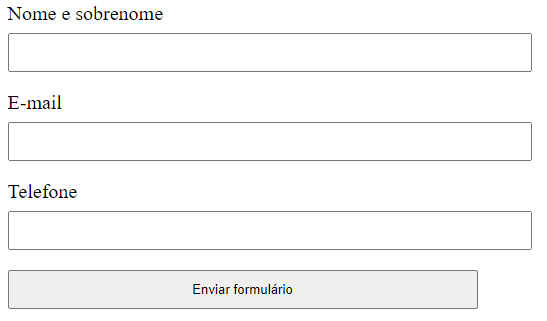
    display: block;

    margin: 0 0 20px;

    padding: 10px 25px;

    width: 50%;

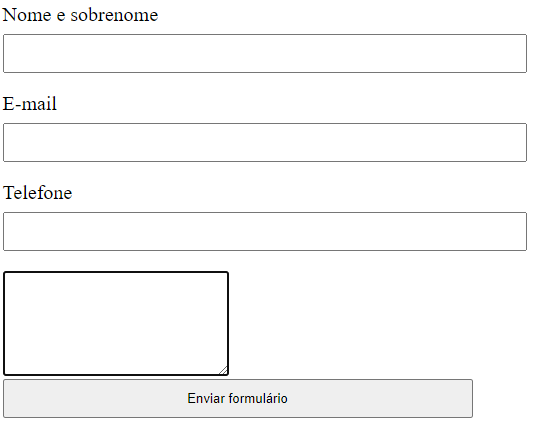
}



**Formulário que se coloca mais de uma linha de texto:**

<textarea> </textarea> é uma tag de conteúdo. Logo é uma caixa de texto que pode alterada o tamanho manualmente pelo usuário, ou podemos delimitar as linhas e coluna dessa caixa. O textarea, é usado para entradas de texto de mais de uma linha.

<textarea></textarea>



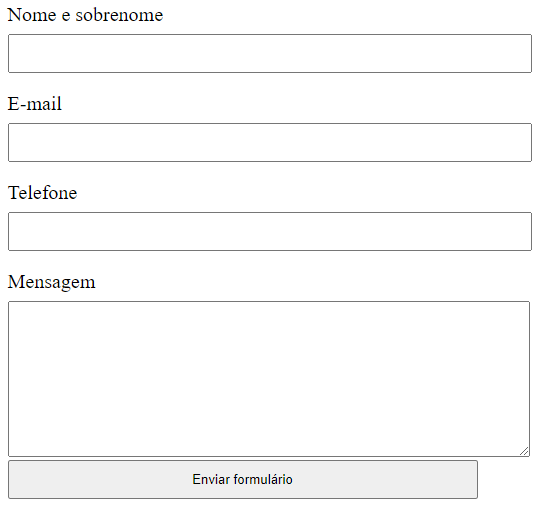
Delimitando/padronizando tamanho da área da caixa (linhas e colunas) de texto:

 <textarea cols="68" rows="10"></textarea>

<label for="mensagem">Mensagem</label>

<textarea cols="68" rows="10" id="mensagem"></textarea>

<textarea> com label e id.



**Formulário de selecionar:**

<label for="radio-email">E-mail</label>

<input type="radio" value="email" id="radio-email">

    <main>

        <form>

                <label for="nomesobrenome">Nome e sobrenome</label>

                <input type="text" id="nomesobrenome">

                <label for="email">E-mail</label>

                <input type="text" id="email">

                <label for="telefone">Telefone</label>

                <input type="text" id="telefone">

                <label for="mensagem">Mensagem</label>

                <textarea cols="68" rows="10" id="mensagem"></textarea>

                <label for="radio-email">E-mail</label>

                <input type="radio" value="email" id="radio-email">

                <label for="radio-telefone">Telefone</label>

                <input type="radio" value="telefone" id="radio-telefone">

                <label for="radio-whatsapp">Whatsapp</label>

                <input type="radio" value="whatsapp" id="radio-whatsapp">

                <input type="submit" value="Enviar formulário">

        </form>

    </main>



Colocar formulário de selecionar no mesmo grupo, para que quando clique em um ele desmarque o restante, só pode ficar um selecionado:

 <div>

                    <p>Como prefere o nosso contato?</p>

                    <label for="radio-email">E-mail</label>

                    <input type="radio" name="contato" value="email" id="radio-email">

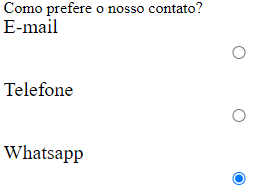
                    <label for="radio-telefone">Telefone</label>

                    <input type="radio" name="contato" value="telefone" id="radio-telefone">

                    <label for="radio-whatsapp">Whatsapp</label>

                    <input type="radio" name="contato" value="whatsapp" id="radio-whatsapp">

</div



*Qual a propriedade de um input do tipo radio que preciso criar para que eles façam parte do mesmo grupo?*

A propriedade name só pode ser "preenchida" uma única vez, por isso que, quando eu seleciono um dos itens, ele desmarca o outro, mantendo somente um selecionado.

**Formulário – Criando formulário de Checkbox**

<label><input type="checkbox">Gostaria de receber nossas novidades por e-mail?</label>



**Formulário – ajustando e padronizando:**

HTML:

<main>

        <form>

                <label for="nomesobrenome">Nome e sobrenome</label>

                <input type="text" id="nomesobrenome" class="input-padrao">

                <label for="email">E-mail</label>

                <input type="text" id="email" class="input-padrao">

                <label for="telefone">Telefone</label>

                <input type="text" id="telefone" class="input-padrao">

                <label for="mensagem">Mensagem</label>

                <textarea cols="68" rows="10" id="mensagem" class="input-padrao"></textarea>

                <div>

                    <p>Como prefere o nosso contato?</p>

                    <label for="radio-email"><input type="radio" name="contato" value="email" id="radio-email">E-mail</label>

                    <label for="radio-telefone"><input type="radio" name="contato" value="telefone" id="radio-telefone">

                        Telefone</label>

                    <label for="radio-whatsapp"><input type="radio" name="contato" value="whatsapp" id="radio-whatsapp">Whatsapp</label>

                </div>

                <label class="checkbox"><input type="checkbox">Gostaria de receber nossas novidades por e-mail?</label>

                <input type="submit" value="Enviar formulário">

        </form>

    </main>

CSS:

form {

    margin: 40px 0;

}

form label, form p {

    display: block;

    font-size: 20px;

    margin: 0 0 10px;

}

.input-padrao {

    display: block;

    margin: 0 0 20px;

    padding: 10px 25px;

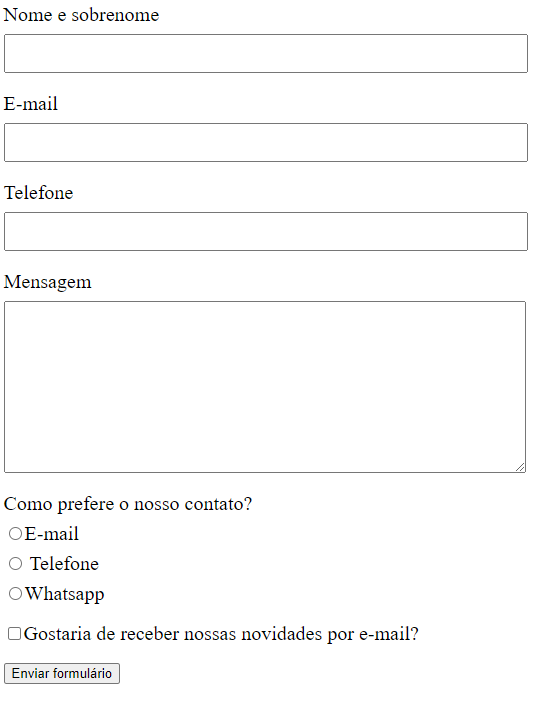
    width: 50%;

}

.checkbox {

    margin: 20px 0;

}



**CSS repetido:**

*Qual a melhor abordagem quando temos uma configuração de CSS aplicada para um item e queremos replicar ela para um item parecido?*

- Usar os seletores dos itens separados por vírgula. Essa é uma forma simples de fazermos a mesma configuração de CSS funcionar para os dois itens, sendo eles seletores dos elementos, de ids ou de classes.

- Criar uma classe. Quando temos um estilo que pode ser repetido, uma excelente estratégia é extrair isso para uma classe e usar a classe para aplicar esses estilos nos nossos elementos.

form label, form p {

    display: block;

    font-size: 20px;

    margin: 0 0 10px;

}

.input-padrao {

    display: block;

    margin: 0 0 20px;

    padding: 10px 25px;

    width: 50%;

}

**Hieráquia CSS (ordem de prioridade):**

0 – Estilo inline -> style="color: purple"

1 – id -> #teste

2 – class -> .class

3 – tag -> p

Exemplo de força:



Ganha a prioridade quem tiver mais, logo precisamos pensar em qual especifico é o nosso marcador, para que possamos tomar melhores decisões quando criamos o CSS. Por isso recomenda-se sempre criar classes para os elementos. C

**Formulário de opções de seleção (caixa de opção) – estilo dropdown:**

O select, que é seletor, um campo de seleção de um item, e o option, que representa cada opção do seletor.

<select>

                        <option>Manhã</option>

                        <option>Tarde</option>

                        <option>Noite</option>

                    </select>

Ou

<div>

                    <p>Qual o horário prefere ser atendido?</p>

                    <select>

                        <option>Manhã</option>

                        <option>Tarde</option>

                        <option>Noite</option>

                    </select>

                </div>



*A tag <option>. Qual a sua utilidade?*

*Para criarmos os itens selecionáveis de um <select> . A estrutura da tag <select> é composta de um ou mais <option>.*

**Acessibilidade em formulários em HTML5 – via celular**

Deixar os campos adaptados para celular.

<http://mobileinputtypes.com/>

O site mobile input types mostra visualmente qual o resultado esperado em um celular. Exemplos:

* <input type="text">
* <input type="email">
* <input type="number">
* <input type="password">
* <input type="date">
* <input type="datetime">
* <input type="month">
* <input type="search">
* <input type="tel">

Por quais motivos é importante usar os tipos corretos de <input> para os usuários que acessam a página via celular?

- Para facilitar a vida do usuário. É extremamente importante facilitarmos a vida dos nossos usuários sempre.

- Para termos dados mais corretos no formulário. É muito melhor termos dados corretos quando temos o nosso formulário preenchido.

**Formulário – Fazer o campo ser obrigatório para o preenchimento (usado no input)**

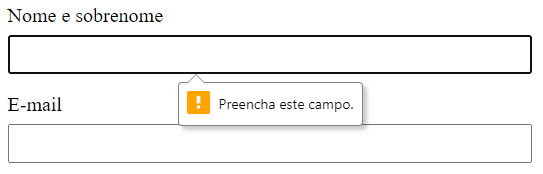
**required** - é usado para fazer o preenchimento do campo obrigatório.

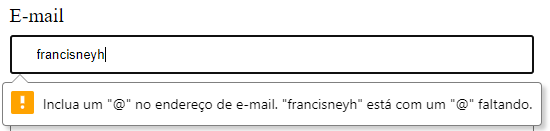
<label for="nomesobrenome">Nome e sobrenome</label>

<input type="text" id="nomesobrenome" class="input-padrao" required>

<label for="email">E-mail</label>

<input type="email" id="email" class="input-padrao" required>





Ao usar o tipo o type certo o navegador mostra para o usuário como precisa ser preenchido. Adicionando a propriedade required transformamos um campo comum em um campo de preenchimento obrigatório. Logo quando adicionamos a propriedade required, o navegador nos ajuda a validar se aquele campo está preenchido.

**Formulário – Mostrar sugestão de preenchimento (texto de exemplo de como preencher) - (usado no input)**

**placeholder** - cria uma sugestão de preenchimento que ocupa o campo até ser preenchido, ao preencher o texto de exemplo desaparece.

 <label for="email">E-mail</label>

<input type="email" id="email" class="input-padrao" required placeholder="usuario@email.com">

<label for="telefone">Telefone</label>

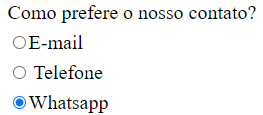
<input type="tel" id="telefone" class="input-padrao" required placeholder="(99) 99999-9999">



**Formulário – Sugerir opção, sempre deixar uma marcada por padrão (usado no input, radio, checkbox)**

checked – a opção fica predefinida marcada até o usuário mudar, logo é uma sugestão.

<label for="radio-whatsapp" ><input type="radio" name="contato" value="whatsapp" id="radio-whatsapp" checked>Whatsapp</label>



**Formulário – Deixar o checkbox marcado (usado no input)**

<label class="checkbox"><input type="checkbox" checked>Gostaria de receber nossas novidades por e-mail?</label>



**Formulário – Fazer o formulário ser mais semântico, mais estruturado e fácil de ser lido:**

<**fieldset**> </fieldset> referente a configuração de um ou mais campos referentes a um assunto especifico. Exemplo: formulários, preenchimento cartões, dados de um endereço. Dentro fieldset não tem parágrafo <p> mas títulos. Os títulos em um <fieldset> são chamados de <legend>. A tag <legend> utilizamos para marcar um título de um grupo de campos em qualquer formulário.

HTML:

<fieldset>

<legend>Como prefere o nosso contato?</legend>

<label for="radio-email"><input type="radio" name="contato" value="email" id="radio-email">E-mail</label>

<label for="radio-telefone"><input type="radio" name="contato" value="telefone" id="radio-telefone">Telefone</label>

<label for="radio-whatsapp" ><input type="radio" name="contato" value="whatsapp" id="radio- whatsapp"checked>Whatsapp</label>

 </fieldset>

<fieldset>

<legend>Qual o horário prefere ser atendido?</legend>

<select>

<option>Manhã</option>

<option>Tarde</option>

<option>Noite</option>

</fieldset>

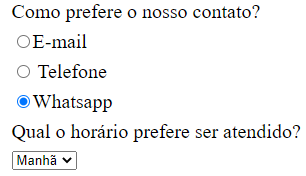
form label, form legend {

    display: block;

    font-size: 20px;

    margin: 0 0 10px;

}



**Acessibilidade no site, alternativa para imagem:**

**Alt** – usamos o alt como alternativa, para melhorar a acessibilidade do site, dessa forma quem tem baixa visão sabe o que está na página, no caso uma imagem. Adiciona uma alternativa à imagem, descrevendo-a, com o atributo alt.

<h1><img src="logo.png" alt="Logo para Barbearia Alura"></h1>

<img src="logo-branco.png" alt="Logo para Barbearia Alura">

**Formulário – Botão – alterando tamanho, fonte, borda, cor, etc**

<input type="submit" value="Enviar formulário" class="enviar">

.enviar {

    width: 40%;

    padding: 15px 0;

    background: orange;

    color: white;

    font-weight: bold;

    font-size: 18px;

    border: none;

    border-radius: 5px;

}

.enviar:hover {

    background: red;

}







Com o hover quando passamos o mouse por cima do botão ele fica laranja escuro instantaneamente, sem suavidade, ou seja a transição de cores do laranja claro para o vermelho (ou laranja escuro) não é suave.

**Transição – suavizando as transições**

transition – permite suavizar, ou seja colocar um delay de tempo para que a transição não seja realizada instantaneamente. Logo o transition permite configurar a transição visual de um elemento, dessa forma utilizamos a propriedade transition para prover uma transição visual de um elemento.



**Formulário – Botão – Cursor mostrar a mão quando estiver encima de um elemento**

cursor: pointer;

Mostra que o visualizador do mouse é a mão não a seta. Para mostrar para o usuário que o botão é clicavel.

.enviar {

    width: 40%;

    padding: 15px 0;

    background: orange;

    color: white;

    font-weight: bold;

    font-size: 18px;

    border: none;

    border-radius: 5px;

    transition: 1s background;

    cursor: pointer;

}

**Formulário – Botão – Aumentar o tamanho do botão quando estiver com o curso posicionado por cima:**

transform – aumenta o tamanho do botão/elemento.

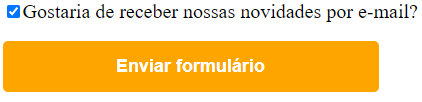
transform: scale(1.2);

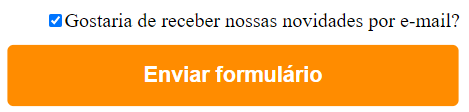
.enviar:hover {

    background: darkorange;

    transform: scale(1.2);

}





*Qual propriedade do CSS usamos para aumentar um elemento proporcionalmente?*

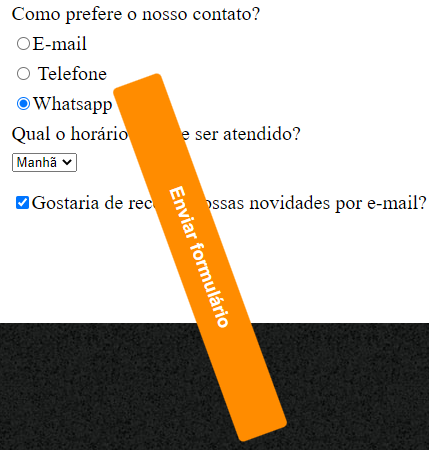
*transform: scale() . Utilizamos a propriedade transform: scale() para aumentar um elemento proporcionalmente.*

**Aplica a transição em todos elementos para que a transição seja suavizada:**

transition: 1s all;

**Mudar o grau de um elemento:**

transform: rotate(70deg);



Ordem de sobrescrita do CSS:

No CSS a leitura sempre é feita de cima para baixo. Logo o último sobrescreve o último o anterior.

.enviar:hover {

    background: darkorange;

    transform: scale(1.2);

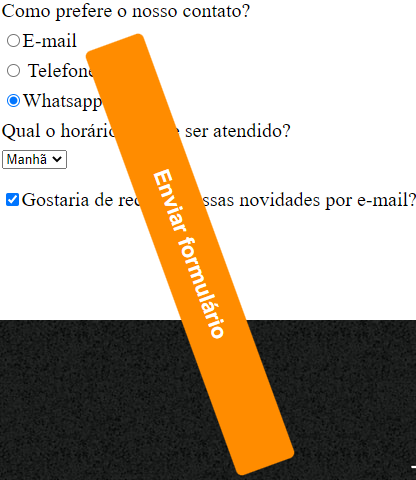
    transform: rotate(70deg);

}

Logo a propriedade transform: scale(1.2); vai ser ignorado e só vai ler o transform: rotate(70deg);

Para que leia/some as duas propriedades no transform precisa juntar elas, dessa forma:

transform: scale(1.2) rotate(70deg);



Dessa forma foi aplicado o aumento de escala em 20% (1.2) e a rotação no elemento.

**Tabela – criação de tabela**

A tabela é um item simples, mas trabalhoso de se criar pois exige que se utilize muitas tag, estas muito fáceis de se entender.

<table> </table> cria uma tabela, esta que é um sequência de linhas e colunas.

<tr> cria uma linha.

<td> cria uma célula que armazena os dados/conteúdo que queremos.

<table>

            <tr>

                <td>Dia</td>

                <td>Horário</td>

            </tr>

            <tr>

                <td>Segunda</td>

                <td>8h às 20h</td>

            </tr>

            <tr>

                <td>Quarta</td>

                <td>8h às 20h</td>

            </tr>

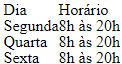
            <tr>

                <td>Sexta</td>

                <td>8h às 20h</td>

            </tr>

 </table>



Dentro de uma tabela podemos separa-la em três partes: cabeçalho, conteúdo, rodapé.

*Dividir uma tabela de forma mais semântica:*

*As tags <thead>, <tbody> e <tfoot> ajudam a deixar o conteúdo da tabela mais bem dividido e mais semântico.*

<thead></thead> cabeçalho da tabela.

<th> célula pertence ao cabeçalho da tabela, conteúdo fica centralizado na tabela.

<tbody></tbody> corpo da tabela, conteúdo.

 <table>

            <thead>

                <th>Dia</th>

                <th>Horário</th>

            </thead>

            <tbody>

                <tr>

                    <td>Segunda</td>

                    <td>8h às 20h</td>

                </tr>

                <tr>

                    <td>Quarta</td>

                    <td>8h às 20h</td>

                </tr>

                <tr>

                    <td>Sexta</td>

                    <td>8h às 20h</td>

                </tr>

            </tbody>

        </table>

**Tabela – Configurando o estilo da tabela - estilizar a tabela:**

table {

    margin: 20px 40px;

}

thead {

    background: #555555;

    color: white;

    font-weight: bold;

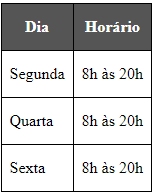
}

td, th {

    border: 1px solid #000000;

    padding: 15px 8px;

}



**Tabela – juntar/mesclar células**

<tr>

<td colspan="5">Rio de Janeiro</td>

</tr>

**Tabela - tags**

* A tag table, que representa a tabela
* A tag tr, que representa a linha da tabela
* A tag td, que representa a célula da tabela
* A tag thead, que representa o cabeçalho da tabela
* A tag tbody, que representa o corpo da tabela
* A tag th, que representa a célula do cabeçalho da tabela
* A tag tfoot, que representa o rodapé da tabela

**HTML – Conteúdo em um só, fechado como se fosse uma sessão, para um objetivo**

**<section></section>**

*Diferença de <div> para section:*

*<div> é só divisão visual, e a*

*<section> é uma divisão onde aquilo tem um conteúdo complexo, tudo semanticamente faz o mesmo sentido, título e um texto para a mesma coisa.*

*Qual a diferença entre usar uma <div> ou uma <section>?*

*Só devemos usar <section> quando o bloco for semântico. ara um bloco onde o conteúdo tenha o mesmo significado, o mesmo sentido, usamos uma <section>.*

<section class="principal">

<h2 class="tituloCentralizado">Sobre a Barbearia Alura</h2>

      <p>Localizada no coração da cidade a <strong>Barbearia Alura </strong>traz para o mercado o que há de melhor para o seu cabelo e barba. Fundada em 2019, a Barbearia Alura já é destaque na cidade e conquista novos clientes a cada dia.</p>

       <p id="missao"><em>Nossa missão é: <strong>"Proporcionar auto-estima e qualidade de vida aos clientes"</strong></em></p>

       <p>Oferecemos profissionais experientes e antenados às mudanças no mundo da moda. O atendimento possui padrão de excelência e agilidade,

</section>

<section class="beneficios">

<h3 class="tituloCentralizado">Benefícios</h3>

<ul>

      <li class="itens">Atendimento aos clientes</li>

            <li class="itens">Espaço diferenciado</li>

            <li class="itens">Localização</li>

            <li class="itens">Profissionais Qualificados</li>

</ul>

<img src="beneficios.jpg" class="imagemBeneficios">

</section>

**Comentário CSS:**

/\* \*/

/\* css da página inicial \*/

**OBS:**

* Não é ideal criar estilos usando TAGS, o ideal é usar classe para tudo.
* Se a imagem do logo for a primeira coisa que vai exibir para os usuários, usar a TAG <h1> no cabeçalho (header).

<h1><img src="logo.png"></h1>

* Por padrão através do navegador o mesmo coloca uma borda branca, cor do texto do link azul e o visual do link sublinhado, ele também cria estilos que não pedimos. Logo a comunidade se juntou e criou um arquivo de reset.css para zerar o que o navegador está fazendo, logo é um arquivo básico que é muito utilizado para tirar os arquivos que o navegador cria automaticamente. Logo o link arquivo de reset.css é colocado antes no head para limpar.

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

* Podemos dimensionar a <div> e os conteúdos que estão dentro da <div> de forma padronizada. A largura/width: 940px é padrão na web, porque a maioria das boas resoluções é de 1260px de largura/width, logo utilizando width: 940px está em uma boa margem.
* TAGS Semânticas: Deixam o código mais legível, tanto para nós, quanto para o navegador. Logo as tags semânticas fazem com que o navegador entenda melhor o nosso conteúdo, e com isso carregue melhor a nossa página.

Indentação: sempre indentar o código para ele ficar mais coerente e fácil de ser lido.

* O CSS é feito para apresentação. Quando se quer criar uma ação/comportamento tipo a página recarregar, preencher algum item, precisa utiliza JavaScript.
* Quando temos um arquivo CSS principal ele recebe o nome style.css , é o nome padrão.
* Sempre que temos imagem em uma página precisamos ter uma atenção tremenda, pois se um usuário com baixa visão acessar ele não vai conseguir ver.
* No CSS a leitura sempre é feita de cima para baixo. Logo o último sobrescreve o último o anterior.
* Sempre que queremos falar de estilo usamos uma classe, e sempre que queremos falar de comportamento usamos um identificador.