**CURSO DE PROGRAMAÇÃO**

**Curso de HTML5, CSS3 e JAVASCRIPT**

Domínio - <https://gustavoguanabara.github.io/>

***HTML5***

{} chaves

< > Colchetes angulares

Gustavoguanabra – sub domínio

Github – domínio

IO – TLD

HTTP’s – protocolo

**HTML – conteúdo, constitui-se de:**

* Textos
* Tabela
* Números
* Imagens
* Vídeos

**CSS – Design, constitui-se de:**

* Cores
* Sombras
* Tamanhos
* Posicionamento

**JAVA SCRIPT – Interatividade, constitui-se de:**

* Menus
* Animações
* Pop ups
* Validações

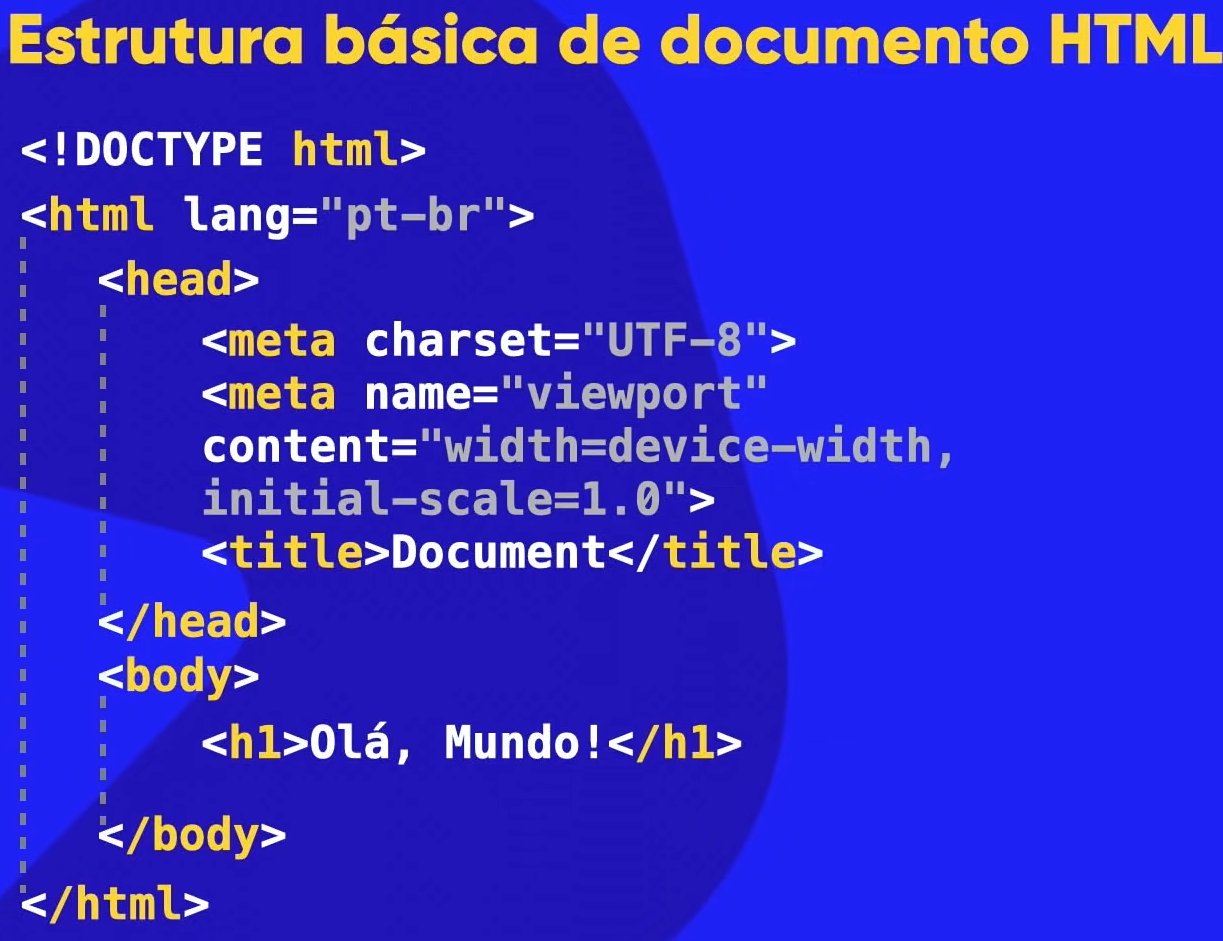
**Comandos em HTML**

**Para colocar títulos ou Tags em HTML** usamos a tag **<h>** para iniciar e **</h>** para fechar.

Exemplo: **<h1>**titulo**</h1>**

**Para usarmos parágrafos em HTML** usando a tag **<p>** para iniciar e **</p>** para fechar.

Exemplo: **<p>**parágrafo**</p>**



* Dentro do **VScode**, ao criarmos um novo arquivo com a tag **index.html** e digitamos ! (ponto de exclamação) o mesmo insere automaticamente a estrutura básica do HTML.
* Para criarmos uma linha horizontal no **VScode** usamos a tag **<hr>**
* Para usarmos quebras de linha no **VScode** usamos o comando **<br>** que significa **Break Roll =** quebra de linha.
* Para destacarmos ícones em um site usamos as tags **&lt;ICONE&gt;**
* Para adicionarmos comentários dentro do **VScode**, a tag **<!--** com isso podemos usar comentários dentro do programa sem que ele afete nosso layout externo do site.

Para adicionarmos Emojis dentro do **VScode** sempre usamos as tags **&#x+** o código do emoji e finalizamos com ; (ponto e vírgula).

* Para adicionarmos imagens dentro do **VScode** usamos a tag **img,** sendo que a tag **ALT** é o campo de explicação da imagem.
* Para mudarmos o **Favicon** de um site, usamos a tag **Link** encima da linha de **tittle**, escolhemos a opção **link:favicon**, apagamos o conteúdo entre as “ “ (aspas) e apertamos **CTRL+Espaço** para selecionarmos um ícone dentro da pasta do index ou subpastas.
* Para adicionarmos parágrafos randômicos dentro do visual code primeiro criamos um parágrafo **<p>** e depois inserimos a tag **lorem**.
* **H1** e o título principal, sendo **h2** o subtítulo e **h3** o subtítulo do subtítulo, podendo haver até 6 camadas sendo o **h6** o máximo.
* Para colocarmos um texto em negrito usando a tag **<b>** ou usando a tag **STRONG**.
* Para usarmos textos em itálico usamos a tag **<i>** e arrastamos o texto para dentro do comando ou podemos selecionar a parte do texto que queremos colocar em itálico usar a tag **CRTL+SHIT+P** e digitar o comando **ABB**, apertamos a tecla I e damos Enter.
* O comando **CRTL+SHIT+P** Pode ser usado para diversas funções como colocar títulos h1, parágrafos **<p>,** destacar palavras **<b>,** **<STRONG>,** **<I>** ou **<EM>.**
* Paga marcarmos um texto usando a tag **MARK** junto com a tag **CRTL+SHIT+P.**
* Para mudarmos a cor da marcação usamos a tag **MARK** junto com a tag **STYLE** juntamente coma opção background-color.
* Para mudarmos todas as marcações em geral no texto inserimos a tag **STYLE** logo abaixo do títulos, onde vamos inserir a tag **MARK+{}** onde vamos selecionar a opção background-color.
* Para colocarmos um texto onde o mesmo deve ser lido, mas não considerado usamos a tag **CRTL+SHIFT+P** junto da tag **del**.
* Para marcarmos um texto como inserido para dar uma ênfase e indicar que ele foi adicionado depois usamos a tag **CRTL+SHIFT+P** seguido da tag **ins**.
* Para adicionarmos uma palavra, número sobre um texto usamos a tag **CRTL+SHIFT+P** com o comando **SUP (SUPERIOR).**
* Para adicionarmos uma palavra, número abaixo de um texto usamos a tag **CRTL+SHIFT+P** com a tag **SUB(INFERIOR).**
* Para deixarmos uma fonte com os mesmos padrões de tamanho envelopamos com a tag **CODE**.
* Para adicionarmos tubulações em códigos envelopamos a tag **CODE** com a tag PRE, então poderemos fazer várias coisas como adicionar espaços apertando ENTER, ou recuar parágrafos selecionando o texto e apertando os comandos SHIFT+TAB para recuar ou somente TAB para darmos espaço.
* Para colocarmos citações simples e sem quebra, usamos o envelopamento junto da tag **Q**.
* Para colocarmos abreviações das palavras envelopamos a palavra que queremos abreviar com a tag **ABBR**.
* Para fazer citações completas com link usamos a tag **Blackquote**.
* Para elaborarmos um texto em listas simples, usamos a tag **OL** e dentro do comando **OL** usamos a tag **LI**.
* Para colocarmos uma lista em sequência usamos a tag **type** para termo opções de numeração podemos usar a tag **start** (padrão numérico) se quisermos começar a lista de numerações dizentes do número 1.
* Para colocarmos listas não ordenadas usamos a tag **UL** e podemos colocar a tag **type+circle**,**disk** ou **square**.
* Para colocarmos um termo usamos o comando **DT** e para colocarmos a descrição de um termo usamos a tag **DD** abaixo de **DT.**
* Para adicionarmos links externos no site, usamos a tag de envelopar seguido da tag A, onde colocamos o endereço dentro da tag **href.**
* Para mantermos o cliente em nossa página quando ele clicar em links em nosso site, usando a tag **rel=\_ blank e targuet=external** assim o novo link irá abrir em uma nova janela.
* Para definirmos qual será a próxima página que queremos acessar clicando nos links usamos a tag **rel=next** e a anterior usamos a tag **rel=prev.**
* Para voltarmos na página inicial ou secundaria, envelopamos a frase que queremos e usamos a tag ../ e então escolhes quais das opções queremos colocar como opção de volta.
* Para inserirmos um link de download usamos a tag de lista desorganizada **UL** ou lista organizada **OL** e dentro colocamos a tag **LI** de lista e envelopamos com a tag de link **A**.
* Para adicionarmos imagens referentes a links de **youtube**, **facebook**, **linkedin** ou afins salvamos uma imagem com proporção 16x16 no **gimp** ou baixamos direto do site **icon archive** e inserimos dentro do visual code usando as tags **UL** ou **OL** seguidos de **LI** e **img,** depois envelopamos usando o comando **CRTL+SHITF+P** selecionamos e usando a tag de inserir links **A** e posteriormente inserimos o link da rede social e depois usamos a tag **rel\_blank** e target\_**external** para abrir os links em uma nova aba.
* Para inserirmos imagens dinâmicas para um site usamos a tag **Picture** e depois inserimos a tag **source:media:type.** É importante sempre colocar o número de dimensionamento da imagem e a sigla px no final da numeração.
* Podemos pegar músicas gratuitas da biblioteca de áudio do **youtube**.
* Para inserirmos áudio em um site usamos a tag **áudio** em seguida selecionamos o arquivo de música que deve estar dentro de uma pasta do projeto que está em aberto e usamos a tag **controls autoplay** para adicionarmos um miniplayer de áudio dentro do site com a música escolhida, a maioria dos navegadores são compatível com 3 formatos, sendo eles mp3/ogg/wav, podemos fazer as músicas rodarem automaticamente simplesmente adicionando a tag **loop** a tag **controls autoplay**, ficando assim **controls autoplay loop**.
* Usamos o **gimp** para editar imagens e o **handbreak** para redimensionar formatos de vídeos.
* Para inserir vídeos dentro de um site temos que nos atentar ao tamanho do vídeo, um tamanho maior fará o cliente pagar mais valor pelo trafego dos dados, usamos a tag vídeo, podemos ajustar o tamanho do vídeo no site usando a tag **width** e depois inserimos a tag **controls** para colocarmos o player de vídeo, lembrando que o arquivo do vídeo deve estar na mesma pasta do projeto.
* Para economizarmos trafego de dados em nosso site, podemos usar um link do vídeo externo para economizar banda, podemos usar o **vimeo/youtube**, simplesmente clicamos em incorporar, copiamos o conteúdo e colamos dentro do **VScode**.

***CSS3***

* Para mudarmos o fundo do site vamos até a tag **body** e inserimos a tag **style** e depois a tag **backgroud:color** e inserimos a cor que queremos.
* Para mudarmos a corda dos títulos **h1** ou subtítulos **h2** e assim subsequente usamos o comando **style:color** dentro do **h1** afins, exemplo: **h1** **style:color:blue**.
* Para mudarmos a fonte adicionamos a tag **font** logos depois de **body** e selecionamos a que nos agrada, importante sempre adicionar os comando de CSS após o ponto e vírgula ( ; ).
* Para alinharmos um texto usamos o comando **style:text-align** e escolhemos entre as várias opções, justificar, centralizar e afins.
* Podemos mudar o tamanho da fonte usando o comando **font:size**, colocamos o tamanho de fonte queremos e inserimos o tamanho e colocamos px.
* Podemos aplicar estilos dentro de um site usando o comando **style** abaixo do **title**, dentro do comando **style** usamos o comando body usando chaves {}e usamos a tag que queremos aplicar no site. Lembrando tudo que estiver dentro da tag body terá o efeito aplicado, importante sempre colocar ponto e vírgula no fim de cada declaração, não precisa colocar espaço entre as tag’s.
* Podemos aplicar o mesmo estilo de CSS em vários sites usando a tag link:css e criando uma nova página, podemos colocar o nome que queremos, após isso apertamos **CRTL+clique** sobre o nome do arquivo que colocamos e criamos uma nova página, dentro desta página vamos trabalhar todos os estilos de CSS que queremos, podendo ser aplicado para inúmeras páginas, outro modo e copiando o conteúdo **style** e body e colando nas páginas, porém e muita mais trabalhoso, **exemplo abaixo:**

    <title>Site de Exemplo</title>

    <link rel="stylesheet" href="cores02.css">

* **CRTL+x** recorta o conteúdo dentro do **VScode**, em caso de incompatibilidade usamos a tag **@charset "UTF-8";**
* Sempre colocamos os estilos dentro do início das tags, se quisermos estilos únicos por páginas, precisamos aplicar direta e de maneira individual em cada página.
* O **style inline** sempre vai priorizar o site, depois o local e depois o externo.
* Para inserirmos degrade em CSS em nosso site, usamos a tag **style** seguido pela tag **body** com as chaves {} depois inserimos a tag **background-image:linear-gradient**(escolhemos a posição do degrade, geralmente usamos to right, left, top ou bottom, colocamos virgula e adicionamos as cores que queremos sem espaço separado por virgula, **exemplo abaixo:**

    <title>Gradiente em CSS</title>

    <style>

        \*{ /\*configurações globais para CSS\*/

            height: 100%;

        }

        body{

  background-image:linear-gradient(to top,#FF8102,#C72394,#8832B1);

background-attachment: fixed;

        }

    </style>

* Usamos a tag **background-attachment: fixed;** para deixarmos o gradiente fixo dentro do site.
* Podemos mudar o ângulo do gradiente substituindo o to top,left,right,bottom por ângulos+deg ou até mesmo negativo, além disso podemos colocar porcentagem de exibição nas cores, **exemplo abaixo:**
* **background-image: linear-gradient(45deg,#FF8102,#C72394,#8832B1);**
* Usamos a tag **background-attachment: fixed;** para fixar o degrade em nosso site.
* Usamos a tag **width** com o dimensionamento em **px** para dimensionar o tamanho do fundo de um conteúdo ou texto.
* Usamos a tag **padding** com o dimensionamento em **px** para aumentar ou diminuirmos a margem de um texto ou fundo.
* Usamos a tag **margin:auto** para centralizarmos um texto ou conteúdo no meio do site.
* Usamos a tag **main** para representarmos o conteúdo principal do site, podendo ele ser diverso, imagem, texto ou afins.
* Usamos a tag **border-radius** para arredondarmos as bordas do conteúdo principal (main).
* Usamos a tag **box-shadow** para colocarmos sombra em nosso conteúdo principal, sendo o parâmetro inicial:
* Para o lado
* Para baixo
* Espalhar
* Transparência
* Usamos a tag **text-shadow** para adicionarmos cores aos títulos, subtítulos ou parágrafos, sempre usando o dimensionamento que queremos com a tag **px** no final, **exemplo abaixo:**

@charset "UTF8";

\*{

    font-family: Arial, Helvetica, sans-serif;

}

body{

 background-image:linear-gradient(to right,#F2A679,#D9695F);

}

main{

    background-color: white;

    width: 600px;

    padding: 50px;

    margin: auto;

    border-radius: 20px;

    box-shadow: 5px 5px 15px #3a1d1dcb;

}

h1{

    color:#8C4646;

    text-align: center;

    text-shadow: 1px 1px 2px #552b2b98;

}

h2{

    color: #5D8C7B;

    text-shadow: 1px 1px 1px #5D8C7B;

    text-align: left;

}

p{

    text-align: justify;

}

* Um site tem como tamanho natural de suas fontes 16px, recomendação oficialmente recomendadas, em ou px.
* Para mudarmos o peso da fonte usamos a tag font-weight, usando o **font-weight:bolder** o texto acaba ficando em negrito.
* Para mudarmos o estilo da fonte usamos a tag font-style.
* Podemos usar uma tag chamando **shorthand** para mudarmos a identidade das letras somente copiando o conteúdo das tag em uma única linha, adicionando suas propriedades, **exemplo abaixo:**

        h1{

            /\*shorthand font

            font-style->font-weight->font-size->font-family

            font-style: italic;

            font-weight: bolder;

            font-size: 3em

            font-family: 'work sans',sans-serif;

            font: italic bolder 3em 'work sans',sans-serif;

        }

        h2{

            font: bold 2em 'work sans';

        }

* Usando o **google fonts,** podemos copiar e colar estilos que gostaríamos dentro do **VSCODE**, **exemplo abaixo:**

<style>

@import url('https://fonts.googleapis.com/css2?family=Pacifico&display=swap');

 @import url('https://fonts.googleapis.com/css2?family=Cinzel:wght@400;500;600;700;800;900&display=swap');

        body{

           font-family: 'pacifico',Arial, Helvetica, sans-serif;

            font-size: 16px;

            color: black

        }

        h1{

            font-family: 'cinzel', Times, serif;

            font-size: 5em;

        }

        h2{

            font-family: 'pacifico', Times, serif;

            font-size: 3em;

        }

</style>

* Para adicionarmos fontes externas, baixamos a fonte, um site bom e o Dafont, depois do download, extraímos e recortamos ou copiamos o arquivo para dentro da pasta do projeto ou criamos uma pasta dentro da pasta do projeto. Depois usamos a tag **@font-face**{ para adicionarmos essa fonte externa, no campo src: url podemos usar o comando CRTL+ESPAÇO e selecionarmos a fonte diretamente da pasta, **exemplo abaixo:**

<style>

        @font-face {

            font-family: 'lol';

            src: url('Black\ Way\ -\ Personal\ Use.otf')

        format('opentype')}

        @font-face {

            font-family: 'lol2';

            src: url('love\ larry\ ttf.ttf')

format('truetype')}

</style>

* Depois podemos livremente adicionar esse estilo de fonte para onde quisermos, títulos <h1>, subtítulos <h2> ou parágrafos <p>, basta colocarmos o nome da **font-family** que escolhemos para a fonte externa, **exemplo abaixo:**

        }

        h1{

            font-family: 'lol';

            font-weight: lighter;

            font-size: 5em;

        }

        h2{

            font-family: 'lol2';

            font-weight: normal;

            font-size: 5em;

        }

* Podemos capturar fontes de qualquer site usando a extensão para o google chrome **Fonts Ninja**, após identificada a fonte que queremos copiamos o nome e a buscamos em sites de fontes como o **Google fonts** ou o **DAfont.**
* Podemos capturar fontes direto de imagens, para isso somente salvamos e depois colamos e arrastamos as imagens para dentro do site **find Picture**.com, selecionamos a área da imagem que contem a fonte que queremos e depois ajudamos o site a identificar os caracteres da imagem, após isso o site nos dará fontes iguais ou semelhantes aquelas, então copiamos o nome da fonte e buscamos a mesma caso seja gratuita em site de fonts, abaixo um exemplo de uma fonte encontrada de uma imagem e identificado no site **font finder** e baixada do site **DAfont**, **exemplo abaixo:**

    <title>Capturando fontes em imagens</title>

    <style>

        @font-face {

            font-family:'hoedown';

            src: url('Hoedown-Shadow.ttf')

            format('opentype')

        }

        @font-face {

            font-family:'hoedown2';

            src: url('Hoedown.ttf')

            format('opentype')

        }

        h1{

            font-family: 'hoedown';

            font-size: 5em;

        }

        h2{

            font-family: 'hoedown2';

            font-size: 5em;

        }

    </style>

* Abaixo estão os formatos de tipos de fontes que devemos inserir na tag **format,** **exemplo abaixo:**

            tipos de formatos

            - opentype (otf)

            - truetype (ttf)

            - embedded-opentype

            - truetype-aat

            - svg

* Para darmos espaçamento nos parágrafos usamos a tag **text-indent: 20px**, podemos alinhar os textos usando o comando **text-align**: ele nos dará um leque de opções que podemos escolher conforme acharmos melhor, podemos centralizar ao meio, a direita ou a esquerda, **exemplo abaixo:**

    <title>Alinhando textos e imagens em CSS</title>

    <style>

        body{

            font: normal 16px arial, verdana, serif;

        }

        h1{

            text-align: center;

        }

        h2{

            text-align: right;

        }

        p{

            text-align: justify;

            text-indent: 30px;

        }

    </style>

* Para mudarmos os estilos dos textos de maneira individual, além do comando inline podemos identificar as tags como principais, assim atribuindo estilos e afins de maneira única neles, **exemplo abaixo:**

 <h1 id="principal">Criando Sites com HTML e CSS</h1>

h1#principal{

    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

    text-align: center;

    background-color: rgb(66, 117, 11);

    color: rgb(182, 209, 233);

}

* Temos seletores únicos para personalização que sofrem transformações quando aplicado de HTML para CSS, **exemplo abaixo:**

em HTML é id = em CSS é # (cerquilha)

  em HTML é class = em CSS é . (ponto)

: (dois pontos) significa uma pseudo-classe

* Quando já tivermos nosso **ID** principal, para usamos estilos ou afins em outros textos que gostamos usamos a tag **class=’nome que quisermos’** isso em HTML, depois vamos para nossa página de CSS e colocamos a **tag+ponto+nome** que escolhemos para a class, e então trabalho dentro do texto o que quisermos, cores, tamanhos, fundos ou afins, **exemplo abaixo:**

<h2 class="básico">HTML Básico</h2>

<h2 class="intermediário">HTML intermediário</h2>

<h2 class="avançado">HTML Avançado</h2>

<h2 class="básico">CSS básico</h2>

<h2 class="intermediário">CSS intermediário</h2>

<h2 class="avançado">CSS Avançado</h2>

<h1 class="destaque">Aprendendo CSS</h1>

**PARTE EM HTML5 ACIMA**

.básico{

    color: red;

}

.intermediário{

    color: blue;

}

.avançado{

    color: rgb(163, 163, 43);

}

}

.destaque{

    background-color: darkmagenta;

}

**PARTE EM CSS3 ACIMA**

* Podemos colocar mais de uma classe, sendo necessário somente separá-las com espaço.
* Podemos usar a tag **div** para criarmos formas dentro de um site, assim usamos varias tag para dimensionar a largura e altura da forma, cor e afins, **exemplo abaixo:**

<style>

        div{

          /\*tudo na mesma linha\*/  display: inline-block;

          /\*cor de fundo\*/  background-color: lightgray;

          /\*borda\*/         border: 1px solid black;

          /\*alinhamento de texto\*/  text-align: center;

         /\*altura\*/   height: 200px;

         /\*largura\*/   width: 200px;

        /\* hover significa: quando eu passar o mouse encima\*/

}

        div:hover{

            background-color: yellow;

        }

    </style>

</head>

<body>

    <div id="d1">01</div>

    <div id="d2">02</div>

    <div id="d3">03</div>

* Podemos inserir texto escondidos e mostrarmos os mesmo quando passar o mouse sobre eles usando as tags **div > p { e div:hover > p{** é necessário somente inserimos um parágrafo dentro de uma div e inserirmos o texto que queremos deixar temporariamente oculto, primeiro usamos a tag **div > p { e a tag display:none;** para escondermos o texto, em seguida usamos a tag **div:hover >p{** e a tag **display:block.** Para revelarmos o texto quando passarmos o mouse por cima, **exemplo abaixo:**

<body>

    <h1>Exemplo de hover</h1>

    <p>Passe o mouse sobre o texto abaixo:</p>

    <div>Passe o mouse aqui

        <p>TEXTO ESCONDIDO</p>

    </div>

    <div id="d2">Passe o mouse aqui 2

        <p>TEXTO ESCONDIDO 2</p>

    </div>

    <p>fim do exemplo</p>

</body>

 div > p {

            display: none;

        }

        div:hover > p {

            display: block;

        }

* Podemos mudar a cor dos links de sites inserindo a tag **a**{ **exemplo abaixo:**

        a {

            color: red;

            text-decoration: none;

            font-weight: bold;

        }

* Podemos também mudar a cor dos links já visitados inserindo a tag **a:visited{** **exemplo abaixo:**

a:visited {

            color: darkgoldenrod;

        }

* Podemos também mudar a cor do link quando colocamos o mouse sobre ele e também adicionarmos um sublinhado usando a tag **a:hover**{ **exemplo abaixo:**

 a:hover {

            color: black;

            text-decoration: underline;

        }

* Podemos também inserirmos uma cor ao clicar em um site usando a tag **a:active**{ **exemplo abaixo:**

a:active{

            color:yellow

        }

* Podemos indicar em nosso site que a palavra do texto ou link é realmente um link, inserindo a tag **a::after**{ **exemplo abaixo:**

 a::after{

            content: '[link]';

            text-decoration: none;

            font-weight: normal;

            color: darkgray;

        }

* Podemos inserir ícones tanto antes quanto depois dos links usando a tag **a::before{ e a** tag **a::after{** então inserimos a tag **content:** e colamos os ícones que copiamos, **exemplo abaixo:**

        }

        a::before{

            content:'▶' ;

        }

        a::after{

            content: '◀'

        }

* Podemos personalizar link únicos adicionando uma classe a eles usando a tag class=’nome que quisermos’ e depois em CSS3 colocamos a tag .nome que escolhemos::before{ content:’icone’ para ícones, imagens e afins antes e a tag .nome que escolhemos:after{contente: ‘icone’ para ícones, imagens e afins, depois, **exemplo abaixo:**

        }

        .especial::before{

            content:'▶' ;

        }

        .especial::after{

            content: '◀'

<h1>Personalizando Links</h1>

   <ul>

       <li><a href="https://gustavoguanabara.github.io" target="blank">Repositório github</a></li>

       <li><a href="https://youtube.com/cursoemvideo/" class="especial">Canal do Youtube</a></li>

       <li><a href="https://www.cursoemvideo.com">Site do CursoemVideo</a></li>

   </ul>

* Para configuramos a altura e largura de um texto usamos as tag **height** (altura) e **width** (largura),
* podemos traçar uma linha ao redor do conteúdo usando a tag **border**,
* podemos mudar o distanciamento da borda do conteúdo usando a tag **padding**.
* Podemos mudar o distanciamento da borda externa do conteúdo usando a tag **margin**.
* Podemos mudar a corda da linha da borda usando a tag **outline**.
* Podemos verificar informações de sites, altura, largura margens e afins usando a ferramenta do google chrome **DevTools**, basta clicarmos em inspecionar.
* Podemos também testar os tamanhos e estilos dos conteúdos aplicados no site inspecionando o mesmo com o **DevTools,** as modificações usando o **DevTools** não mudam o formato original do conteúdo, **exemplo abaixo:**



* Podemos centralizar a margem usando a tag margin:**auto**
* Podemos usar a tag **outline** para colocarmos uma linha entre a margem e a borda, **exemplo abaixo:**

            outline-width: 5px;

            outline-style: dashed;

            outline-color: blueviolet;

**SHORTHANDS**

* Podemos simplificar as tags usando uma única sequencia simples de tag, **exemplo abaixo:**

            border-width: 10px;

            border-style: solid;

            border-color: darkblue;

abaixo podemos ver o comando simplificado que da o mesmo resultado do que da imagem acima:

           border:10px solid darkblue;

Para usarmos o **shorthand** para a tag **padding**, usamos a tag **padding** seguindo o sentido de aplicação no sentido horário de um relógio sendo top,right,bottom,left, **exemplo abaixo:**

            padding-left: 10px;

            padding-top: 10px;

            padding-bottom: 10px;

            padding-right: 10px;

abaixo podemos ver o comando simplificado que dá o mesmo resultado do que da imagem acima:

            padding: 10px 10px 10px 10px;

Para usarmos o **shorthand** para a tag **margin**, usamos a tag **margin** seguindo o sentido de aplicação no sentido horário de um relógio sendo top,right,bottom,left, **exemplo abaixo:**

            margin-top: 20px;

            margin-right: 20px;

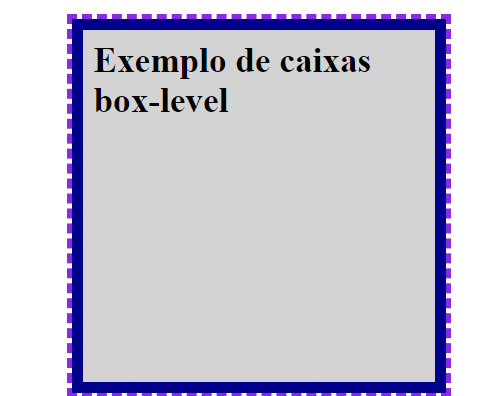
            margin-bottom: 40px;

            margin-left: 20px;

abaixo podemos ver o comando simplificado que dá o mesmo resultado do que da imagem acima:

            margin: 20px auto 20px auto;

Abaixo temos um exemplo de **box-level:**

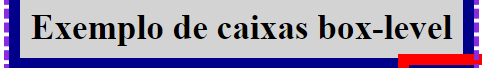


Abaixo temos um exemplo de **inline-level**:



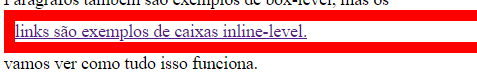
Podemos transforma um estilo **box-level** para um estilo **inline-level** usando a tag **display:inline;** porém como estamos mudando para inline-level não podemos regular altura (height) nem largura (width) **exemplo abaixo:**

            display: inline;



Podemos também transformas estilos **inline-level** para estilos **box-level** usando a tag **display:block**, **exemplo abaixo:**

            display: block;



Para adicionarmos um cabeçalho usamos a tag **header**.

Para adicionarmos um conteúdo principal usamos a tag **main**.

Para adicionarmos um roda pé usamos a tag **footer**.

Para adicionarmos um menu com link usamos a tag **nav**.

Para adicionarmos seções em nosso site usamos a tag **section**.

Para adicionarmos artigos usamos a tag **article**, para usarmos uma side, ou seja, um assunto relativo ao assunto do artigo usamos a tag **aside**.

Devemos organizar os estilos e seções do HTML em CSS, **exemplo abaixo:**

 <style>

        body{

            font-family: Arial, Helvetica, sans-serif;

            background-color: dimgray;

            margin: 0px;

        }

        header{

            background-color: white;

            padding: 10px;

            margin: 10px;

        }

        article{

            background-color: lightgray;

            padding: 5px;

        }

        nav{

            background-color:rgb(172, 161, 161);

            padding: 3px;

        }

        nav > a{

            text-decoration: none;

            color: white;

            font-weight: bold;

            margin-right: 5px;

        }

        nav > a:hover{

            text-decoration: underline;

            color: chartreuse;

        }

        }

        article > aside {

            background-color: rgb(156, 155, 155);

        }

        main{

            background-color: white;

            padding: 10px;

            margin: 10px;

        }

        footer{

            background-color: black;

            color: white;

            text-align: center;

            padding: 1px;

            margin: 0px;

        }

    </style>

**Conteúdo em CSS acima**

<body>

    <header>

        <h1>Meu site</h1>

        <nav>

            <a href="#">link</a>

            <a href="#">link</a>

            <a href="#">link</a>

            <a href="#">link</a>

            <a href="#">link</a>

        </nav>

    </header>

    <main>

        <section id="assuntos"></section>

        <p>Esportes Politica e Tecnologia</p>

        <section id="noticias">

            <article>

                <h2>Noticias sobre o Brasil</h2>

                <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Consequuntur itaque adipisci expedita totam consequatur perspiciatis sunt distinctio fuga? Animi temporibus at quam architecto unde necessitatibus sunt laboriosam nisi odit ab?</p>

                <aside>

                    <p>Artigo escrito por julio cesar baccin</p>

                </aside>

            </article>

        </section>

    </main>

    <footer>

        <p>Desenvolvido pelo julio cesar baccin</p>

    </footer>

</body>

**Conteúdo em HTML acima**

Podemos copiar estilos diretamente do **DevTools** para p **VSCcodem** basicamente precisamos somente copias os estilos configurados do **Devtools** para o **VScode** (dica de ouro), **exemplo abaixo:**

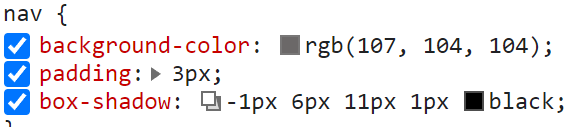
        nav{

            background-color:rgb(107, 104, 104);

            padding: 3px;

            box-shadow: -1px 6px 11px 1px black;

              }



O estilo da tag **box-shadow** foi configurada diretamente do **devtool**s e depois os valores somente foram copiados e colados no **VScode** adicionando assim a propriedade ao site.

Podemos fazer um círculo em CSS adicionando as seguintes configurações, **exemplo abaixo:**

   <div id="circulo"></div>

        #circulo{

          height: 100px;

          width: 100px;

          background-color: white;

          border-radius: 50%;

        }

Podemos criar uma borda personalizada em nosso site usando a tag **border-image-source: url**(‘**escolhemos o arquivo ou colocamos o nome da borda**’);

Podemos cortar a borda usando a tag **border-image-slice**.

Podemos repetir a borda usando a tag **border-image-repeat**, **exemplo abaixo:**

    <style>

        h1{

           border: 20px solid black;

           padding: 10px;

           border-image-source:url('borda.png');

           border-image-slice: 27;

           border-image-repeat: repeat

        }

    </style>

Podemos usar o site **Mockflow** para fazermos pré-visualizações de projetos de site.

Podemos dar várias denominações e pré modelos de cores e letras, onde podemos trocá-las facilmente usando a tag **:root,** **exemplo abaixo:**

@charset "UTF-8";

@import url('https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap');

@font-face {

    font-family:'Android' ;

    src: url("../fontes/idroid.otf") format('opentype');

    font-weight: normal;

}

/\*

#c5ebd5

#83elad

#3ddc84

#2fa866

#1a5c37

#063d1e

\*/

:root{

    --cor0:#c5ebd5;

    --cor1:#83e1ad;

    --cor2:#3ddc84;

    --cor3:#2fa866;

    --cor4:#1a5c37;

    --cor5:#063d1e;

    --fonte-padrao: Arial ,Verdana ,Helvetica ,sans-serif;

    --fonte-destaque: 'Bebas Neue', cursive;

    --fonte--android: 'android', cursive;

}

body{

    background-color:var(--cor0);

    font-family: var(--fonte-padrao);

}

header > h1{

    color:var(--cor5);

    font-family: var(--fonte-destaque);

}

main h1{

    color:var(--cor3);

    font-family: var(--fonte--android);

}

Podemos variar as corres e o tipo de fonte adicionando a tag **var** tanto em font quanto em color, então podemos escolher aquelas aplicadas na tag **:root,** assim podendo mudar facilmente as opções desejadas.

Podemos tirar as **margens** e **paddings** usando uma configuração global, sendo \*{}, **exemplo abaixo:**

\*{

    margin:0px;

    padding: 0px;

}

Podemos adaptar o conteúdo de nosso site para vários dispositivos, usando as tag **min-width** e **max-width**, também usamos a tag **margin:auto** para que as imagens e textos se adaptem melhor dentro do site e para várias tamanhos de tela de exibição, dessa maneira podemos também usar a **variável img**{ com a tag **width:100%** para adaptarmos as imagens dentro do site, melhorando assim a visualização e evitando que a imagem se quebre, **exemplo abaixo:**

<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>Teste de responsividade</title>

    <style>

        body{

            background-color: darkgray;

        }

        main{

            background-color: white;

            padding: 20px;

            border-radius: 10px;

            min-width: 320px;

            max-width: 800px;

            margin: auto;

        }

        img{

            width: 100%;

        }

    </style>

</head>

Podemos usar a extensão do google Chrome **window resizer** para ver o tamanho da nossa tela e simular tamanhos de telas diferentes.

Podemos usar a tag **Picture** junto com as tag **source media=’max-width: 800px)”** e **srcset=”localdaimagem/nomedaimagem.tipodaimagem** para mostrarmos versões diferentes da mesma imagem, claro que devemos adaptar o tamanho de exibição que queremos usando a tag **width** e também o tamanho da própria imagem, **exemplo abaixo**:

<picture>

            <source media="(max-width: 800px)" srcset="imagens/irina-blok-pq.jpg">

            <img src="imagens/irina-blok.jpg" alt="irina-blok">

        </picture>

No código acima estamos dizendo que até **800px** de largura queremos que seja exibida a imagem menor, pois se adapta melhor a dispositivos mobile, hoje onde a grande maioria dos acessos a sites se concentra, então temos uma versão da mesma imagem porem com uma largura maior que é apresentada quando a **viewport** ou seja a tela de exibição da pessoa que está acessando nosso site for maior que 800px ou seja largura, mostrará essa imagem maior.

Podemos centralizar imagens usando a tag **margin:auto**, também podemos colocar a **proporção em %** que queremos da imagem usando width:100%, **exemplo abaixo:**

main img.pequena{

    width: 50%;

    display: block;

    margin: auto;

}

Podemos adicionar um estilo gradiente onde quisermos apenas colocando o elemento que queremos e depois usando a tag **background-image: linear-gradient**(**então escolhemos a direção que iniciaremos, posteriormente colocamos a cor inicial do degrade e a final**, **exemplo abaixo:**

header {

    background-image: linear-gradient(to bottom, var(--cor3), var(--cor5));

Podemos colocar uma animação no menu de navegação **(nav)** usando a tag **transition-duration:** **exemplo abaixo:**

nav > a{

    color: white;

    padding: 10px;

    text-decoration: none;

    font-weight: bold;

    transition-duration: 0.5s;

}

nav > a:hover{

    background-color: var(--cor3);

    color: var(--cor5);

Podemos atribuir uma função especial aos parágrafos usando a tag **margin** e a **tag text-align,** sendo assim a mesma aplicada a todos os parágrafos, **exemplo abaixo:**

main p {

    margin: 15px 0px;

    text-align: justify;

}

O ideal quando formos dimensionar imagens dentro de nosso site e termos 2 tamanhos de imagens 1 para aparelhos menores e outra para aparelhos com uma **viewport** maior, o ideal e configuramos o **max-width** com a largura máxima que a imagem tem, assim não perdemos qualidade da mesma na hora da exibição.

Para colocarmos o recuo nos parágrafos usamos a tag **text-indent:** **exemplo abaixo:**

text-indent: 30px;

Para adicionarmos espaçamento entre as linhas de texto usamos a tag line-heigth: **exemplo abaixo:**

line-height: 2em;

Podemos colocar um gradiente transparente usando a tag **transparent**, **exemplo abaixo:**

  background-image: linear-gradient(to right, var(--cor1),transparent);

Podemos organizar uma lista dentro do conteúdo da side usando a tag **list-style-position:** então escolhemos a opção que queremos, **exemplo abaixo:**

aside >ul {

    list-style-position: inside;

}

Lembrando somente usamos > na declaração para colocarmos efeito que estão dentro de outras declaraçãos, **exemplo:**

**Aside >** a – isso significa que todos os links dentro de aside terão as propriedades que forem atribuídas a declaração.

**Main** a – isso significa que todo link dentro do conteúdo principal terá as propriedades que forem atribuídas na declaração

Podemos dividir nossa lista usando a tag **colums:** **exemplo abaixo:**

aside>ul {

    list-style-position: inside;

    columns: 2;

}

Podemos adicionar símbolos e emolis em uma lista usando a tag **list-style-type:** então colocamos o próprio emoji ou símbolo ou então pegamos somente a numeração antes de u+ e colamos dentro das aspas “ “ sempre usando barra invertida \ **exemplo abaixo:**

aside>ul {

    list-style-position: inside;

    columns: 2;

    list-style-type: "\2714"

}

Podemos adicionar espaços entre o texto e os emojis usando a tag **00A0**, sendo que cada declaração dessas significa um espaço que deve ser separado por uma barra invertida, **exemplo abaixo:**

list-style-type: "\✔️\00A0\00A0";

Para colocarmos ícones ou emojis em links externos usamos a tag com uma subclasse, **exemplo abaixo**:

}

a.externo::after{

    content: "\🔗";

}

<p>A ideia de ter um mascote foi amadurecendo e a missão foi passada para uma profissional da área. A

                ilustradora Russa <a href="https://www.irinablok.com/" target="\_blank" class="externo">Irina Blok</a>, também

                funcionária do Google, ficou com a missão de representar o pequeno robô de uma maneira mais agradável.

            </p>

Em HTML coloco class e em CSS usamos o. (ponto)

Em HTML coloco ID e em CSS usaremos #

Para dimensionarmos o tamanho de um vídeo usamos os **comandos abaixo:**

div.video{

    background-color: var(--cor5);

    margin: 0px -20px 30px -20px;

    padding: 20px;

    padding-bottom: 59%;

    position: relative;

}

div.video > iframe {

    position: absolute;

    top: 5%;

    left: 5%;

    width: 90%;

    height: 90%;

}

Além das declarações em CSS acima envelopamos o link da incorporação, ou seja, o iframe com a tag **div** e colocamos o elemento class, **exemplo abaixo:**

    <div class="video">

                <iframe width="560" height="315" src="https://www.youtube.com/embed/62JB64aQC9A"

                    title="YouTube video player" frameborder="0"

                    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"

                    allowfullscreen></iframe>

            </div>

Quando tivermos vários elementos, é melhor classificarmos com a classe **class**, exemplo: vídeos, botões, parágrafos.

Quando tiver elementos específicos e melhor classificarmos com a classe **ID**, exemplo: meus botões, meu vídeo e afins.

**GIT/GITHUB**

Podemos subir sites após fazermos alterações no arquivo base abrindo o mesmo pelo **GitHub Desktop** e selecionarmos o repositório em que está o arquivo e clicarmos em Open in **Visual Studio Code**, após as alterações o programa irá detectar que nosso arquivo base sofreu alterações, então clicamos em **commit** e depois em **push** para que as alterações sejam lançadas em nosso arquivo no servidor remoto.

**Planos de fundo**

Abaixo vemos 3 declarações de como mudar a imagem de fundo, observamos que primeiramente damos uma classe a eles com o nome quadrado e depois os damos um **ID** onde colocamos cores dos planos de fundos distintos, sendo que usando a classe quadrado escolhemos primeiramente as dimensões dos nossos elementos e com os **ID** as propriedades internas dos elementos, **exemplos abaixo:**

 <style>

        div.quadrado {

            border: 2px solid black;

            width: 300px;

            height: 300px;

            display: inline-block;

            background-color: lightgray;

            border-radius: 10px;

        }

        div#q1 {

            background-color: skyblue;

        }

        div#q2 {

            background-image: linear-gradient(to bottom, yellow, red);

        }

        div#q3 {

            background-image: url(imagens/pattern003.png)

        }

    </style>

</head>

<body>

    <div class="quadrado" id="q1">

    </div>

    <div class="quadrado" id="q2">

    </div>

    <div class="quadrado" id="q3">

    </div>

</body>

Podemos colocar imagens diretamente do google, simplesmente clicando com o botão direito e selecionando o comando copia endereço da imagem, então colocamos na tag **background-image: url();** **exemplo abaixo:**

    <style>

        body{

            background-image:url("https://static.wikia.nocookie.net/naruto/images/8/8d/Nice\_Guy.PNG/revision/latest?cb=20151127190814&path-prefix=pt-br");

            background-size: 500px 500px;

            background-repeat: repeat;

        }

    </style>

Podemos escolher quais ângulos da imagem vamos colocar em nosso background usando a tag **background-position**, **exemplo abaixo:**

<style>

        div.bloco{

            background-image: url("imagens/wallpaper003.jpg");

            height: 200px;

            border: 1px solid black;

            border-radius: 20px;

            margin: 10px;

        }

        div#q1{

            background-position: left top;

        }

        div#q2{

            background-position: left center;

        }

        div#q3{

            background-position: right bottom;

        }

        div#q4{

            background-position: center center;

        }

    </style>

</head>

<body>

    <div class="bloco" id="q1">

    </div>

    <div class="bloco" id="q2">

    </div>

    <div class="bloco" id="q3">

    </div>

    <div class="bloco" id="q4">

    </div>

</body>

Podemos fazer o posicionamento de uma imagem usando a tag height e dimensionando esse posicionamento com uma terminação em VH (VIEW HEIGHT) sendo **height:100vh** o centro do site, **exemplo abaixo:**

<style>

    body{

        background-image:url("https://static.wikia.nocookie.net/naruto/images/8/8d/Nice\_Guy.PNG/revision/latest?cb=20151127190814&path-prefix=pt-br");

        background-size: 100px;

        background-repeat:no-repeat;

        background-position: center center;

        border: 1px dashed black;

        height:98vh;

    }

</style>

Podemos dimensionar o tamanho de uma imagem de fundo usando a tag **background-position: cover;** essa tag faz com que a imagem de fundo seja picotada e para que tenha total exibição em qualquer tamanho de view port ou telam **exemplo abaixo:**

<style>

        body{

            background-color:black;

            background-image:url(imagens/wallpaper002.jpg);

            background-position: center top;

            background-repeat: no-repeat;

            height: 100vh;

**background-size: cover;**

        }

        main{

            background-color:white;

            width: 400px;

            padding: 10px;

            margin:auto;

            border: 1px solid darkblue;

            border: radius 15px;

        }

    </style>

Também podemos usar as tag **background-size:auto** e **background-size: contain**, porém em algumas viewports a tela de exibição pode apresentar bordas laterais.

Podemos usar **shorthands** para backgrounds, usando a tag **background** inicialmente e depois inserindo a seguinte sequência:

**Color > image > repeat > [size] > attachment**

Exemplo da shorthand abaixo:

background: black url(imagens/wallpaper002.jpg) center center no-repeat fixed;

A tag size dentro da shorthand não funciona, assim temos que colocá-lo separado, **exemplo abaixo:**

    body{

             height: 100vh;

            background: black url(imagens/wallpaper002.jpg) center center no-repeat  fixed;

            background-size:cover;

        }

**Aprenda CSS Position em 10 Minutos - Tutorial CSS para Iniciantes**

Usando a tag **position:relative** habilita para que consigamos trabalhar a posição dos conteúdos da maneira que quisermos, usando as tag.

* Top: 10px ou 100%
* Bottom: 10px ou 100%
* Rigth: 10px ou 100%
* Left: 10px ou 100%

Usando a tag **position:absolute** faz com que a forma tenha seu próprio fluxo de posicionamento absoluto, como se ignorasse todo conteúdo existente no documento, desde que essa forma não esteja dentro de uma outra forma com a tag **position:relative, Exemplo abaixo:**

<title>Alinhamento Vertical</title>

    <style>

        #container{

            background-size: 100% 100%;

            background-image:url(imagens/target001.png);

            height: 500px;

            background-color:purple;

            padding:10px;

        }

        #conteudo{

            background-size: 100% 100%;

            background-image:url(imagens/target001.png);

            position: absolute;

            height: 200px;

            width: 400px;

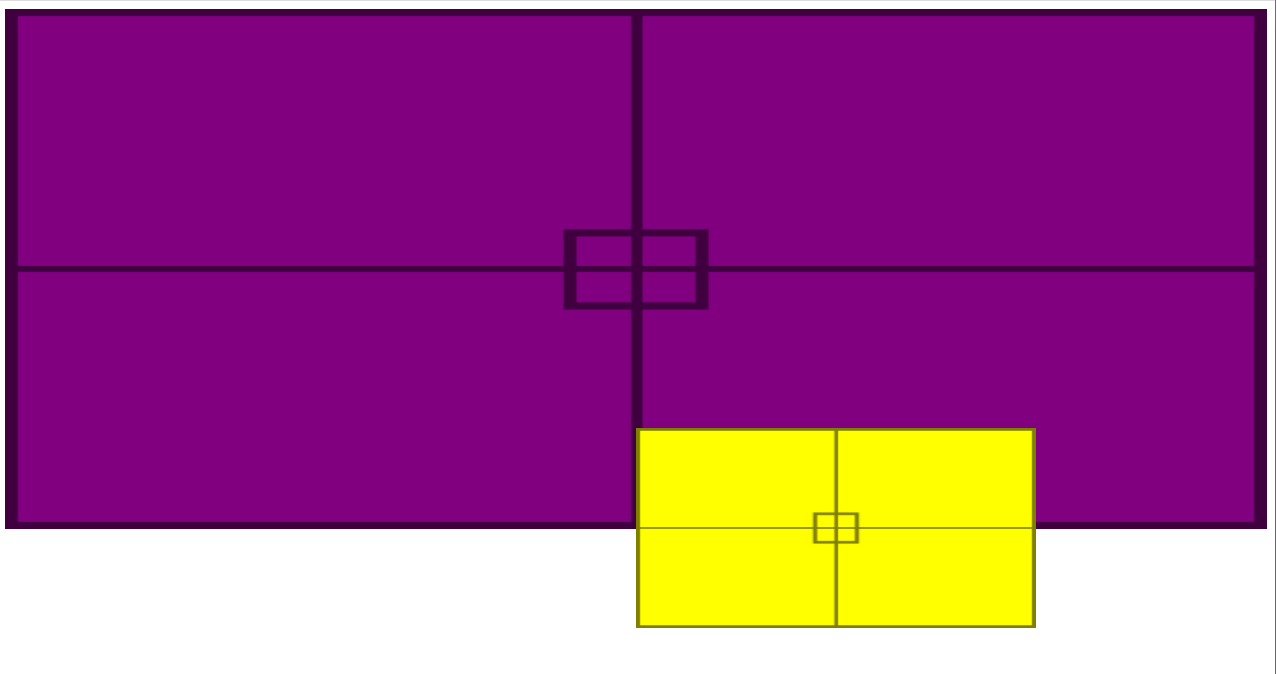
            background-color:yellow;

            left:50%;

            top:50%;

        }

    </style>

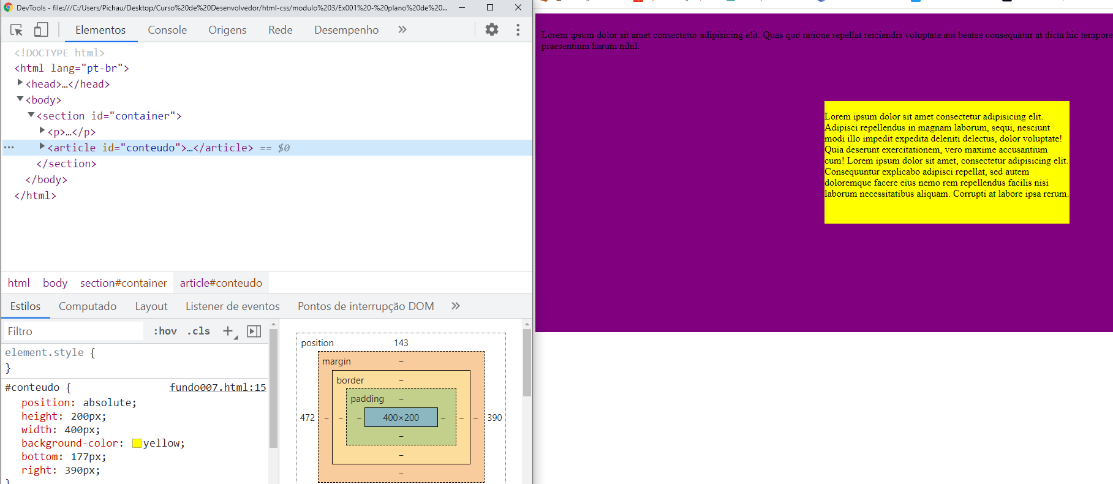


Usando a tag **position:fixed** deixa o elemento fixo na tela, não se importando com as propriedades de textos ou afins que estão abaixo dele, ideal para fazer botões ou header para navegação.

* Position:fixed
* Top:10px

Usando a tag **position:sticky** deixa o elemento grudado na tela ao scrolar a tela, geralmente é uma tag usada a elaboração de menus de navegação.

Podemos simular o local onde queremos dos elementos usando a ferramenta inspecionar código fonte do devtool, então precisamos somente alterarmos as propriedades conforme o posicionamento que desejamos, **exemplo abaixo:**



Podemos fazer também o posicionamento de elementos usando a tag **transform:translate** e colocando a quantidade que queremos de movimentação, **exemplo abaixo:**

transform:translate(-50%, -50%);

Usando **<a> href: target=’blank’** – fará com que os links abram, porém um vai sobressair o clic do outros, no caso se eu tiver 2 links e clicar em 1 quando o site está carregando após eu clicar no outro este outro irá carregar no lugar do 1 tomando seu lugar na mesma página e janela.

**Usando <a> href: target=’\_blank’** – fará que qualquer link que seja clicado abra uma nova página em uma nova janela.

Usando a tag **text-transform: uppercase** podemos deixar todas as letras em maiúsculo, **exemplo abaixo:**

header > h1{

    padding-top: 50px;

    text-transform: uppercase;

}

Usando a tag **font-variant: small-caps** deixamos somente as letras iniciais maiúsculas, **exemplo abaixo:**

header > h1{

    padding-top: 50px;

    font-variant: small-caps;

}

Podemos adaptar o tamanho da fonte as diversas viewports colocando a propriedade de VW (viewport) dentro da tag **font-size: 10vw** (10% da viewport), com isso conseguimos fazer com que os textos se adaptem aos tamanhos da tela diminuindo e aumenta conforme o tamanho da tela, **exemplo abaixo:**

section {

    padding-top: 10vh;

    padding-bottom: 10vh;

    line-height: 2em;

    padding-left: 30px;

    font-family: var(--fonte3);

    font-size: 3.5vw;

}

header > h1 {

    padding-top: 50px;

    font-variant: small-caps;

    font-family:var(--fonte2);

    font-size: 10vw;

}

Para colocarmos fontes da internet, podemos copiar o **embed** do site **google fonts** e depois colarmos no projeto, após colarmos os links dentro da pasta de CSS usamos a tag global **:root** para adicionarmos variantes de fontes para adicionarmos em nosso projeto, **exemplo abaixo:**

@charset "UTF-8";

@import url('https://fonts.googleapis.com/css2?family=Passion+One&display=swap');

@import url('https://fonts.googleapis.com/css2?family=Sriracha&display=swap');

:root{

    --fonte1: Verdana, "Geneva", Tahoma, sans-serif;

    --fonte2: 'Passion One', "cursive";

    --fonte3: 'Sriracha', "cursive";

    }

Após isso basta escolher uma das fontes usando a tag **fonte-family:var(crtl+espaço); exemplo abaixo:**

html, body {

    background-color: darkgray;

    min-height: 100vh;

    font-family: var(--fonte1);

}

Para trabalharmos imagens dentro de textos, primeiramente tratamos dos estilos, larguras, fontes e afins das formas, depois inserimos as imagens, **exemplo abaixo:**

    <section class="imagem" id="img01">

        <p>

            Não é mais como era antes<br>

            É tudo abreviado<br>

            "Você" só tem duas letras<br>

            O "O" e o "E" foi riscado<br>

            Para declarar o amor<br>

            Basta botar uma flor<br>

            E um coração desenhado.

        </p>

    </section>

section.imagem {

    background-color: rgb(56, 56, 56);

    color: white;

    box-shadow: inset 6px 6px 13px 0px rgba(0, 0, 0, 0.473);

    background-size:cover;

    background-attachment: fixed;

}

section.imagem > p {

    display: inline-block;

    background-color: rgba(0, 0, 0, 0.514);

    padding: 5px;

    text-shadow: 4px 4px 0 black;

}

section#img01{

    background-image: url(imagens/background001.jpg);

    background-position: right center;

}

section#img02{

    background-image: url(imagens/background002.jpg);

}

Usamos a **classe imagem** para tratarmos da:

Cor de fundo = **background-color**

Cor dos parágrafos = **color**

Sombra da forma = **box-shadow**

Tamanho da forma = **background-size**

Tipo de ligação = **background-attachment**

Depois usamos a **ID”img01”** para adicionarmos a imagem e seu posicionamento de foco:

**Background-image: url (pasta da imagem crtl+espaço)**

**Background-position: right center;**

Para criamos uma tabela em html usamos a tag **table** (tabela) depois inseremos a tag **tr** (linha) e depois a tag **td** (dados da tabela), podemos personalizar a tabela conforme queremos, inserindo **classes** e **ID’s** para personalizações mais individuais, **exemplo abaixo:**

 <style>

        body{

            font-family: Arial, Helvetica, sans-serif;

        }

        table{

           border-collapse:collapse;

           width: 400px;

           height: 400px;

           border: 2px solid red;

           font-size: 5vh;

        }

        td{

            text-align: center;

            padding: 10px;

            font-weight: bolder;

            border: 1px solid black;

        }

        tr.linha1{

            background-color: darkgrey;

        }

        tr.linha2{

            background-color: blue;

        }

        td#amarelo{

            background-color: yellow;

        }

        td#vermelho{

            background-color: red;

        }

        h1{

            font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

            background-color: aqua;

            display: inline-block;

            padding-bottom: 30px;

            width:400px;

        }

    </style>

</head>

<body>

    <h1>Minha primeira tabela</h1>

    <!-- HIERARQUIA DE TABELAS (simples)

        TABLE = tabela

            TABLE ROW = linha de tabela

                TABLE HEADER = cabeçalho de tabela

                TABLE DATA = dados da tabela-->

    <table>

        <tr class="linha1"><!-- primeira linha-->

            <td>A1

            <td>B1

            <td>C1

        </tr>

        <tr class="linha2"><!-- segunda linha-->

            <td>A2

            <td>B2

            <td>C2

        </tr>

        <tr>

            <td>A3</td><!-- terceira linha-->

            <td>B3</td>

            <td>B3</td>

        </tr>

        <tr>

            <td id="vermelho">A4</td><!-- quarta linha-->

            <td id="amarelo">B4</td>

            <td>C4</td>

        </tr>

    </table>

Podemos inserir uma tabela com cabeçalho, corpo e um rodapé para nossa tabela usando:

**Thead** = cabeçalho

**Tbody =** corpo

**Tfoot** = rodapé

Iniciamos com a tag **table** dentro de **body** depois inserimos a tag **thead** para o cabeçalho, dentro de **thead** inserimos a tag **tr** para as linhas e dentro de **tr** inserimos **th** para colocamos os dados do cabeçalho, **exemplo abaixo:**

<body>

    <h1>Uso de tabelas grandes</h1>

    <!--ANATOMINA PARA TABELAS GRANDES

        TABLE

            THEAD

                TR, TD, TH

            TBODY

                TD, TD, TH

            TFOOT

                TR, TD, TH

    -->

    <table>

        <thead>

            <tr>

                <th class="p1">Posição</th>

                <th class="p1">Estado</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td>São Paulo</td>

                <td class="num">45.919.049</td>

            </tr>

            <tr>

                <td>Minas Gerais</td>

                <td class="num">21.169.791</td>

            </tr>

            <tr>

                <td>Rio de Janeiro</td>

                <td class="num">17.264.943</td>

            </tr>

        </tbody>

        <tfoot>

            <tr>

                <th class="p1">Total de habitantes</th>

                <td class="num">84.353.783</td>

            </tr>

        </tfoot>

    </table>

</body>

Podemos inserir uma legenda nas tabelas usando a tag **caption**, **exemplo abaixo:**

<table>

        <caption>População das unidades federativas</caption>

        <thead>

            <tr>

                <th class="p1">Posição</th>

                <th class="p1">Estado</th>

            </tr>

        </thead>

Usamos **th** sempre que queremos colocar um título dentro da tabela, e usamos **td** sempre que queremos colocar um dado.

Para dizermos ao HTML que um título pertence a uma coluna atribuímos ao **thead (th)** a classe **scope=”col”**

Para dizermos ao HTML que um título pertence a uma linha usamos atribuímos ao **thead (th)** a classe **scope=”row” , exemplo abaixo:**

<table>

        <caption>População das unidades federativas</caption>

        <thead>

            <tr>

                <th class="p1" scope="col">Posição</th>

                <th class="p1" scope="col">Estado</th>

            </tr>

        </thead>

<tfoot>

            <tr>

                <th class="p1" scope="row">Total de habitantes</th>

                <td class="num">84.353.783</td>

            </tr>

        </tfoot>

Podemos aplicar um efeito **zebrado** a tabela aplicando propriedades as linhas **(tr)** que estão dentro de **tbody** (corpo), **exemplo abaixo:**

        tbody > tr:nth-child(odd) {

            background-color:lightgrey;

            font-weight: bold;

        }

        tbody > tr:nth-child(even) {

            background-color:white;

            text-decoration: underline;

        }

Para deixarmos o cabeçalho da tabela fixo enquanto rolamos a tabelas podemos usar a tag **position: relative** dentro da tabela **(table),** e posteriormente criamos uma declaração exclusiva para o thead **(th)** e trow **(tr),** inserindo a tag **position: sticky,** **exemplo abaixo:**

        table{

            width: 500px;

            border-collapse: collapse;

            text-align: left;

            font-family: cursive;

            position: relative;

        }

        thead > tr > th {

            position: sticky;

            top: 0;

            background-color: rgba(80, 77, 77, 0.966);

        }

Podemos usar a tag **colspan** (horizontal) para mesclarmos células dentro de uma coluna da tabela e usamos a tag **rowspan** (na vertical)para mesclarmos células dentro de uma linha da planilha, em ambas tag temos que dizer a quantidade de células que queremos que a declaração venha a ocupar, **exemplo abaixo**:

  <tr>

            <td>D</td>

            <td>E</td>

            <td rowspan="3">F</td>

        </tr>

        <tr>

            <td>A</td>

            <td colspan="2">B</td>

        </tr>

Podemos pintar as colunas de maneira individual usando a tag **colgroup** então dentro inserimos a tag **col** e lhe atribuímos uma **class,** lembrando que as ordens das colunas começam em ordem decrescente (de cima pra baixo 1, 2, 3 e afins....) **exemplo abaixo:**

        <colgroup>

            <col class="nome">

            <col class="sexo">

            <col class="idade">

            <col class="profissao">

        </colgroup>

        <caption>Cadastro</caption>

        <tr>

            <th scope="col">nome</th>

            <th scope="col">sexo</th>

            <th scope="col">idade</th>

            <th scope="col">profissão</th>

        </tr>

        }

        col.nome{

            background-color: chartreuse;

        }

        col.sexo{

            background-color: blue;

        }

        col.idade{

            background-color: burlywood;

        }

        col.profissao{

            background-color: darkgreen;

        }

Além disso, podemos usar a tag **span** e colocarmos quantas colunas queremos que tenham a mesma propriedade da atual, **exemplo abaixo:**

<col class="idade" span="2">

Conforme a **viewport** de exibição em que a tabela está sendo exibida vai diminuindo, e criado uma barra de rolagem na mesma onde acaba rolando não somente o conteúdo da tabela, mas como o conteúdo do texto também, para que a tabela se adapte a **viewport** podemos usar a tag **overflow**, lembrando que devemos selecionar toda nossa tabela e atribuirmos uma div a ela, então atribuímos uma **class** na qual vamos aplicar a tag **overlfow**, **exemplo abaixo:**

 <div id="container">

        <table>

            <caption>Tabelas Responsivas</caption>

            <thead>

                <tr>

                    <th>Produto</th>

 }

        div#container{

            overflow: auto;

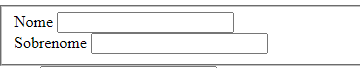
        }

    </style>

**Formulário de login - responsivo**

Usamos sempre a tag **form** para iniciarmos um formulário de cadastro, será dentro desta tag que vamos inserir todos os comandos para a criação do formulário.

Usamos a tag **fieldset** para criamos uma borda ao redor do dos formulários de preenchimento, **exemplo abaixo:**



Para criarmos um formulário de cadastro usamos a tag **label** para exibir a informação que o usuário vai ler então usamos a tag **input** onde configuramos o campo de preenchimento do cadastro, o tipo de informação e o comando de recebimento de texto, **exemplo abaixo:**

<body>

    <div>

        <h1>Cadastro de DEVs</h1>

        <p>Complete suas informações</p>

        <br>

    </div>

    <form>

        <fieldset>

            <div>

                <label>Nome</label>

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

            </div>

            <div>

                <label>Sobrenome</label>

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

            </div>

        </fieldset>

    </form>

Usamos a tag **input=”text”** para definirmos o tipo do campo de preenchimento podendo ser comprido **‘email’** ou redondo **‘radio’.**

Usamos atag **label** para definirmos o nome que queremos que apareça antes do campo de formulário de cadastro.

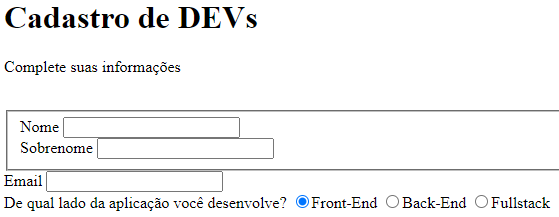
Usamos a propriedade **Checked** se já quisermos deixar um campo do **type=’radio’** selecionado ao abrirmos a pagina do formulário, **exemplo abaixo:**

            <label>De qual lado da aplicação vocêdesenvolve?</label>

            <label>

                <input type="radio" name="devweb" value="frontend" checked>Front-End

            </label>



Usando a tag **select id”nome”** podemos atribuir um aba de opção de seleção para o usuário, inserindo a tag **option** podemos listar as opções de escolhas da aba de seleção, usando a declaração **<option selected disable value=” “>nome</option**> podemos deixar uma mensagem inicial de seleção dentro da aba, **exemplo abaixo:**

        <div>

            <label>Senioridade</label>

            <select id="Senioridade">

                <option selected disabled value="">Selecione</option>

                <option value="Junior">Junior</option>

                <option value="Pleno">Pleno</option>

                <option value="Sênior">Sênior</option>

            </select>

        </div>



Podemos usar a tag **input** com a propriedade **checkbox** para colocarmos caixas de seleção para os usuários, além disso precisamos dar uma **ID** e um **NAME** para a **checkbox**, posteriormente inserimos um **label ‘for’** com o nome do **ID** e **NAME** que definimos inicialmente na tag **input**, estamos usando duas tags **<br>** na linha do **label** para adicionarmos **2 linhas de espaço** entre o texto e o campo de seleção, **exemplo abaixo:**

        <fieldset>

            <div>

                <label for="">Selecione as tecnologias que utiliza</label><br><br>

                <input type="checkbox" id="tecnologia1" name="tecnologia1" value="HTML" checked>

                <label for="tecnologia1">HTML</label>

                <input type="checkbox" id="tecnologia2" name="tecnologia2" value="CSS">

                <label for="tecnologia2">CSS</label>

                <input type="checkbox" id="tecnologia3" name="tecnologia3" value="JavaScript">

                <label for="tecnologia3">JavaScript</label>

                <input type="checkbox" id="tecnologia4" name="tecnologia4" value="PHP">

                <label for="tenologia4">PHP</label>

                <input type="checkbox" id="tenologia5" name="tecnologia5" value="C#">

                <label for="tecnologia5">C#</label>

                <input type="checkbox" id="tecnologia6" name="tecnologia6" value="Python">

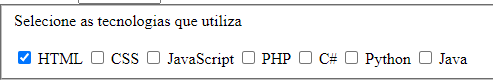
                <label for="tecnologia6">Python</label>

                <input type="checkbox" id="tecnologia7" name="tecnologia7" value="Java">

                <label for="tecnologia7">Java</label>

            </div>

        </fieldset>



Podemos inserir um campo onde o cliente pode inserir comentários e textos sobre suas experiencias usando a tag **textarea**, onde vamos configurar o **name**=” “ **id**=’ ‘ **cols=’**largura do campo’  **rows=’**quantia de linhas’ além disso podemos inserir um botão de envio das informações usando a tag **button** e selecionando a opção **type=’submit’**, **exemplo abaixo:**

        <div>

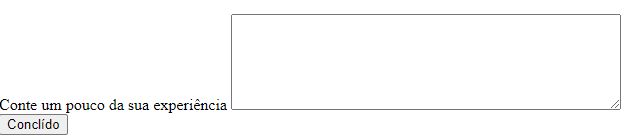
            <br>

            <label for="">Conte um pouco da sua experiência</label>

            <textarea name="experiência" id="experiência" cols="50" rows="6"></textarea>

        </div>

<button type="submit">Concluído</button>



Usando a tag **required** no **final** da tag **input**, fazemos com que as informações do usuário não posam ser enviadas até que aquele campo seja preenchido, **exemplo abaixo:**

        <fieldset class="grupo">

            <div>

                <label for="nome">Nome</label>

                <input type="text" name="nome" id="nome" required>

            </div>

            <div class="campo">

                <label for="sobrenome">Sobrenome</label>

                <input type="text" name="sobrenome" id="sobrenome" required>

            </div>

        </fieldset>



**TELA DE LOGIN HTML/CSS – Responsiva**

Usando a tag **placeholder** dentro da tag **input**, fazemos com que seja exibido um nome dentro de um campo para preenchimento, **exemplo abaixo:**

<body>

    <main class="container">

        <h2>login</h2>

        <form action="">

            <div class="input-field">

                <input type="text" name="username" id="username" placeholder="Enter Your Username">

                <div class="underline"></div>

            </div>

            <div class="input-field">

                <input type="password" name="password" id="password" placeholder="Enter Your Password">

                <div class="underline"></div>

            </div>

            <input type="submit" value="Continue">

        </form>

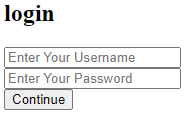
</main>

</body>

****

Podemos inserir um botão usando a declaração **<input type=”submit” value=”Continue”>** a propriedade **value** vai receber o nome que digitarmos entre as aspas, e a propriedade **type** vai apresentar a forma que escolhermos, **exemplo abaixo:**

<input type="submit" value="Continue">



Usamos a tag **I** para destacar alguma parte do texto que é importante por algum motivo, podendo ser expressões, icones na declaração abaixo ela está apresentando o **ícone do twitter**:

        <div class="footer">

            <span>Or connect With Social Media</span>

            <div class="social-fields">

                <div class="social-field twitter">

                    <a href="#">

                        <i class="fab fa-twitter"></i>

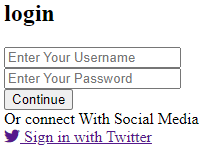
                        Sign in with Twitter

                    </a>

                </div>

            </div>

        </div>



Podemos usar as seguintes configurações de **responsividade em CSS** para deixarmos nosso conteúdo centralizado, **exemplo abaixo:**

body{

    background:linear-gradient(45deg, #8e2de2, #4a00e0);

    background-repeat: no-repeat;

    min-height: 100vh;

    min-width: 100vw;

    display: flex;

    align-items: center;

    justify-content: center;

}

A tag **align-items** tem **3 propriedades** primordiais, sendo:

**Align-items: center** – vai alinhar os elementos ao centro do elemento pai

**Align-items: flex-start** – vai alinhar os elementos ao início do elemento pai

**Align-items: flex-end** – vai alinhar os elementos na parte de baixo do elemento pai

Lembrando que o elemento deve ter a propriedade **display:flex;** todo esse alinhamento será feito no sentido **vertical,** ou seja, de cima para baixo.

A tag **justify-content** tem **3 propriedades** primordiais, sendo:

**Justify-content: center** – vai alinhar os elementos ao centro do elemento pai

**Justify-content: flex-start** – vai alinhar os elementos ao início do elemento pai

**Justify-content: flex-end** – vai alinhar os elementos na parte de baixo do elemento pai

Lembrando que o elemento deve ter a propriedade **display:flex;** todo esse alinhamento será feito no sentido **horizontal,** ou seja, da esquerda para direita.

Para usar ícones em **HTML** usamos o site **fontawsome**, assim copiamos e colamos o **<script>** do nosso usuário para que possamos poder usar todos os ícones do site, **exemplo abaixo:**

<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">

    <script src="https://kit.fontawesome.com/fad7c832c4.js" crossorigin="anonymous"></script>

    <title>Tela de login</title>

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

</head>

Podemos usar **pseudo classe** :: para atribuirmos posicionamento e formas ao elementos, no exemplo abaixo usamos a **pseudo classe** **:: before** para colocarmos uma linha abaixo do subtítulo **h2** (login) que está dentro de um elemento pai **(main):**

main h2::before{

    content: "";

    position: absolute;

    height: 4px;

    width: 25px;

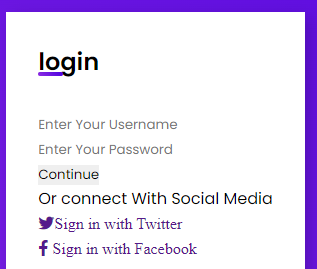
    bottom: 3px;

    left: 0;

    border-radius: 8px;

    background:linear-gradient(45deg, #8e2de2, #4a00e0);

}



Para trabalharmos o estilho de **pseudo classe**, colocamos a declaração iniciando pela **classe atribuída inicialmente > elemento primário** e ai sim colocamos a **pseudo classe**, abaixo estamos mudando a cor do placeholder, **exemplo abaixo:**

            <div class="input-field">

                <input type="text" name="username" id="username" placeholder="Enter Your Username">

                <div class="underline"></div>

            </div>

.input-field > input::placeholder{

    color: rgba(0,0,0,0.5)

}



Para trabalharmos o estilo dos botões com a tag **input,** colocamos inicialmente o **elemento mãe** seguida do **elemento secundário** com sua caracterização (type), abaixo estamos estilizando o botão de login, **exemplo abaixo:**

        <form action="">

            <input type="submit" value="Continue">

        </form>

form > input[type="submit"]{

    margin-top: 2rem;

    padding: 0.4rem;

    background:linear-gradient(to left, #4776e6, #8e54e9);

    cursor: pointer;

    color: white;

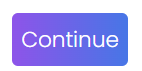
    font-size: 0.9rem;

    font-weight: 300;

    border-radius: 4px;

    transition: all 0.3s ease;

}



Para fazermos com que o botão tome todo o comprimento do **elemento pai**, precisamos atribuir ao seu elemento pai a tag **display:flex;** e **flex-direction:column**; fazendo assim com que os elementos filhos dentro deste elemento pai sejam posicionados em colunas, **exemplo abaixo:**

        <form action="">

            <input type="submit" value="Continue">

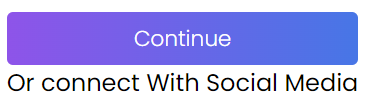
        </form>

form{

    display: flex;

    flex-direction: column;

}



Podemos fazer animações de cores em botões colocando inicialmente o **elemento pai** seguido do **elemento filho** com seu **type** e usamos a **pseudo calsse :hover**, com isso estamos aplicando a cor do gradiente da esquerda para a direita, **exemplo abaixo:**

        <form action="">

            <input type="submit" value="Continue">

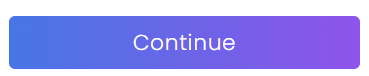
        </form>

form > input[type="submit"]:hover{

    letter-spacing: 0.5px;

    background:linear-gradient(to right, #4776e6, #8e54e9);

}



Abaixo estamos aplicando um estilo em CSS, estamos dizendo na declaração que queremos aplicar o estilo no elemento filho (span) que está dentro do elemento pai, (footer), **exemplo abaixo:**

<div class="footer">

            <span>Or connect With Social Media</span>

</div>

.footer > span{

    color: rgba(0, 0, 0, 0.7);

    font-size: 0.8rem;

    text-align: center;

}

****

Abaixo estamos aplicando um estilo em CSS no elemento filho com a **div classe=”social-field”,** onde o mesma está dentro da classe pai **div classe=”footer”, exemplo abaixo:**

        <div class="footer">

            <span>Or connect With Social Media</span>

        <div class="social-field">

                <div class="social-field twitter">

                    <a href="#">

                        <i class="fab fa-twitter">Sign in with Twitter</i>

                    </a>

                </div>

                <div class="social-field facebook">

                    <a href="#">

                        <i class="fab fa-facebook-f"> Sign in with Facebook</i>

                    </a>

                </div>

            </div>

        </div>

.footer > .social-field{

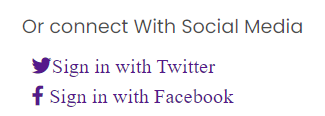
    padding: 0.4rem;

    border-radius: 4px;

    font-size: 0.85rem;

    width: 100%;

}



Abaixo estamos estilizando em CSS os elementos **twitter** e **facebook** usando a seguinte declaração. **footer** > .**social-fields** > .**twitte**r e .**foote**r > .**social-fields** > .**facebook**, lembrando que estamos aplicando estilos de maneiras individual, inicialmente fazemos a declaração das classes pais até chegar classes filhos, **exemplo abaixo:**

<div class="footer">

            <div class="social-field">

                <div class="social-field twitter">

                    <a href="#">

                        <i class="fab fa-twitter">Sign in with Twitter</i>

                    </a>

                </div>

                <div class="social-field facebook">

                    <a href="#">

                        <i class="fab fa-facebook-f"> Sign in with Facebook</i>

                    </a>

                </div>

            </div>

</div>

.footer > .social-field > .twitter{

    background: linear-gradient(to right, #56a7f2, #468aca);

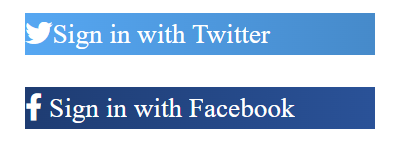
    margin-bottom: 1rem;

}

.footer > .social-field > .facebook{

    background:linear-gradient(to right, #1e3c72, #2a5298);

}



Quando quisermos dar um espaçamento em um único elemento devemos fazer a declaração de modo a colocar primeiramente o **elemento pai** seguida **primeira classe** junto com o **pseudo elemento** **:first-child, exemplo abaixo:**

<form action="">

            <div class="input-field">

                <input type="text" name="username" id="username" placeholder="Enter Your Username" required>

                <div class="underline"></div>

            </div>

form .input-field:first-child {

    margin-bottom: 1.5rem;

}



Um **pseudo-elemento** CSS é uma palavra-chave adicionada a um seletor que permite que você estilize uma parte específica do elemento selecionado. Por exemplo, o pseudo-elemento [::first-line](https://developer.mozilla.org/pt-BR/docs/Web/CSS/::first-line) aplica o estilo apenas na primeira linha de um parágrafo.

Uma ***pseudo-classe*** CSS é uma palavra-chave adicionada a seletores que especifica um estado especial do elemento selecionado. Por exemplo, [:hover](https://developer.mozilla.org/pt-BR/docs/Web/CSS/:hover) pode ser usado para alterar a cor de um botão quando o usuário passar o cursor sobre ele.

Podemos adicionar uma linha abaixo do campo de preenchimento **input** atribuindo uma **class=”underline”** onde depois vamos atribuir as configurações em CSS, abaixo vemos as declarações seguem o raciocínio de colocar a **classe** do **elemento pai** seguida da **classe** do **elemento filho**, e então aplicar as configurações de CSS para estilização, abaixo vemos como fazer a inserção de uma linha com background que faz uma animação de quando clicamos dentro do campo de preenchimento do input, partindo do lado esquerdo usando a tag **transform-origin:left,** **exemplo abaixo:**

 <div class="input-field">

                <input type="text" name="username" id="username" placeholder="Enter Your Username" required>

                <div class="underline"></div>

 </div>

.input-field .underline::before {

    content: '';

    position: absolute;

    height: 1px;

    width: 100%;

    bottom: -5px;

    left: 0;

    background: rgba(0, 0, 0, 0.2);

}

.input-field .underline::after {

    content: '';

    position: absolute;

    height: 1px;

    width: 100%;

    bottom: -5px;

    left: 0;

    background: linear-gradient(45deg, #8e2de2, #4a00e0);

    transform: scaleX(0);

    transition: all .3s ease-in-out;

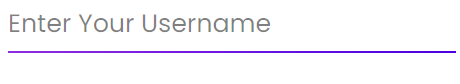
    transform-origin: left;

}

.input-field input:focus ~ .underline::after {

    transform: scaleX(1);

}



Com a tag **transform: scalex(0)** deixamos a animação oculta até que clicamos dentro do input, para isso precisamos inserir outra declaração com a tag **transfomr: scalex(1)**

**Site de Serviços - responsivo**

Abaixo estamos criando um menu de navegação usando a tag (**nav)**, dentro colocamos uma lista desordenada **(ul)** e depois usamos a tag (**li)** para lista de itens junto da tag (**a)** de links para listarmos os links de conteúdo de exibição de maneira individual, **exemplo abaixo:**

    <nav><a *href*="#">Julio Fotos</a>

        <ul>

            <li><a *href*="#sobre">sobre</a></li>

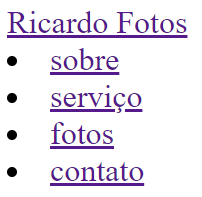
            <li><a *href*="#serviço">serviço</a></li>

            <li><a *href*="#foto">fotos</a></li>

            <li><a *href*="#contato">contato</a></li>

        </ul>

    </nav>



nav {

    display: flex;

    flex-direction: column;

    align-items: center;

    background-color: #191970;

}

nav a{

    text-decoration: none;

    color: #ffffff;

    font-weight: 800;

    font-size: 32px;

    font-family: "Yellowtail",sans-serif

}

nav ul{

    display: flex;

    flex-direction :column;

    align-items: center;

    list-style: none;

    padding: 0;

    width: 100%;

}

nav ul li{

    text-align: center;

    width: 100%;

}

nav ul li a{

    display: inline-block;

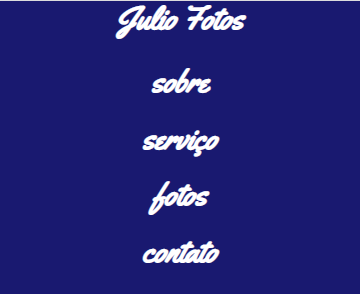
    padding: 10px 0;

    width: 100%;

    text-decoration: none;

    color: #ffffff;

}



Abaixo montamos um cabeçalho com o **id “topo”** **(header)** para nosso site colocando informações iniciais, dentro temos um titulo **(h1)** e um paragrafo **(p),** além de um link **(a),** com a **classe “botão”** onde posteriormente em CSS estilizamos um botão no link **(a) exemplo abaixo:**

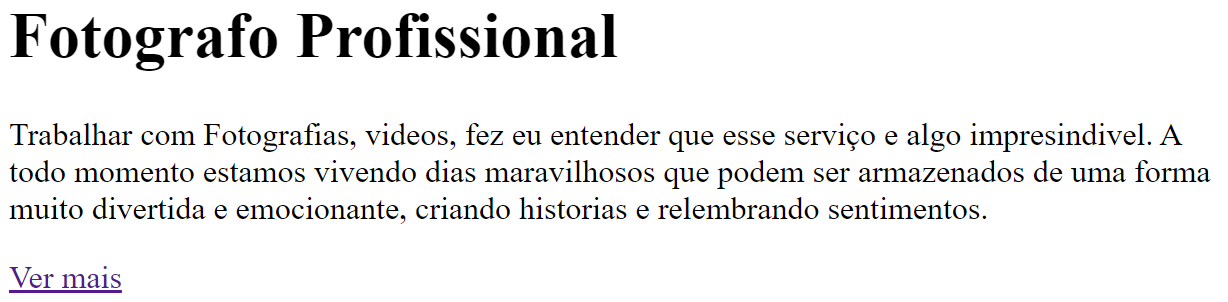
    <header *id*="topo">

        <h1>Fotógrafo Profissional</h1>

        <p>Trabalhar com Fotografias, vídeos, fez eu entender que esse serviço e algo imprescindível. A todo momento estamos vivendo dias maravilhosos que podem ser armazenados de uma forma muito divertida e emocionante, criando historias e relembrando sentimentos.</p>

        <a *href*="#serviço" *class*="botao">Ver mais</a>

    </header>



header{

    display: flex;

    flex-direction: column;

    align-items: center;

    background-image: url(*img/camera.jpg*);

    background-size: cover;

    background-repeat: no-repeat;

    background-position: center;

    color: #ffffff;

    align-items: center;

    padding: 20px;

}

header h1{

    color: #ffffff;

}

header p{

    margin-bottom: 2em;

}

header a*:hover*{

    background-color: #ffffff9f;

    color: #000000;

}

*.botao*{

    color: white;

    text-decoration: none;

    padding: 10px 30px;

    background-color: #191970;

    font-size: .9em;

    border-radius: 3px;

}



Abaixo temos uma seção com o **id “sobre”** **(section)** com um subtítulo **(h2)** e um parágrafo **(p)** falando sobre os serviços, além disso temos um link **(a)** que da no contato de contratação do serviço, **exemplo abaixo:**

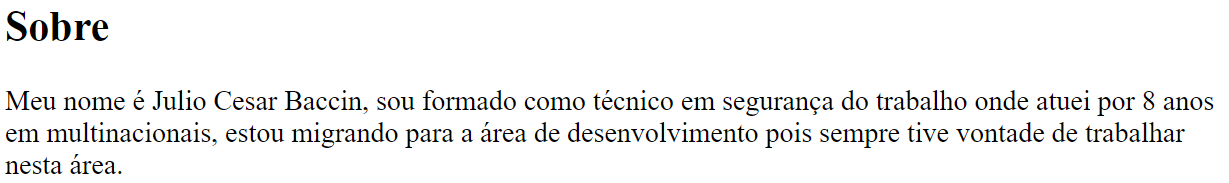
    <section *id*="sobre">

        <h2>Sobre</h2>

        <p>Meu nome é Julio Cesar Baccin, sou formado como técnico em segurança do trabalho onde atuei por 8 anos em multinacionais, estou migrando para a área de desenvolvimento pois sempre tive vontade de trabalhar nesta área. </p>

        <a *href*="#contato" *class*="botao">Contrate Agora</a>

    </section>



*#sobre*{

    display:flex;

    flex-direction: column;

    align-items: center;

    padding: 10px 10px 20px 20px;

    background-color: #191970;

    color: #ffffff;

}

*#sobre* h2{

    color: #ffffff;

}

*#sobre* p{

    margin-bottom: 2.5em;

    max-width: 1000px;

}

*#sobre* a*:hover*{

    background-color: #1919709c;

    color: white;

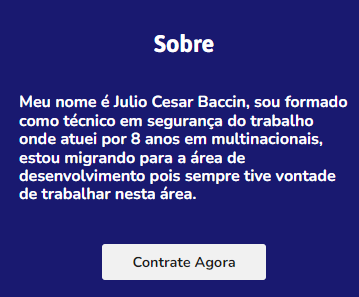
}

*#sobre* *.botao*{

    background-color: #f1f1f1;

    color: #1c1c1c;

}



Abaixo temos uma seção com o **id “serviços”** **(section)** sobre os serviços oferecidos onde temos um subtítulo em **(h2)** dentro desta seção **(section)** criamos **4 divs,** cada **div** tem uma imagem **(img)** com um subtítulo menor **(h3)** e um parágrafo **(p)** com a descrição relativa a cada serviço atribuído na **div**, **exemplo abaixo:**

    <section *id*="serviços">

        <h2>Serviços</h2>

        <div>

            <img *src*="img/p1.png" *alt*="Camera">

            <h3>Fotografias</h3>

            <p>Guarde momento importantes, como casamento, formaturas, aniversários para sempre com alta qualidade</p>

        </div>

        <div>

            <img *src*="img/p2.png" *alt*="Filme">

            <h3>Edição de Video</h3>

            <p>Quer transformar alguma lembrança em vídeo em algo emocionante com trilha sonora, textos e cores fantásticas?</p>

        </div>

        <div>

            <img *src*="img/p3.png" *alt*="Videoplayer">

            <h3>Captação de imagens</h3>

            <p>Vai participar de algum evento importante em sua vida, então pe importante recordar tudo a cada segundo para tornar tudo imortal</p>

        </div>

        <div>

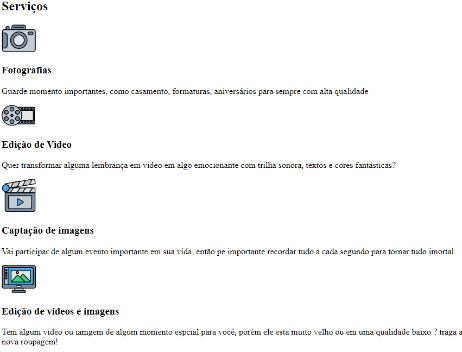
            <img *src*="img/p4.png" *alt*="Edição">

            <h3>Edição de vídeos e imagens</h3>

            <p>Tem algum vídeo ou imagem de algum momento especial para você, porém ele está muito velho ou em uma qualidade baixa? traga até nós que tornaremos seus momentos especiais com uma nova roupagem! </p>

        </div>

    </section>



*#serviços*{

    display: flex;

    flex-direction: column;

    align-items: center;

    padding: 30px;

}

*#serviços* div{

    display: flex;

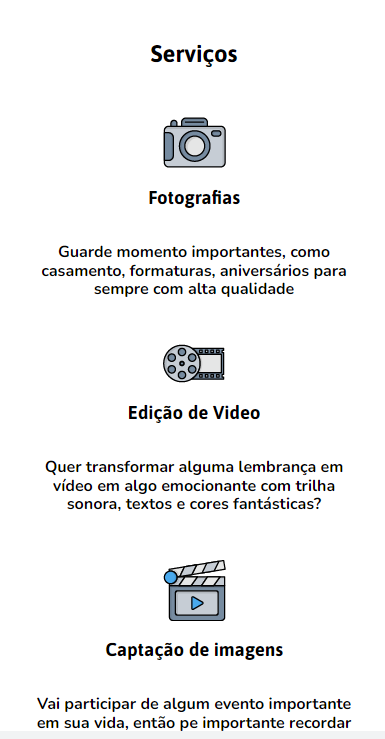
    flex-direction: column;

    align-items: center;

    text-align: center;

    margin-top: 30px;

}



Abaixo temos a seção com **id”fotos”** **(section)** de fotos com um subtítulo em **(h2)** e algumas imagens **(img)** com suas descrições **(alt=’’), exemplo abaixo:**

    <section *id*="fotos">

        <h2>Fotos</h2>

        <img *src*="img/ceu.jpg" *alt*="ceu">

        <img *src*="img/grama.jpg" *alt*="grama">

        <img *src*="img/montanha.jpg" *alt*="montanha">

        <img *src*="img/mar.jpg" *alt*="mar">

        <img *src*="img/arvores.jpg" *alt*="arvores">

        <img *src*="img/ponte.jpg" *alt*="ponte">

    </section>





*#fotos*{

    display: flex;

    flex-direction: column;

    align-items: center;

    padding-top: 20px;

}

*#fotos* h2{

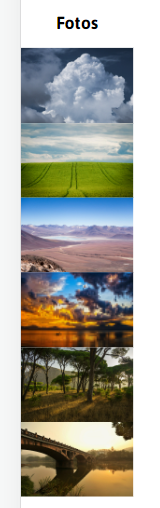
    margin-bottom: 20px;

}

*#fotos* img{

    width: 100%;

}



Abaixo temos a seção **(section)** de contatos, temos um subtítulo em **(h2),** um paragrafo **(p)** e dentro desta **section** temos **2 divs**, uma para apresentando o telefone celular com uma imagem **(img)** e descrição **(alt=” “)** e um paragrafo (p) envelopado com um link **(a)** onde estamos colocando nosso contato celular dentro e fora do link, já na outra **div** temos a mesma configuração porem voltada para o **e-mail**, onde dentro do link **(a href=” “)** usamos mailto, **exemplo abaixo:**

    <section *id*="contato">

        <h2>Contato</h2>

        <p>Vem falar com a gente</p>

        <div>

            <img *src*="img/tel.png" *alt*="telefone">

            <p><a *href*="tel:42991271991">(42) 991271991</a></p>

        </div>

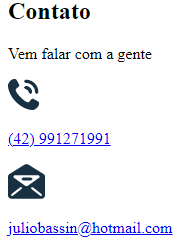
        <div>

            <img *src*="img/mail.png" *alt*="email">

            <p><a *href*="mailto:juliobassin@hotmail.com">juliobassin@hotmail.com</a></p>

        </div>

    </section>



*#contato*{

    display: flex;

    flex-direction: column;

    padding: 20px;

    text-align: center;

}

*#contato* h2, p{

    color: #000000;

}



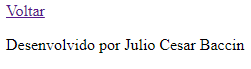
Abaixo temos nosso rodapé **(footer),** dentro temos um link **(a)** com um **href=” #topo”**>voltar< para voltar ao topo do site, **exemplo abaixo:**

    <footer>

        <a *href*="#topo">Voltar</a>

        <p>Desenvolvido por Julio Cesar Baccin</p>

    </footer>



footer{

    text-align: center;

    padding: 10px;

    background-color: #191970;

}

footer a{

    text-decoration: none;

    color: #00ffff;

}

footer p{

    color: #ffffff;

}

footer a*:hover*{

    background-color: #00ffff3b;

    text-decoration: underline;

}



Após toda a configuração em **HTML** e **CSS** acima, aplicamos um **@media screen and (min-width: 768px),** isso significa que toda propriedade acima será aplicada em dispositivos e **viewports** menores que 768px, acima disso a configuração abaixo que será aplicada, vamos ver abaixo qual foi a configuração aplicada para dispositivos com **mais de 768px** e suas mudanças:

@media screen and (min-width: 768px) {

**Menu de navegação, (nav):**

    nav{

        flex-direction: row;

        justify-content: space-around;

    }

    nav ul{

        flex-direction: row;

        width: 70%;

    }



**Cabeçalho, (header):**

    header{

        height: 600px;

        height: 60vh;

        justify-content: center;

    }



**Seção sobre, (section):**

*#sobre*{

        height: 250px;

        height: 30vh;

        justify-content: center;

        min-height: 250px;

    }



**Seção serviços, (section):**

*#serviços*{

        flex-direction: row;

        flex-wrap: wrap;

        align-items: flex-start;

        max-width: 1000px;

        margin: 0 auto;

        }

*#serviços* h2{

         text-align: center;

         width: 100%;

     }

*#serviços* div{

         flex-grow: 1;

         flex-basis: 150px;

         padding: 0 10px;

     }



**Seção fotos, (section):**

*#fotos*{

         flex-direction: row;

         flex-wrap: wrap;

     }

*#fotos* h2{

        width: 100%;

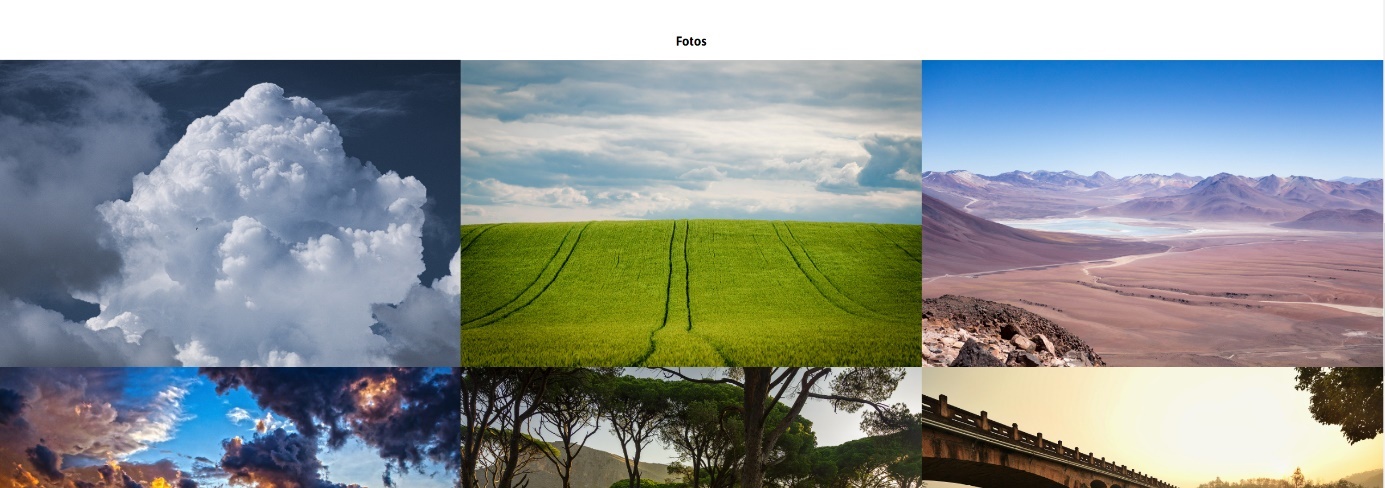
        text-align: center;

     }

*#fotos* img{

         width: 33.3333333%

     }



**Seção contatos, (section):**

*#contato*{

         flex-direction: row;

         flex-wrap: wrap;

         justify-content: space-around;

         max-width: 500px;

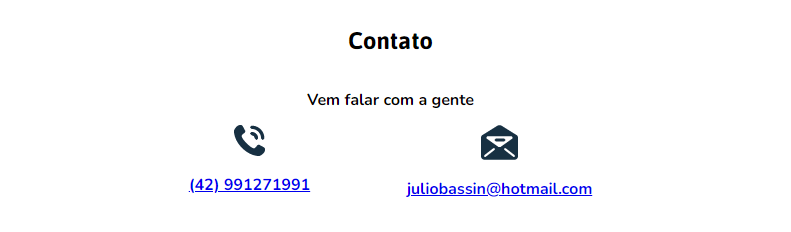
         margin: 0 auto;

     }

*#contato* h2, *#contato* p{

         width: 100%;

     }



**TELA DE LOGIN TRANSPARENTE – RESPONSIVA**

Abaixo temos a estrutura básica de uma tela de login, podemos observar que é criado uma seção **(section)** dentro do corpo **(body)**, após isso e criado uma **div** com a **class (form-container),** então criamos um título **(h1)** e dentro criamos uma forma **(form)** para o formulário, já dentro de forma criamos uma **div** com a **classe (control)** para cada informação individual, sendo elas, o nome **(name)** a senha **(password),** também criamos uma mensagem usando **span** com a propriedade **“check-box”** e o texto **“remembre me**”, para criarmos uma box de marcação com texto.

<body>

    <section>

        <div class="form-container">

            <h1>login form</h1>

            <form action="#">

                <div class="control">

                    <label for="name">Name</label>

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

                </div>

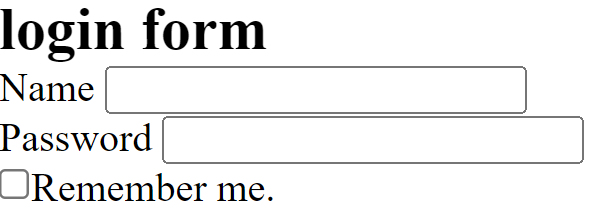
                <div class="control">

                    <label for="psw">Password</label>

                    <input type="password" name="password" id="password">

                </div>

                <span><input type="checkbox">Remember me.</span>



section{

    position: relative;

    height: 100vh;

    width: 100%;

    background-image: url(pexels-francesco-ungaro-2995276.jpg);

    background-size: cover;

    background-repeat: no-repeat;

    background-position: center right;

}

.form-container{

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%,-50%);

    background: linear-gradient(rgba(0,0,0,0.3), rgba(0,0,0,0.2));

    width: 380px;

    padding: 50px 30px;

    border-radius: 10px;

    box-shadow: 7px 7px 60px #000;

}

span input[type="checkbox"]{

    margin-right: 5px;

}

h1{

    text-transform: uppercase;

    font-size: 2em;

    text-align: center;

    margin-bottom: 2em;

}

.control input{

    width: 100%;

    display: block;

    padding: 10px;

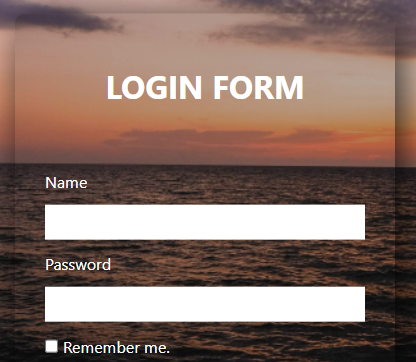
    color: #222;

    border: none;

    outline: none;

    margin: 1em 0;

}



Abaixo criamos uma **div** com a classe **(control)** e dentro um botão de login usando a propriedade **type=”submit”** com o **value=”Login”** .

                <div class="control">

                    <input type="submit" value="Login">

                </div>

</form>



.control input[type="submit"]{

    background-color: crimson;

    color:#fff;

    text-transform: uppercase;

    font-size: 1.2em;

    opacity: .7;

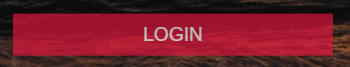
    transition: opacity .3s ease;

}

.control input[type="submit"]:hover{

    opacity: 1;

}



Abaixo, já fora da forma do formulário **(form)** criamos uma **div** com a classe **(link)** e então inserimos um **a** **(link**) com o texto de **Forgot Password ?.**

            <div class="link">

                <a href="#">Forgot Password ?</a>

            </div>

        </div>

    </section>

</body>



.link{

    text-align: center;

}

.link a{

    text-decoration: none;

    color: #fff;

    opacity: .6;

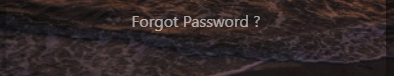
    transition: opacity .3s ease;

}

.link a:hover{

    opacity: 1;

}



**Formulário de login – transparente julio**

Abaixo iniciamos o conteúdo com o HTML, inserimos uma tag **section (seção)** dentro do **body** que será onde colocaremos nosso **backgroud-image url(),** logo abaixo colocamos uma **div** com a **classe =”container”** que será onde colocaremos nosso conteúdo e estilizaremos a forma em css, também inserimos um titulo **(h1)** para nosso formulário e abaixo inserimos a tag **form (formulário),** dentro da tag **form (formulário)** inserimos 3 **divs** com a mesma **classe=** **“conteúdo”** e dentro de cada **div** colocamos um **nome de exibição (label)** e um **campo de preenchimento (input)** do **tipo texto (type=”text”),** abaixo temos uma **mensagem de exibição (span)** com um **campo de preenchimento (input)** do tipo **checkbox (type=”checkbox”)** com o texto **remembre me.**

<body>

    <section>

        <div class="container">

            <h1>Formulário de login - julio cesar baccin</h1>

            <form action="#">

                <div class="conteudo">

                    <label for="#">Login</label>

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

                </div>

                <div class="conteudo">

                    <label for="#">Password</label>

                    <input type="password" name="password" id="password">

                </div>

                <span><input type="checkbox">Remember-me.</span>

            </form>

        </div>

    </section>

</body>

\*{

    margin: 0;

    padding: 0;

    border: 0;

    box-sizing: border-box;

}

section{

    position: relative;

    background-image: url(pexels-andy-vu-3244513.jpg);

    width: 100%;

    height: 100vh;

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center bottom;

    font-family: 'Abel', sans-serif;

}

.container{

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

    background:linear-gradient(rgba(0,0,0,0.3) rgba(0,0,0,0.2));

    width: 380px;

    padding: 50px 30px;

    border-radius: 10px;

    box-shadow: 7px 7px 50px #000;

}

.container h1{

    font-size: 2em;

    text-transform: uppercase;

    color: white;

    text-align: center;

    margin-bottom: 20px;

}

.conteudo input{

    width: 100%;

    display: block;

    padding: 10px;

    color: #222;

    border: none;

    outline: none;

    margin: 0.5em 0;

    opacity: .9;

    transition:  opacity .3s ease;

}

.conteudo input:hover{

    opacity: .5;

}

.conteudo label{

    font-size: 1em;

    color: white;

    text-transform: uppercase;

}

span input[type="checkbox"]{

    margin-right: 5px;

}

span{

    color: white;

    font-size: 1em;

}



Abaixo temos outra **div** com a **classe (conteúdo)** que será nosso botão de login, aplicamos um **input** do tipo **submit (type=”submit”)** com **valor “login” (value=”login”).**

<body>

    <section>

        <div class="container">

            <h1>Formulário de login - julio cesar baccin</h1>

            <form action="#">

                <div class="conteudo">

                    <input type="submit" value="Login">

                </div>

            </form>

        </div>

    </section>

</body>

.conteudo input[type="submit"]{

    background-color: #BF9484;

    color: rgba(255, 255, 255, 0.842);

    text-transform: uppercase;

    font-size: 1em;

    opacity: .8;

    transition: opacity .3s ease;

}

.conteudo input[type="submit"]:hover{

    opacity: 1;

}



Abaixo, já fora da tag **form** **(formulário)** temos uma **div** com a **classe=”link”** e dentro atribuímos um **link** **(a)** com o texto **forgot password ?**

<body>

    <section>

        <div class="container">

            <h1>Formulário de login - julio cesar baccin</h1>

            <form action="#">

            </form>

                <div class="link">

                    <a href="#">Forgot Password ?</a>

                </div>

        </div>

    </section>

</body>

.link{

    text-align: center;

    margin-top: 20px;

}

.link a{

    color: white;

    text-decoration: none;

    opacity: .8;

    transition: opacity .3s ease;

}

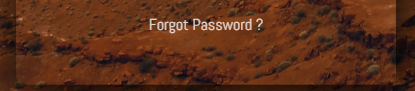
.link a:hover{

    opacity: 1;

    font-size: 1.2em;

    transition: .3s ease;

}



**Pagina de contato com formulário**

Iniciamos nossa pagina criando uma **section** com a **classe=”contato”** que será o elemento pai depois de **body**, será dentro desta **section** que vamos aplicar os demais elementos, criamos uma **div** com a **class=” conteúdo”** onde inserimos um subtítulo **(h2)** e um parágrafo **(p).**

<body>

    <secion class="contato">

        <div class="conteudo">

            <h2>Contact us</h2>

            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aperiam error quis cum voluptas perspiciatis voluptatum asperiore</p>

        </div>

    </secion>

</body>



@charset "UTF-8";

@import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: 'Poppins', sans-serif;

}

*.contato*{

    position: relative;

    min-height: 100vh;

    padding: 50px 100px;

    display: flex;

    justify-content: center;

    align-items: center;

    flex-direction: column;

    background-image: url(*pexels-eberhard-grossgasteiger-1612351.jpg*);

    background-size:cover;

    background-position: center center;

}

*.contato* *.conteudo*{

    max-width: 800px;

    text-align: center;

}

*.contato* *.conteudo* h2{

    font-size: 36px;

    font-weight: 500;

    color:#fff;

}

*.contato* *.conteudo* p{

    font-weight: 300;

    color:#fff;

}



Depois criamos uma **div** com a **classe=”container”** será esta **div** que usaremos posteriormente para estilizarmos a forma do nosso formulário e colocaremos as demais informações. Dentro da **div=”container”** criamos uma **div** com a **classe=” contatoinfo”** e dentro desta div criamos outra **div** com a **classe=” box”** e dentro dela colocamos outras **divs** com a **classe=” ícone”** e **classe=” texto”** onde colocamos dentro um subtítulo **(h3)** e um parágrafo **(p)** com nossas informações de contato, replicamos estas informações acima em mais **2 divs abaixo**, cada **div** com informações do **endereço, telefone, E-mail e ícones respectivos.**

<body>

    <secion class="contato">

        <div class="container">

            <div class="contatoinfo">

                <div class="box">

                    <div class="icon"><i class="fa-solid fa-map-location-dot"></i></div>

                    <div class="texto">

                        <h3>Endereço</h3>

                        <p>Avenida Pinheiros, 720, Centro - Trindade do Sul/RS</p>

                    </div>

                </div>

                <div class="box">

                    <div class="icon"><i class="fa-solid fa-phone"></i></div>

                    <div class="texto">

                        <h3>Telefone</h3>

                        <p>(42) 9 9127-1991</p>

                    </div>

                </div>

                <div class="box">

                    <div class="icon"><i class="fa-solid fa-envelope"></i></div>

                    <div class="texto">

                        <h3>E-mail</h3>

                        <p>juliobassin@hotmail.com</p>

                    </div>

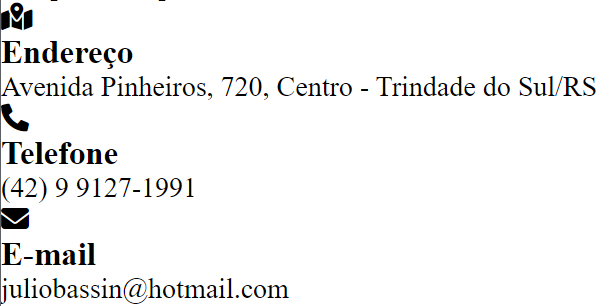
                </div>

            </div>

        </div>

    </secion>

</body>



*.container*{

    width:100%;

    display: flex;

    justify-content: center;

    align-items: center;

    margin-top: 30px;

}

*.container* *.contatoinfo*{

    width: 50%;

    display: flex;

    flex-direction: column;

}

*.container* *.contatoinfo* *.box*{

    position: relative;

    padding: 20px 0;

    display: flex;

}

*.container* *.contatoinfo* *.box* *.icone*{

    min-width: 60px;

    height: 60px;

    background: #fff;

    border-radius: 50%;

    display: flex;

    align-items: center;

    justify-content: center;

    font-size: 22px;

}

*.container* *.contatoinfo* *.box* *.texto*{

    display: flex;

    flex-direction: column;

    color: #fff;

    font-size: 16px;

    margin-left: 20px;

    font-weight: 300;

}

*.container* *.contatoinfo* *.box* *.texto* h3{

    font-weight: 500;

    color: #00bcd4;

}



Abaixo então criamos uma **div** com a **classe=”formadecontato”** e dentro inserimos a tag **form( formulário)**, onde inserimos um subtítulo **(h2)** e logo abaixo algumas **divs** com a **classe=”inputbox”** onde dentro colocamos um campo de preenchimento **(input)** do tipo texto **(type=”text”)** um nome **(name=”#”)** e usamos o elemento **required=”required”** que avisa o usuário sobre a necessidade do preenchimento do campo **(input),** logo abaixo temos uma mensagem com o nome de exibição **(span),** replicamos a **div** acima mais **3 vezes** sendo que na **div do e-mail** na parte do tipo em vez do tipo text usamos o **type=”e-mail”** e na parte da área na **div** no campo de texto do usuário usamos a tag **textarea** para criamos um campo maior para que o usuário possa relatar suas experiencia ou afins, finalizamos com a **div** do botão, colocando um input e dentro um **type=”submit”** com o valor **send (value=”send”).**

<body>

    <secion class="contato">

<div class="container">

<div class="formadecontato">

                <form action="#">

                    <h2>Mandar Mensagem</h2>

                    <div class="inputbox">

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

                        <span>Nome Completo</span>

                    </div>

                    <div class="inputbox">

                        <input type="email" name="#" required="required">

                        <span>E-mail</span>

                    </div>

                    <div class="inputbox">

                        <textarea required="required"></textarea>

                        <span>Digite sua Mensagem</span>

                    </div>

                    <div class="inputbox">

                        <input type="submit" name="#" value="Send">

                    </div>

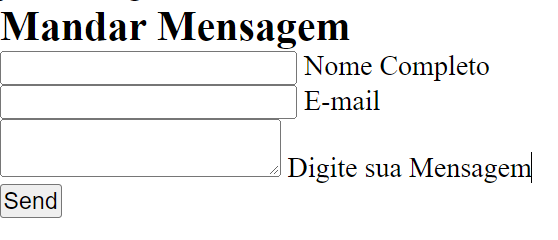
                </form>

            </div>

        </div>

    </secion>

</body>



*.formadecontato*{

    width: 40%;

    padding: 40px;

    background:#fff;

}

*.formadecontato* h2{

    font-size: 30px;

    color: #333;

    font-weight: 500;

}

*.formadecontato* *.inputbox*{

    position: relative;

    width: 100%;

    margin-top: 10px;

}

*.formadecontato* *.inputbox* input, textarea{

    width: 100%;

    padding: 5px 0;

    font-size: 16px;

    margin: 10px 0;

    border: none;

    border-bottom: 2px solid #333;

    outline: none;

    resize: none;

}

*.formadecontato* *.inputbox* span{

    position: absolute;

    left: 0;

    padding:  5px 0;

    font-size: 16px;

    margin: 10px 0;

    pointer-events: none;

    transition:0.5s;

    color: #666;

}

*.formadecontato* *.inputbox* input*:focus* ~ span,

*.formadecontato* *.inputbox* textarea*:focus* ~ span{

    color: #e91e63;

    font-size: 12px;

    transform: translateY(-20px);

}

*.formadecontato* *.inputbox* input*:valid* ~ span,

*.formadecontato* *.inputbox* textarea*:valid* ~ span{

    color: blue;

}

*.formadecontato* *.inputbox* input[*type*="submit"]{

    width: 100px;

    background: #00bcd4;

    color:#fff;

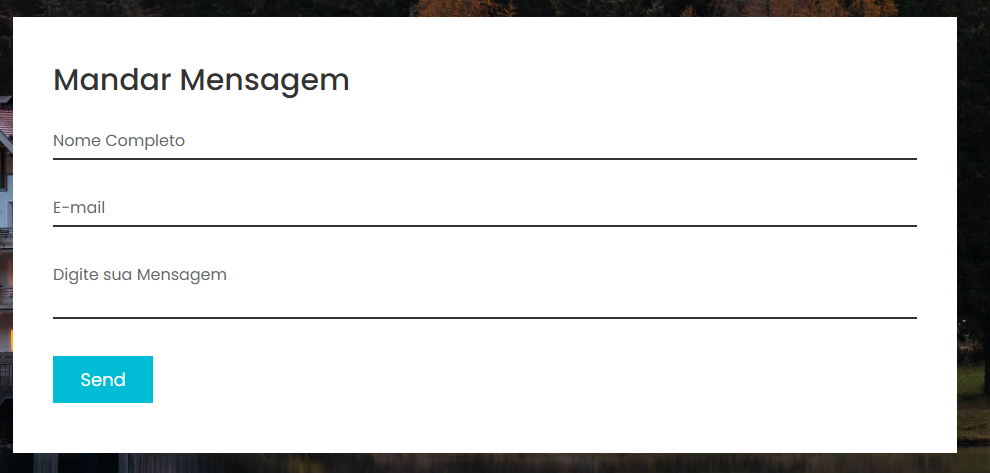
    border: none;

    cursor: pointer;

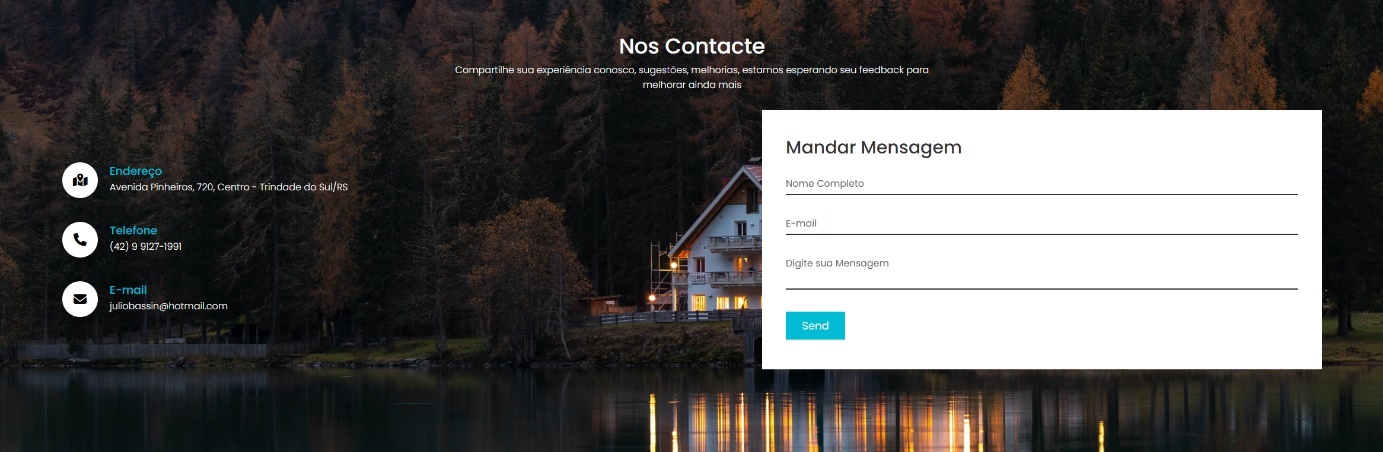
    padding: 10px;

    font-size: 18px;

}



Abaixo temos uma print do site completo em uma **viewport de 2560x937**



Depois de toda a configuração acima, aplicamos um **@media carry** para configurarmos a responsividade do nosso formulário de contato, no caso em questão aplicamos um **@media (max-width: 991px)**, o que significa que as propriedades acima, somente serão aplicadas em telas com uma **viewport acima de 991px**, vamos ver abaixo quais configurações foram aplicadas em telas com uma **viewport menor que 991** e como ficarão:

@media (max-width: 991px){

*.contato*{

        padding: 50px;

    }

*.container*{

        flex-direction: column;

    }

*.container* *.contatoinfo*{

        margin-bottom: 40px;

        width: 100%;

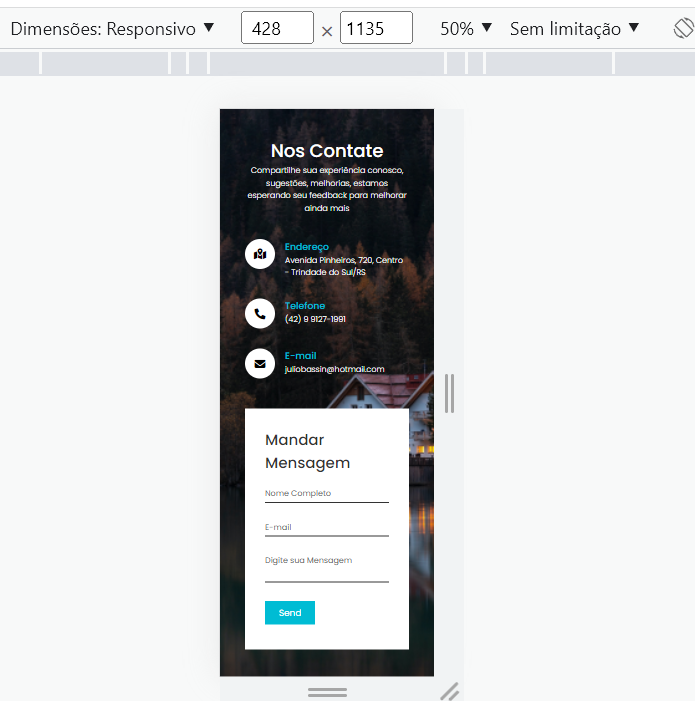
    }

*.container* *.formadecontato*{

        width: 100%;

    }

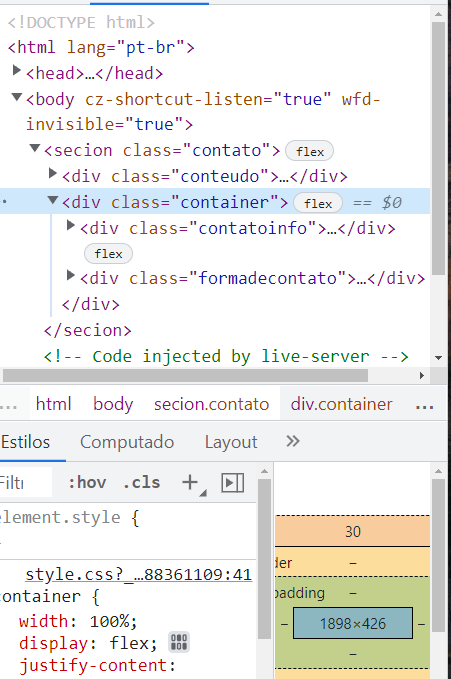
}

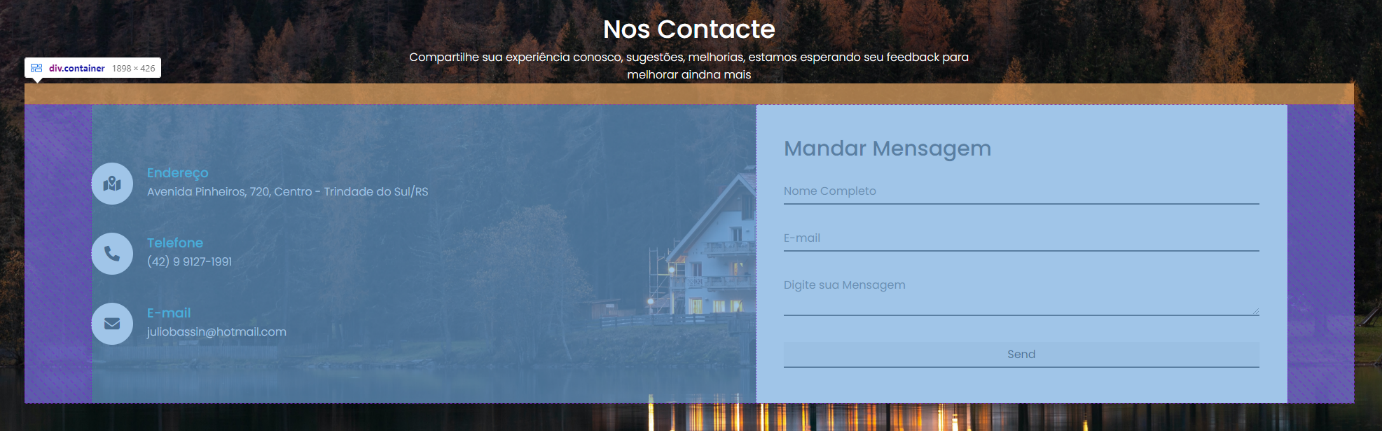


**Largura dos elementos com width 100%**

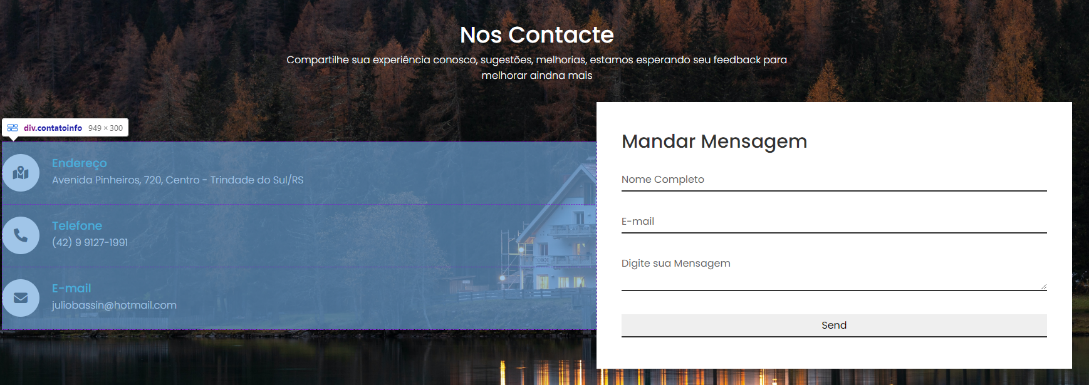
Os **elementos filhos** quando atribuídos **width: 100%** significa que eles **vão ocupar 100% da largura do elemento pai**, podemos verificar esse **100% em pixel**, logo se quisermos atribuir largura menores para elementos filhos, basta verificarmos a quantia da largura (width) em pixel no vscode, se temos uma largura do elemento pai de 100% com 1078 pixel e queremos colocar metade no elemento filho (dentro do pai) basta inserirmos **width 50%** ou **dividirmos 1078 por 2.**

Observamos abaixo o elemento pai com a **classe =”container”** possui uma largura de **1898px,** ele está com uma configuração de **width 100%,** o que significa que a largura dele irá ocupar 100% do comprimento de seu **elemento pai**, no caso a **div** na qual ele está dentro, que na imagem podemos identificar como o elemento **section class=”contato”.**

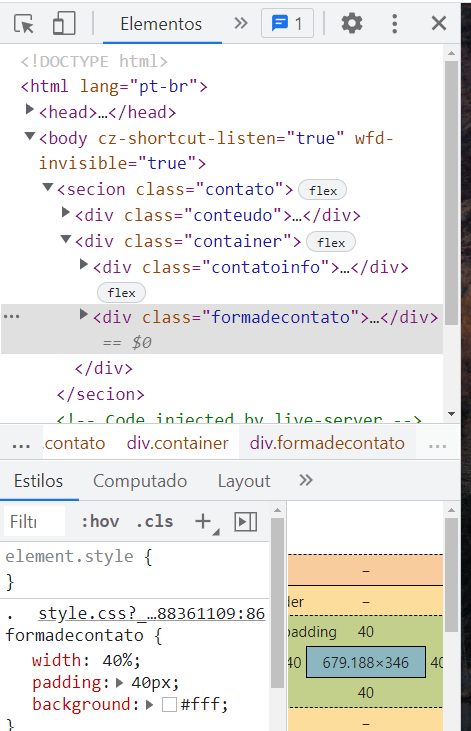


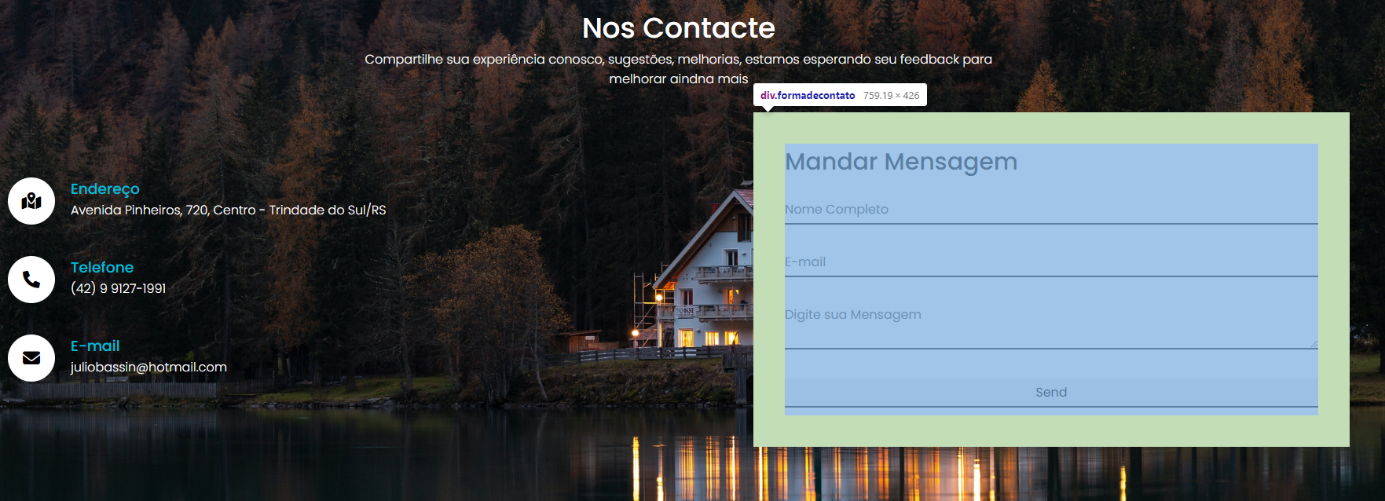


O elemento pai **classe=”container”** possui 2 elementos filhos sendo eles a **div class=”contatoinfo”** e a **div class=”formadecontato”,** abaixo podemos observar que o elemento filho **div class=”contatoinfo”** está setado em **50% width,** o que significa que ele irá ocupar uma largura total de 50% do comprimento total de seu elemento pai, no caso a **div class=”container”**.



Já abaixo o outro elemento filho **div class=”formadecontato”** está setado com **40% de width**, então o mesmo vái ocupar **somente 40% do largura total de seu elemento pai** no caso a **div class=”container”.**





**Website com gif**

O posicionamento dos itens **dentro do HMTL e muito importante** pois ele pode definir qual item ocupara o **espaço primeiro**, podemos observar abaixo que o elemento **img** **classe =” logo”** vem depois da lista de itens **(UL) > (LI)** então ele acaba sendo posicionado depois da lista em nosso site:

<body>

    <div *class*="hero">

        <nav>

            <ul>

                <li><a *href*="#">Home</a></li>

                <li><a *href*="#">Sobre</a></li>

                <li><a *href*="#">Contato</a></li>

                <li><a *href*="#">Portifolio</a></li>

            </ul>

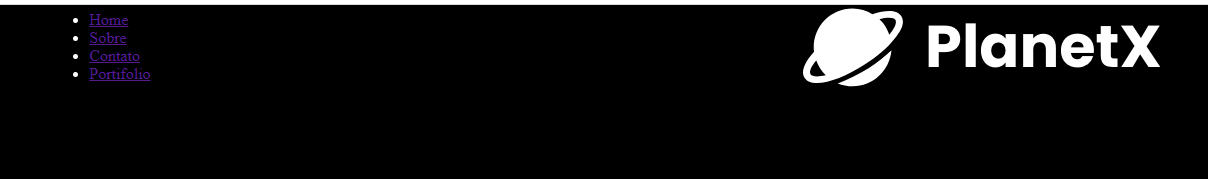
            <img *src*="planetX-logo.png" *class*="logo">

        </nav>

    </div>

</body>

</html>



Já abaixo temos a situação inversa o elemento **img class=”logo”** vem posicionado primeiro que que a lista de itens **(ul)** então ele acaba sendo posicionado primeiramente que o menu dentro do site:

<body>

    <div *class*="hero">

        <nav>

            <img *src*="planetX-logo.png" *class*="logo">

            <ul>

                <li><a *href*="#">Home</a></li>

                <li><a *href*="#">Sobre</a></li>

                <li><a *href*="#">Contato</a></li>

                <li><a *href*="#">Portifolio</a></li>

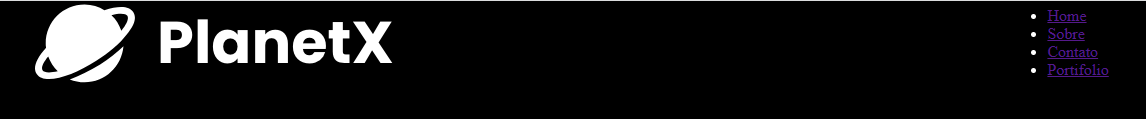
            </ul>

        </nav>

    </div>

</body>

</html>



Podemos observar abaixo que tanto a lista de itens **(UL)** e os itens **(li)** quanto o logo **img** **class=”logo”** estão dentro do elemento nav (menu de navegação), estamos aplicando em CSS a propriedade **justify-content: space between** que faz com que o elemento crie uma espaço entre eles fazendo assim um posicionando dependendo da quantidade de itens dentro deste elemento.

