

## RESUMINDO HTML

Fazer uma Linha: **HR**

Quebra a linha: **BR**

Fazer < : **&lt**

Fazer > : **&gt**

Comentário HTML: **<!--....-->**

Símbolos especiais:

Símbolo	Descrição	Entitiy
	<i>Espaço em branco</i>	<code>&amp;nbsp;</code>
®	<i>Marca registrada</i>	<code>&amp;reg;</code>
©	<i>Copyright</i>	<code>&amp;copy;</code>
™	<i>Trade Mark</i>	<code>&amp;trade;</code>
€	<i>Euro</i>	<code>&amp;euro;</code>
£	<i>Libra (pound)</i>	<code>&amp;pound;</code>
¥	<i>Yen</i>	<code>&amp;yen;</code>
¢	<i>Cent</i>	<code>&amp;cent;</code>
∅	<i>Vazio</i>	<code>&amp;empty;</code>
Σ	<i>Soma</i>	<code>&amp;sum;</code>
Δ	<i>Delta</i>	<code>&amp;Delta;</code>
←	<i>Seta esquerda</i>	<code>&amp;larr;</code>
↑	<i>Seta acima</i>	<code>&amp;uarr;</code>
→	<i>Seta direita</i>	<code>&amp;rarr;</code>
↓	<i>Seta baixo</i>	<code>&amp;darr;</code>

Emoji: **&#x** ( \_\_ é para incluir código)

Incluir Imagem: **<img src="" alt="">**

Favicon: no <head> colocar **link:favicon**

Escrever texto aleatório: **lorem**

Negrito: **Strong**

Itálico: *Em*

Texto deletado: ~~Del~~

Texto inserido: Ins

Abreviação: **abbr**

Lista ordenada: **Ol**

Lista não ordenada: **Ul**

Começar a lista: **Li**

Links externos: `<a> target="_blank" rel="external"`

Links internos: `<a> rel="next" ou "prev"`

Links p/download: `<a> download= nome do arquivo type = ".../">`

```
<body>
  <h1>Usando links externos</h1>
  <p>Você também pode acessar o meu <a href="https://www.instagram.com/gabriel_d_jesus12/"> Instagram</a></p>
  <p>Você pode acessar o meu perfil no <a href="https://github.com/GabrielJesus012"
  target="_blank"rel="external"> GitHub</a></p>

  <h1>Usando links internos</h1>
  <p>Está é a primeira página do site. Se você quiser, acesse também <a href="pag02.html" rel="next">segunda
  página</a></p>
  <p>Você também pode acessar a nossa <a href="noticias/pag03.html" target="_self" rel="next">página de
  notícias</a></p>

  <h1>Usando link para download</h1>
  <UL> <!--Usei Ul>LI*4>A-->
    <li><a href="livro/10 - Ligações em toda parte.pdf" download="10 - Ligações em toda parte.pdf"
    type="application/pdf">Baixar Livro em PDF</a></li>
    <li><a href="livro/10 - Ligações em toda parte.zip" download="10 - Ligações em toda parte.zip"
    type="application/zip">Baixar Livro compactado em ZIP</a></li>
  </UL>
</body>
```

Type mais utilizados:

- |                   |              |              |
|-------------------|--------------|--------------|
| • application/zip | • video/mp4  | • audio/mpeg |
| • text/html       | • video/H264 | • font/ttf   |
| • text/css        | • video/JPEG | • image/jpeg |
| • text/javascript | • audio/aac  | • image/png  |

Imagem dinâmica: `<picture> <source:media:type> *mude min-width para max-width`  
`<img> //Ordem: P>M>G`

```

9 <body>
10 <h1>Imagem dinâmica</h1>
11 <p>Tente abrir esse site em vários dispositivos diferentes ou simplesmente aumente e diminua o tamanho do
    seu navegador</p>
12 <picture>
13 <source media="(max-width:750px)" srcset="imagens/foto-p.png" type="image/png">
14 <source media="(max-width: 1050px)" srcset="imagens/foto-m.png" type="image/png">
15 
16 </picture>
17
18 </body>
19 </html>

```

Áudio: `<audio preload="metadata" controls> <source src>`

```

18 <h1>Reproduzindo audio</h1>
19 <p>Vamos aprender a reproduzir audios em HTML5</p>
20 <!-- FORMATOS ACEITOS NOS NAVEGADORES: MP3 WAV OGG-->
21 <audio preload="metadata" controls>
22 <source src="midia/guanacast-33.mp3" type="audio/mpeg">
23 <source src="midia/guanacast-33.ogg" type="audio/ogg">
24 <source src="midia/guanacast-33.wav" type="audio/wav">
25 <p>Seu navegador não suporta áudio.</p>
26 <p>Clique <a href="midia/guanacast-33.mp3" download="guanacast-33.mp3" type="audio/mpeg">aqui</a> para
    baixar o arquivo MP3</p>
27 </audio>
28 <p>Clique <a href="midia/guanacast-33.mp3" download="guanacast-33.mp3" type="audio/mpeg"
    target="_blank">aqui</a> se quiser baixar o arquivo MP3</p>
29 </body>
30 </html>

```

Vídeo:

> localmente: `<video width poster controls> <source src>`

```

9 <body>
10 <h1>Inserindo vídeos hospedados localmente</h1>
11 <p>Este vídeo está hospedado no meu próprio servidor</p>
12 <video width="500" poster="imagens/capa.png" controls>
13 <source src="midias/video.mp4" type="video/mp4">
14 <source src="midias/video.m4v" type="video/mp4">
15 <source src="midias/video.webm" type="video/webm">
16 <source src="midias/video.ogv" type="video/ogv">
17 <p>Seu Navegador não tem compatibilidade com reprodução de vídeos.</p>
18 </video>
19 <p>Clique <a href="midias/video.mp4" download="video.mp4" type="video/mp4" target="_blank">aqui</a> se
    quiser baixar o vídeo</p>
20 </body>
21 </html>

```

> Externo: copiar `intregar` (iframe)

## Grouping Tags de Semantica:

Cabeçalho: `<header>`

Para menus, links, navegação: `<nav>`

Principal: `<main>`

Seção: `<section>`

Artigo: `<article>`

Periférico ao artigo (ex:escritor/autor): `<aside>`

Rodapé: `<footer>`

*Usando isso posso personalizar cada um do jeito que preferir!*

O diagrama à esquerda ilustra a hierarquia de uma página web. No topo, há um bloco 'header' contendo um sub-bloco 'nav'. Abaixo, o bloco 'main' contém um 'section' e um 'aside'. O 'section' possui dois 'article', cada um com seu próprio 'aside'. No rodapé, há um bloco 'footer'. À direita, o código HTML demonstra a implementação desses elementos, com comentários em português explicando o conteúdo de cada tag.

```
9 <header>
10   <h1>Meu Site</h1>
11   <nav>link link link link...</nav>
12 </header>
13 <main>
14   <section>
15     <article>
16       <h2>Título</h2>
17       <p>Texto do artigo</p>
18     </article>
19     <article>
20       <h2>Título</h2>
21       <p>Texto do artigo</p>
22       <aside>
23         conteúdo periférico do artigo
24       </aside>
25     </article>
26   </section>
27   <aside>
28     conteúdo periférico do site
29   </aside>
30 </main>
31 <footer>
32   conteúdo do rodapé
33 </footer>
```

Análise o diagrama do lado esquerdo e o código do lado direito da imagem acima. Veja a hierarquia entre os elementos e quais deles estão dentro um do outro.

## Tabelas

Para criar uma tabela a primeira coisa que faz é criar a tag `<table>` que delimita a tabela.

### Hierarquia de tabelas simples:

- Table= tabela
- Table Row(TR) = linha de tabela
- Table Header (TH) = cabeçalho de tabela
- Table Data(TD) = dado de tabela

//

Tudo que é dado: td

Tudo que é titulo: th

//

### Anatomia para tabelas grandes

- Table
- Caption (legenda da tabela) SÓ TEXTO
- Thead (para cabeçalho) no qual pode ter tr,td, th
- Tbody (para corpo) no qual pode ter tr,td, th
- Tfoot (para rodapé) no qual pode ter tr,td, th

```
<body>
  <h1>Uso de tabelas grandes</h1>
  <table>
    <thead>
      <tr>
        <th>Estado</th>
        <th>População</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Sao Paulo</td>
        <td class="num">45 919 049</td>
      </tr>
      <tr>
        <td>Minas Gerais</td>
        <td class="num">21 168 791</td>
      </tr>
      <tr>
        <td>Rio de Janeiro</td>
        <td class="num">17 264 943</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <th>Total de Habitantes</th>
        <td class="num">210 147 125</td>
      </tr>
    </tfoot>
  </table>
```

### Mesclagem de célula

Por exemplo, quer que uma célula ocupe 2 espaços

Coluna: `colspan="2"`.

Linha: `rowspan="2"`.

## Iframe

É inlineblock! Sua tag `<iframe>` no qual possui um `src` que pode ser preenchido com um link para um html seu ou da internet (nem todos funcionam) e entre as tag `<iframe>` `</iframe>` é possível *colocar um texto para navegadores que não usam*.

- Para **mudar o tamanho da largura e altura** do frame dentro da tag iframe coloque `height` e `width`.
- Para mudar a **forma de rolagem** do frame dentro da tag iframe coloque `scrolling="..."`.
  - **no** - não permite a rolagem da página, mesmo que ela seja maior que o tamanho estabelecido para o frame.
  - **yes** - habilita a rolagem do conteúdo, mesmo que seu conteúdo não seja grande o suficiente para preencher o frame.
  - **auto** - (padrão) a rolagem é habilitada apenas se o conteúdo for maior que o tamanho do quadro estabelecido.
- Para **centralizar o iframe** coloque `display block` e `margin auto` no css.

\*É possível criar navegação com iframe, basta colocar a tag `"name"` dentro do `<iframe>` e no link você coloca `Target: "nome q colocou"`.

```
<ul>
<li><a href="pagina001.html" target="frame">Primeira</a></li>
<li><a href="#">Segunda</a></li>
<li><a href="#">Terceira</a></li>
</ul>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum incidunt nemo c
ipsum perspiciatis expedita magnam molestias aliquam placeat!</p>

<iframe id="tela" name="frame" src="" frameborder="0">
  <p>infelizmente seu navegador nao suporta</p>
</iframe>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum incidunt nemo c
ipsum perspiciatis expedita magnam molestias aliquam placeat!</p>
```

**\*cuidado com segurança do iframe. (existe tags para segurança)**

## Formulários

Todo formulário tem que estar entre as tags: `<form>` `</form>`

Para fazer as caixinhas é só usar a tag `input` (tem vários) No qual tem que ter `"name"` e `"id"`.

Para enviar a informação é preciso usar duas tags no form o **method** (*método de envio dos dados*) `get` ou `post` e o action (*dizer para onde os dados serão enviados*).

Para fazer **ligação semântica** entre a caixa de texto e seu identificador é usado a tag **<label>** no qual precisa ser preenchido por um id que coloca em for. (*coloca-se o mesmo id do input*).

```
<body>
  <h1>Meu primeiro Formulário</h1>
  <form method="get" action="cadastro.php" autocomplete="off">
    <p><label for="iname">Nome: </label>
      <input type="text" name="Nome" id="iname"></p>
    <p><label for="iSobrenome">Sobrenome: </label>
      <input type="text" name="Sobrenome" id="iSobrenome"></p>
    <p><input type="submit" value="enviar"></p>
  </form>
</body>
```

\* o “name” é mais importante para html e php

\* o “id” é mais importante para JS e label

*Tipos de input:*

- Texto: **input:text**
- Senha: **input:password**
- Enviar: **input:submit**
- Limpar: **input:reset**
- Arquivo: **input:file** (tem que usar o método **post**)
- Numero: **input:number** (se eu colocar a função **step** é para indicar de quanto em quanto ele pula)
- Marcar um ou outro: **input:radio** (no name coloca o mesmo nome nas escolhas para poder mudar)
- Telefone: **input:tel** (se eu quiser regular o modelo do numero coloco **pattern** que começa com ^ e termina com \$)
- Email: **input:email**
- Mês: **input:month** (se eu quero que já venha pre-preenchido uma data coloco **Value**)
- Data: **input:date**
- Hora: **input:time**
- Cor: **input:color**
- Nivel: **input:range** (para colocar minino = **min= “1”** para colocar máximo = **max= “5”**)

- **Aceitar:** `input:checkbox` (geralmente após o input coloca o nome e envelope com `label`)

\*para que sempre que **for obrigatório** não passar para outra parte sem ser preenchido eu coloco no input `required`

\*para colocar **no mínimo** no por exemplo 5 letras: coloca no input `minlength=5`

\*para colocar **no máximo** por exemplo 15 letras: coloco no input `maxlength=15`

\*para mudar o **tamanho da caixa** coloca no input `size=10`

\*para colocar **dica** “por exemplo: nome no usuário” coloca no input `placeholder=”nome do usuário”`

\*para colocar **autocomplete na caixa** só do nome por exemplo, coloco no `form autocomplete= “on”` e no input coloco `autocomplete= “username”`

\*para colocar para **pular em valor** em valor coloca no input `step= “0.5”`

\*para colocar algo **pre-preenchido** uso no input `value`

\*para colocar algo **já marcado** uso no input `checked`

\*Eu posso **envolver** todos os formulários numa tag chamada `fieldset` para fazer um agrupamento e colocar uma tag `Legend` no início para ficar uma legenda

### Tag para ter escolha: `<Select>`

Dentro da tag `<select>` e `</select>` devemos utilizar a `tag option` e é possível envolver todos os option em uma tag `<optgroup>` no qual é informado um label

\* para que algum já venha selecionado é só colocar `selected` no option

```
<label for="iest">Estado</label>
<select name="estado" id="iest">

  <option value="" selected>--->Escolha<---</option>
  <optgroup label="Região Sudeste">
    <option value="MG">Minas Gerais</option>
    <option value="RJ">Rio de Janeiro</option>
    <option value="SP">São Paulo</option>
  </optgroup>

  <optgroup label="Região Nordeste">
    <option value="MA">Maranhão</option>
    <option value="PE">Pernambuco</option>
    <option value="RN">Rio Grande do Norte</option>
  </optgroup>

</select>
```

### Usar tag para escolher, mas também escrever:

Para isso usa o `input do tipo text` com `list=”id da datalist”` logo após usar a tag `<datalist>` com `<option>`



Caso queira **colocar uma área para enviar uma mensagem** podemos usar a tag `textarea` com *cols* e *rows* (recomendado usar com método post)

### Elementos OUTPUT em formulários:

O `<output></output>` se trata de um elemento de saída, utilizado somente para fins de exibição. É um elemento no qual um processo programado pode injetar os resultados de um cálculo ou o resultado de uma ação do usuário por exemplo.

Por ser um elemento de saída, isso significa que: Quando utilizado, a tag de formulário output deve estar associada a um script. O resultado gerado por esse script, como um número por exemplo, deve ser renderizado no elemento output.

EX: O primeiro é para somar 2 números e o segundo para mostrar o nível.

```
<form action="cadastro.php" method="post">

  <p>
    <label for="in1">Número 1: </label>
    <input type="number" name="n1" id="in1" min="0" max="50" required oninput="isoma.innerHTML +=Number(in1.value) + Number(in2.value)">
  </p>

  <p>
    <label for="in2">Número 2: </label>
    <input type="number" name="n2" id="in2" min="0" max="50" required oninput="isoma.innerHTML +=Number(in1.value) + Number(in2.value)">
  </p>

  <p>
    <label for="isoma">Soma:</label>
    <output name="soma" id="isoma">0</output>
  </p>
```

```
<form action="cadastro.php" method="get">

  <p>
    <label for="inum">Número: </label>
    <input type="range" name="num" id="inum" min="0" max="10" value="5" oninput="ival.innerHTML = Number(inum.value)">
    <output id="ival">5</output>
  </p>
```

## Resumo CSS

**Comentário CSS:** `/*`

**Cor de fundo:** `background-color`

Dicas de Cores: <https://color.adobe.com/pt/create/color-wheel> e <https://paletton.com/#uid=1000u0klllaFw0g0qFqFg0w0aF>

**Fazer gradiente:** `Background-image`: *linear-gradient* ou *radial-gradient* (to right, cor1, cor2,...); *Right*= direita; *left*= esquerda; *top* =cima; *bottom*= baixo

// quando usar é recomendado colocar regra global `*{height: 100%}`

## Fonte

**Estilo da fonte:** `font-style` (italic ou normal)

**Peso da fonte:** `font-weight` (*lighter*= mais magro/normal/ *bold*/*bolder* e pode variar também em 100 a 900!)

**Tamanho da fonte:** `font-size` (Normal: 16px ou 1em)

**Mudar a fonte:** `font-family` (3 fontes)

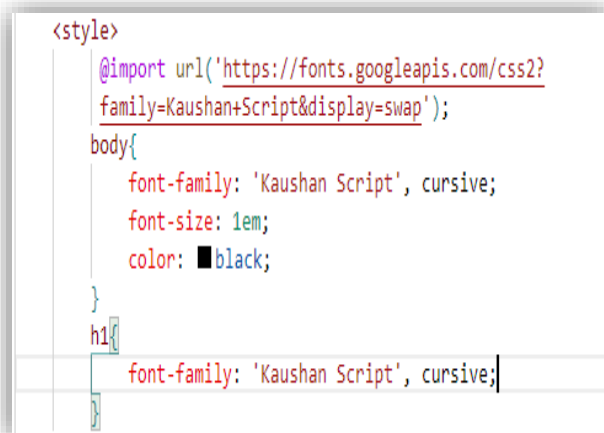
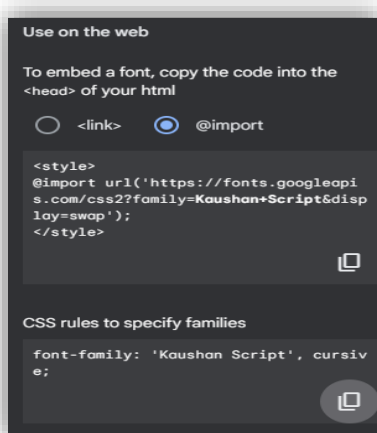
**Decoração da fonte:** `text-decoration` (underline ...)

**Cor da fonte:** `color` (cor selecionada)

Juntar tudo em um só: Ordem > Font: font-style> font-weight> font-size> font-family

**Ex:** `font: italic bolder 2em italic` (Não precisa ter todas as propriedades, mas a sequência deve ser seguida)

OBS: Posso baixar fonte ou pegar uma fonte externa! [Browse Fonts - Google Fonts](#)



```

<style>
  @font-face {
    font-family: 'Blazed';
    src: url(Blazed.ttf) format('opentype'),url(Blazed.
otf) format('truetype');
    font-weight: normal;
    font-style: normal;

    /* tipos de format()
    -opentype (otf)
    -truetype (ttf)
    -embedded-opentype
    -truetype-aat
    -svg
    */
  }
  body{
    font-family: 'blazed', Helvetica, sans-serif;
  }
</style>

```

## Alinhamento

**Alinhamento:** `text-align` (left-esquerda/ right- direita/ center-centralizado/ justify-justificado).

**Mexer a 1º palavra:** `text-indent` (30px)

**Espaçamento entre linhas:** `line-height`: 2em;

## Seletores

**Seletores personalizados:**

**ID:** `<h1 id="principal">` e no css `h1#principal{}` ou `#principal{}` // ID para personalizar apenas 1 texto

**Class:** `<h1 class="basico">` e no css `.basico{}` // Class para personalizar varios textos da mesma classe

*Dicas: com essa class posso destacar só uma palavra, por exemplo uma palavra "destaque" posso colocar a tag `<span class="destaque">` //span seria uma tag só para mudar as coisas*

**OBS:**

Consegue aplicar mais de uma classe em um elemento! É só separar por espaço

Consegue colocar um id e uma classe! Mas id sobrepõe.

## Pseudo-Classes e Pseudo-Elementos

**Pseudo-classes:** símbolo ":" relacionado a um elemento ou uma classe.

Existe varias (*visited- quando visita um site; active- quando clicar no link; hover - quando passar o mouse por cima acontece algo;*)

*EX: quero deixar um texto invisível faz:*

```
div>p{display:none;}
```

*Visível quando passa o mouse:*

```
div:hover>p{display:block;}
```

No CSS existe a pseudo-classe :root{ que guarda os elementos, tipo --cor0;--cor01; fonte...) para usar durante o desenvolvimento, só apertar CTRL + ESPAÇO

```
:root{
  --cor0: #c5ebd6;
  --cor1: #83e1ad;
  --cor2: #3ddc84;
  --cor3: #2fa866;
  --cor4: #1a5c37;
  --cor5: #063d1e;

  --fonte-padrão: Arial, Verdana, Helvetica, Sans-serif;
  --fonte-destaque: 'Bebas Neue', sans-serif;
  --fonte-android: 'android', cursive;
}

body{
  background-color: var(--cor0);
  font-family: var(--fonte-destaque);
}
```

**Pseudo-elemento:** símbolo "::" pode usar para indicar elementos

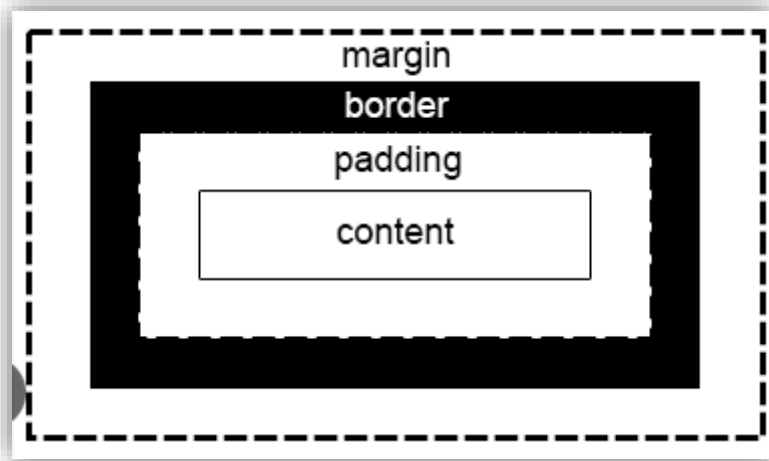
*EX: indicando que é um link: [www.google.com](http://www.google.com){LINK}*

*Usamos para o valor que aparece dps do link:> a::after{content}*

*Usamos para o valor que aparece antes do link:> a::before{content}*

## Caixas

**Caixas:** Toda caixa, tem a "Parte de Conteúdo = **content**", tem uma "**Border**", dessa borda para dentro, temos o "**Padding**", da borda para fora temos o "**Margin**" e, ainda dentro do margin e fora da borda, temos a área do "**Outline**".



**Altura:** Height

**Largura da caixa:** Width

**Borda:** Border 0px solid red (*largura, estilo, cor*)

*//Estilo: solid-solid; dashed- tracejada; dotted- pontilhada; double- dupla; groove- 3d//*

**Entre a borda e o texto:** Padding 15px 10px 15px 10px (*sent. horario*) ou colocar lados iguais só 10px ou colocar 15px 10px -cima baixo 15 e esquerda direita 10

**Margens de um elemento:** Margin:(*auto-centraliza*) 15px 10px 15px 10px (*sent. horario*) ou colocar lados iguais só 10px ou colocar 10px auto 10px auto -cima baixo 10 e centralizado

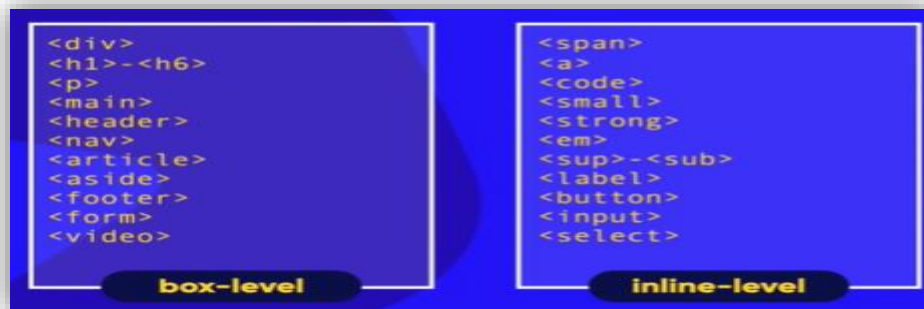
**Dentro da margem:** Outline 0px solid red (*largura, estilo , cor*)

*Dica: Geralmente quando cria site a margin e o padding fica atrapalhando!*

*Defina uma regra geral `*{margin:0px; padding:0px}`*

**Tipos de Caixa:** box-level e inline-level





É possível mudar a característica só colocando `{display: block/inline}`

### Sombras na caixa

**Sombras:** `box-shadow 1px 1px 1px black` (*horizontal, vertical, blur e cor-recomendado preto 50% transparencia*)

**Bordas decoradas:** `border-radius 15px 10px 15px 10px` (*top left, top right, bottom right, bottom left*) só 1 parametro ele repete para todos, se eu colocar 2 parametros ele faz superior esquerdo e inferior direito (1ª parametro) e superior direito e inferior direito (2ª parametro). EX: `border-radius: 10px 20px`

### Personalizar UL ou OL

Quando criamos uma lista **dentro de um aside** por exemplo, geralmente as "bolinhas" da lista irão ficar para fora do elemento pai (aside)... Para resolver isso "`list-style-position`" com o valor "`inside`" (por padrão marcadores de listas são posicionados com o valor "`outside`").

**Colocar coluna:** "`Columns`"

**Mudar o estilo:** `list-style-type: '\codigo do emoji'`; (geralmente fica sem espaço use o emoji `'\00A0` ou `'\0020` após o emoji selecionado)

```
aside>ul{
  list-style-position: inside;
  list-style-type: '\2714\00a0';
  columns: 2;
}
```

### Personalizar Video do YT

Coloque o vídeo dentro de uma div e faz a classe="vídeo"

**Mudar a posição:** `Position`: `static`, `relative`, `absolute`, `fixed` e `sticky`.

*Geralmente o relative vai na div.video {position: relative;}*

*Na `div.video>iframe{position:absolute}`*

**Tirar a barra preta do youtube:** `Padding-bottom` (varia de vídeo para vídeo)

"Position absolute" vem ACOMPANHADA e é usada em conjunto com quatro outras propriedades auxiliares, que são:

`top` - (topo)

`bottom` - (baixo)

`right` - (direita)

`left` - (esquerda)

```
div.video{
  position: relative;
  background-color: var(--cor5);
  margin: 0px -20px 30px -20px;
  padding: 20px;
  padding-bottom: 56.5%;
}

div.video>iframe{
  position: absolute;
  top: 5%;
  left: 5%;
  width: 90%;
  height: 90%;
}
```

## Responsividade

1º Limitar o tamanho mínimo e máximo da tela.

Usa-se no conteúdo **Main!**

O `Min-Width:300px`; (para deixar a tela no mínimo 300px) --- celulares antigos tem essa tela

O `Max-width:1000px` (para deixar a tela no máximo 1000px)

Colocar a `Margin: auto` (para centralizar a tela)

Mas isso não vai “quebrar a imagem” então precisa colocar um código chamado:

`Img{width:100%;}`

Por padrão imagem não é bloco, então se quiser centralizar uma imagem você coloca `display:block` e `margin:auto`

*Dicas:*

*Se o assunto principal tiver muito em cima da “cabeça” do site, pode afastar usando o `padding`!*

*Se o conteúdo principal tiver muito próximo do rodapé, posso criar uma margin-bottom para separar!*

*Para colocar sombra no texto utiliza: text-shadow: 2px 2px 0px black (igual a da caixa).*

*Se eu quiser colocar um efeito por exemplo quando passa o mouse por cima de algo e ele mudar a cor coloco a Transition-duration:0.5s;*

### Imagens de fundo

**Colocar imagem de fundo:** `background-image: url` ('caminho da imagem/nomedoarquivo) – CTRL+ESPAÇO da o caminho; \*pode ser url mesmo!

**Mexe no tamanho da imagem:** `Background-size` (auto, contain, cover)

- *Auto - Esse é o valor padrão*
- *Contain - Muda o tamanho da imagem para que ela sempre seja totalmente exibida na tela, sem nenhum corte. (com distorção)*
- *Cover - Redimensiona a imagem para que ela cubra todo o contêiner, mesmo que para isso ocorram alguns eventuais cortes. (não tem distorção)\*\*.*

**Quanto repete a imagem:** `background-repeat:` (no-repeat, repeat, repeat-x, repeat-y)

- *No-repeat; -> sem repetição (apenas 1 imagem)*
- *Repeat; -> repetir normalmente*
- *Repeat-x; -> repetição na horizontal*
- *Repeat-y-> repetição na vertical*

Ponto de partida é sempre no início da pagina! Mas tem como alterar.

**Mudar de posição da imagem:** `background-position:`

left top	center top	right top
left center	center center	right center
left bottom	center bottom	right bottom



**Não quebrar a imagem:** **Background-attachment:** *scroll* (rola junto com o conteúdo) e *fixed* (imagem fixa)

*\*\*Para efeito Parallax = fixed*

**Tem como fazer tudo junto de uma vez, seguindo a ordem:**

**background:** background-color>background-image>background-position>background-repeat>background-attachment

*Ex: background: black url(imagens/wall001.jpg) center center no-repeat fixed;*

### Centralização vertical

Se for centralizar algo verticalmente precisar ter 2 div (ou seção)

Que seja **“container”** e **“conteúdo”**.

O **container** tem que ter uma **position: relative;** (pode ser div id=”fora”)

O **conteúdo** tem que ter uma **position: absolute;** (pode ser div id=”dentro”)

Quando coloca posicionamento absoluto eu posso colocar 4 propriedades: (vamos usar 2)

**“Left” = 50%**

**“Top” = 50%**

Só que isso só vai centralizar o conteúdo no centro!

Tenho que usar **transform: translate(-50%, -50%)**

### **\*Transform:**

- *Translate - A função de translate serve para mover o elemento para os lados ou para cima/baixo.*
- *Rotate - utilizada para rotacionar o elemento, de 0 a 360 graus.*
- *Scale - Utilizado para redimensionar os elementos, tendo como escala padrão o valor "1". Valores maiores que 1, aumentarão o tamanho do elemento, e valores menores que 1 diminuirão sua escala.*
- *Skew - É utilizado para "distorcer" os elementos, alterando os ângulos.*

## Tabelas

### Alinhamento horizontal e vertical da tabela

**Horizontal:** pegar o td e colocar `text-align: center;` (left/center/right)

**Vertical:** pegar o td e colocar `vertical-align: Middle;` (top/Middle/bottom)

### Efeito zebrado em tabelas

Colocar uma subclasse chamada `nth-child` (2n) por exemplo que em 2 em 2 ele faz

Ou par e ímpar (`odd` -ímpar / `even` -par)

`Tbody>tr:nth-child ( )`

```
tbody > tr:nth-child(odd) { /* 2n-1 */
  background-color: white;
}

tbody > tr:nth-child(even) { /* 2n */
  background-color: lightgray;
}
```

Se eu quiser que o título da tabela por exemplo fique fixo no site (para quando rolar ele ficar mostrando oq é)

Coloco a table com `position: relative`

Colo o `thead>tr>th` com `position: sticky (colado) com top -1px`

### Tabela responsiva

Eu seleciono a tabela inteira `<table>` até `</table>` e **envelopo com uma div com `id container`** e coloco estilo da div com `overflow-x:auto` e horizontalmente já arruma!

- **overflow-x** - A propriedade CSS "overflow-x" é usada para definir o que deve ser mostrado quando o conteúdo transborda dos extremos esquerdo e direito de um elemento em bloco (eixo horizontal).
- **overflow-y** - A propriedade CSS "overflow-y" define o que deve ser mostrado quando o conteúdo transborda dos extremos superior e inferior de um elemento em bloco (eixo vertical).
- **overflow** - A propriedade CSS "overflow" é uma abreviação para as propriedades "overflow-x" e "overflow-y" em uma única declaração.

As propriedades overflow aceitam alguns valores, dentre os principais:

- **visible** - O valor "visible" exibe o conteúdo que exceder o tamanho do elemento fora da área designada para ele, (valor padrão).
- **hidden** - O valor "hidden" simplesmente corta o conteúdo que exceder o tamanho do elemento.
- **scroll** - O valor "scroll" exibe "a todo momento" barras de rolagem para permitir que o usuário role o conteúdo que exceder o tamanho do elemento.
- **auto** - O valor "auto" exibe barras de rolagem somente quando necessário (largura ou altura insuficientes)

## Media Queries

Criar estilos de css para cada tipo de tela.

- **All**= para todos os dispositivos independente da característica;
- **Screen**= Para tela; \*utiliza com media-features (característica de tela)
- **Print**= Para impressão;

\*Mediafeatures são escritos entre parênteses então no media=screen and (...)

...=Orientation:portrait (retrato)

...=Orientation:landscape (paisagem)

...=(min-width:... ) and (max-width:...)

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Testando o Media Query</title>
7   <link rel="stylesheet" href="estilo-geral.css" media="all">
8   <link rel="stylesheet" href="small-screen.css" media="screen and (max-width: 600px)">
9   <link rel="stylesheet" href="phone.css" media="screen and (min-width: 600px) and (max-width: 768px)">
10  <link rel="stylesheet" href="tablet.css" media="screen and (min-width: 768px) and (max-width: 992px)">
11  <link rel="stylesheet" href="pc.css" media="screen and (min-width: 992px) and (max-width: 1200px)">
12  <link rel="stylesheet" href="tv.css" media="screen and (min-width: 1200px)">
13  <link rel="stylesheet" href="print-portrait.css" media="print and (orientation: portrait)">
14  <link rel="stylesheet" href="print-landscape.css" media="print and (orientation: landscape)">
15 </head>
```

Note que agora, além da configuração geral (linha 7) temos configurações especiais para telas pequenas (linha 8), smartphones (linha 9), tablets (linha 10), PCs desktop/notebooks (linha 11) e TVs/telas gigantes (linha 12) com tudo adaptável. Também adicionamos configurações específicas para impressoras no modo retrato (folha em pé, linha 13) e paisagem (folha deitada, linha 14).

```

/* Typical Device Breakpoints
Pequenas telas: até 600px
Celular: de 600px até 768px
Tablet: 768px até 992px
Desktop: 992px até 1200px
Grandes telas: acima de 1200px
*/

```

**OBS:** em impressão não dá para usar background-color e etc...

### Menu responsivo (ver)

Coloca no head: `<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />`

E aonde quero colocar o “hambúrguer”: `<span class="material-symbols-outlined">menu</span>`

Para aparecer o menu ao clicar no “hambúrguer” posso colocar um JS com função: `clickmenu()` e de id ao menu.

Logo em baixo do main pode criar um script para fazer a função `clickmenu`

```

<header>
  <h1>Meu site</h1>
  <span class="material-symbols-outlined" onclick="clickmenu()">menu</span>
  <menu id="itens">
    <ul>

```

```

</main>
<script>
  function clickmenu() {
    if(itens.style.display == 'block') {
      itens.style.display = 'none'
    } else{
      itens.style.display='block'
    }
  }
</script>

```