

Curso em Vídeo HTML5 e CSS:

1 Módulo:

Neste módulo aprendemos o que é o **HTML5** e o básico do **CSS** e suas vertentes como tags, estruturas e desenvolvimento.

HTML5 é a quinta e última versão do HTML, sigla de Hypertext Markup Language, uma das primeiras e principais linguagens de marcação da web. O HTML serve, basicamente, para o desenvolvimento de páginas que contenham conteúdos simples, como textos, imagens, áudio, vídeos e etc.

O HTML é responsável pelo conteúdo do nosso website, como o próprio significado da linguagem ele é uma linguagem de marcação de textos tendo como sua estrutura base:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Meu primeiro Exercício</title>
</head>
<body>
  <h1>Olá, Mundo!</h1>
  <hr>
  <p>Esse é o meu primeiro exercício e eu estou muito
feliz</p>
  <p> Esse é um momento único estou criando meu primeiro
site</p>
</body>
</html>
```

Estrutura do código

O **HTML5** tem o seu código é dividido em duas partes a primeira é toda parte de configuração chamada de **HEAD** onde a configuração do nosso site está escrito, como também possíveis favicon, e também conectamos o **CSS** no futuro esta parte está escrita assim:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Meu primeiro Exercício</title>
</head>
```

A segunda parte do nosso site é o **BODY** é nela que estará escrito todo conteúdo do nosso site como cabeçalho, título, textos e etc.

```
<body>
  <h1>Olá, Mundo!</h1>
  <hr>
  <p>Esse é o meu primeiro exercício e eu estou muito
feliz</p>
  <p> Esse é um momento único estou criando meu primeiro
site</p>
</body>
```

Tags

Dentro da parte de body onde será escrito nosso código utilizamos tags que são escritas da seguinte forma:

```
<h1>Olá, Mundo!</h1>
```

Sempre respeitando a sintaxe determinada por **<TAG> CONTEÚDO </TAG>** existem diversas tag aqui irei citar algumas delas para melhor compreensão:

Entre os textos queremos colocar parágrafos ou nosso conteúdo com isso podemos utilizar a tag **<p>** por exemplo:

```
<body>
  <h1>Olá, Mundo!</h1>
  <hr>
  <p>Esse é o meu primeiro exercício e eu estou muito
feliz</p>
  <p> Esse e um momento único estou criando meu primeiro
site</p>
</body>
```

Lembrando que cada **<p>** significa uma linha então em quando não houver um quebra ou fechamento do **<p>**, ele seguirá a mesma linha

Se você quiser um espaço entre as linhas desta

Forma! Você utiliza, a tag **
** desta forma:

```
<body>
  <h1>Parágrafo e quebras de linha</h1>
  <hr>
  <p>Você pode escrever um parágrafo de qualquer jeito.
    Basta colocar tudo no meio do par de tags p e /p
  </p>
  <br>
  <p>Se precisar quebrar o texto em algum lugar <br>
especifico é só usar a tag br</p>
```

Emojis e Caracteres especiais

Como já citei anteriormente o HTML é uma linguagem de marcação e com isso temos algumas features junto com alguns caracteres que podemos utilizar:

Símbolo	Descrição	Entitiy
	<i>Espaço em branco</i>	
®	<i>Marca registrada</i>	®
©	<i>Copyright</i>	©
™	<i>Trade Mark</i>	™
€	<i>Euro</i>	€
£	<i>Libra (pound)</i>	£
¥	<i>Yen</i>	¥
¢	<i>Cent</i>	¢
∅	<i>Vazio</i>	∅
Σ	<i>Soma</i>	∑
Δ	<i>Delta</i>	Δ
←	<i>Seta esquerda</i>	←
↑	<i>Seta acima</i>	↑
→	<i>Seta direita</i>	→
↓	<i>Seta baixo</i>	↓

Seguindo estas tags acima.

Também podemos inserir emojis utilizando seus códigos através de um site chamado emoji grama, utilizando o código desta forma:

```
<p>Vamos adicionar emojis  
    &#x1F596;  
  
</p>
```

Precisamos apenas do seu código

Imagens e Favicons

Favicons são as imagens que aparecem ao lado do título do nosso site podemos aplicar um favicon na parte de HEAD do nosso código utilizando da linha:

```
<link rel="shortcut icon"
href="Iconscity-Flags-Brasil.256.png" type="image/x-icon">
```

Ficando assim:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link rel="shortcut icon"
href="Iconscity-Flags-Brasil.256.png" type="image/x-icon">
  <title>Colocar um icon no meu site</title>
</head>
```

E podemos também aplicar imagens em nosso site da seguinte maneira:

```

```

Colocando isso no BODY ficando assim:

```
<body>
  <h1>Este é o site de teste de mudar os icons do meu
site</h1>

  <p>Como homenagem ao hamilton utilizarei a imagem do
brasil como icon</p>
  
</body>
```

Podemos colocar qualquer imagem mas devemos nos atentar ao seu tamanho e peso

Hierarquia de títulos:

Podemos utilizar de um recurso para escrevemos parágrafos de maneira mais rápida esse comando se chama **Lorem** quando executa-lo ele terá um texto aleatório assim podemos ver a formatação do site antes de aplicar nosso site em definitivo:

```
<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Doloremque blanditiis neque eligendi fugit, saepe quia molestias at quidem maxime pariatur, com sequuntur a, voluptatem soluta praesentium consectetur corporis enim itaque culpa. </p>
```

```
<h1>Olá, Mundo!</h1> - Ele e o título do nosso texto
```

<h1> Tem uma peculiaridade existem h1 a h5 onde eles seguem ordem de níveis onde o primeiro e o título e o resto são subtítulos.

Podemos definir isso de uma maneira bem legal comparando nosso código e o site executado:

```
index.html > html
2  <html lang="pt-br">
9  <body>
10 <h1>Entendendo Portugal</h1>
11
12 <h2>1. Introdução</h2>
13 <br>
14 <h3>1.1 Identificação da temática</h3>
15 <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Doloremque blanditiis neque eligendi fugit, saepe quia molestias at quidem maxime pariatur, consequuntur a, voluptatem soluta praesentium consectetur corporis enim itaque culpa. </p>
16 <h3>1.2 Caratér excecional de caso de estudo</h3>
17 <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Repellat beatae sapiente saepe porro. Corrupti cupiditate fuga asperiores accusamus dolore voluptas optio quia autem possimus. Necessitatibus molestias placeat porro laborum vel! </p>
18 <br>
19
20 <h2>2. Ordenamento do território em Portugal</h2>
21 <br>
22 <h3>2.1 Os instrumentos de planeamento territorial</h3>
23 <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolorum a laborum porro minus beatae tempore minima labore. Illum similique vitae, accusantium at adipisci voluptates nulla, sed, dignissimos quae aperiam architecto?</p>
24 <h4>2.2 Qualquer loucura</h4>
25 <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore adipisci praesentium voluptas porro, unde sed enim. Id ipsum quisquam quidem ratione magni corporis quis, tempora accusamus sunt consectetur aperiam perspiciatis. </p>
26 </body>
27 </html>
```

Entendendo Portugal

1. Introdução

1.1 Identificação da temática

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Doloremque blanditiis neque eligendi fugit, saepe quia molestias at quidem maxime pariatur, consequuntur a, voluptatem soluta praesentium consectetur corporis enim itaque culpa.

1.2 Caratér excecional de caso de estudo

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Repellat beatae sapiente saepe porro. Corrupti cupiditate fuga asperiores accusamus dolore voluptas optio quia autem possimus. Necessitatibus molestias placeat porro laborum vel!

2. Ordenamento do território em Portugal

2.1 Os instrumentos de planeamento territorial

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolorum a laborum porro minus beatae tempore minima labore. Illum similique vitae, accusantium at adipisci voluptates nulla, sed, dignissimos quae aperiam architecto?

2.2 Qualquer loucura

Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore adipisci praesentium voluptas porro, unde sed enim. Id ipsum quisquam quidem ratione magni corporis quis, tempora accusamus sunt consectetur aperiam perspiciatis.

Formatação dos textos:

Como já foi dito o HTML é uma linguagem de marcação de texto e tem suas propriedades como tal

Para deixar nosso texto em *itálico* podemos utilizar:

```
<P>Para itálico <i>podemos usar a tag i </i> </P>
```

Para deixar nosso texto em **negrito** podemos utilizar:

```
<p>Mas o correto a se utilizar em a opção de <strong>Desta  
forma</strong></p>
```

Textos com **marcação** podemos utilizar de:

```
<p>Podemos usar a tag <mark>Mark</mark></p>
```

Textos pequenos:

```
<p>Para criamos textos pequenos podemos utilizar a tag  
<small>small </small></p>
```

Para textos deletados

```
<p>Para criamos um texto deletado podemos utilizar a tag  
<del>del</del></p>
```

Para textos sublinhados:

```
<p>Para texto sublinhado podemos utilizar a tag  
<ins>ins</ins></p>
```

Para textos sobrescritos:

```
<p>Para textos sobrescrito podemos utilizar a tag sup desta  
forma x <sup>20</sup>+3 </p>
```

Para textos sobrescritos:

```
<p>Para inserir textos sobrescritos podemos utilizar a tag sub  
desta forma H<sub>2</sub>O</p>
```

Listas

Esses dois tipos de listas existe a **OL** e a **UL** qual a diferença entre elas:

A HTML chama de ordered lists todas aquelas listas onde a ordem dos itens é algo muito importante. Um passo-a-passo para criar um bolo, uma lista de aprovados no vestibular e uma lista com os carros mais caros do mundo são exemplos de listas ordenadas.

Para criar uma ordered list, vamos usar a tag:

```
<ol type="1">
  <li>Acordar</li>
  <li>Falar com Ka</li>
  <li>Vê o anel</li>
</ol>
```

A tag possui um parâmetro type, onde configuramos o tipo de marcador da lista atual. As opções de valores para esse parâmetro são:

- 1 - Valor padrão. Criar listas numeradas. Ex: 1, 2, 3, 4, ...
- A - Cria listas alfabéticas em maiúsculas. Ex: A, B, C, D, ...
- a - Cria listas alfabéticas em minúsculas. Ex: a, b, c, d, ...
- I - Criar listas com algarismos romanos em maiúsculas. Ex: I, II, III, IV, ...
- i - Criar listas com algarismos romanos em minúsculas. Ex: i, ii, iii, iv, ...

Já para **UL** sabemos unordered lists, também chamadas de listas com marcadores, que são aquelas onde a ordem dos itens não influenciará no significado da lista. Ela é apenas uma ótima maneira para organizar os itens que não apresentam uma classificação necessariamente. Para criar uma unordered list, vamos usar a tag

```
<ul type="square">
  <li>Pão
  <li>Renzo
  <li>Alan Ruiz
  <li>F. Domiinguez
</ul>
```


Podemos também juntar uma lista dentro da outra e assim sucessivamente

```
<ol>

  <li>antigas</li>

  <ol>

    <li>Python </li>

    <li>PHP</li>

  </ol>

  <li>Novas</li>

  <ol>

    <li>Java </li>

    <li>Ruby </li>

  </ol>

</ol>
```

Links

Podemos também utilizar de links para enaltecer nossos sites usando como exemplo:

```
<p>Você pode vê o vídeo do melhor do mundo <a  
href="https://www.youtube.com/watch?v=tKb5-5yWeCA">SIR  
LEWIS HAMILTON</a></p>
```

Esse exemplo foi feito de sites externos como Youtube, Instagram e etc mas podemos também fazer o mesmo para links internos da nossa máquina desta forma

```
<p>Você pode acessar a minha outra pagina e so clicar <a  
href="002.html"> AQUI</a></p>
```

Podemos também ter no nosso site link para downloads de documentos

```
<ul>

    <li>Baixar meu currículo <a
href="Livros/Curriculo At Elydeivison Cesar.pdf"
download="Curriculum At Elydeivison Cesar.pdf"
type="application/pdf"> Clique aqui</a></li>

</ul>
```

Para reproduzirmos áudios podemos utilizar da:

```
<audio preload="auto"autoplay controls>

    <source src="media/aiiiza.mp3"
type="audio/mpeg">

</audio>
```

Podemos usar esta formatação para enviar email cadastrado:

```
<a href="mailto:eng,elydeivison@gmail.com">Envie um
E-mail</a>
```

Inserindo vídeo hospedados

Podemos realizar vídeos hospedados em sites como youtube para isso basta buscarmos o código HTML do vídeo ao qual o próprio YT nos disponibiliza e escrever da seguinte forma:

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/5MCpTa0yYVM?si=c1D-MSm
0mKdkZJJJC" title="YouTube video player" frameborder="0"
allow="accelerometer; autoplay; clipboard-write;
```

```
encrypted-media; gyroscope; picture-in-picture;  
web-share" allowfullscreen></iframe>
```

Isso escrito no H1

CSS

É um mecanismo para adicionar estilos a uma página web, aplicado diretamente nas tags HTML ou ficar contido dentro das tags `<style>`. Também é possível adicionar estilos adicionando um link para um arquivo CSS que contém os estilos.

Em resumo o CSS é o estilo do nosso site

para conectarmos o CSS utilizamos da seguinte maneira dentro da parte HEAD do nosso código podemos adicionar a linha:

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

ficando assim nosso código

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
  <title>Aprendendo a usar CSS</title>  
  <link rel="stylesheet" href="style.css">  
</head>
```

Este link é redirecionado para outro arquivo que se chama ALguma Coisa.css

Por base o código CSS tem essa base aqui:

```
body {  
  background-color: aliceblue;
```

```
font-family: 'Franklin Gothic Medium', 'Arial Narrow',  
Arial, sans-serif;  
font-size: medium;  
}  
  
h1 {  
color: aquamarine;  
background-color: black;  
}  
  
h2 {  
color: blueviolet;  
}  
  
p{  
text-align: justify;  
}
```

Como podemos ver aqui configuramos as cores, fonte e tudo que for ligada a estilo do nosso site.

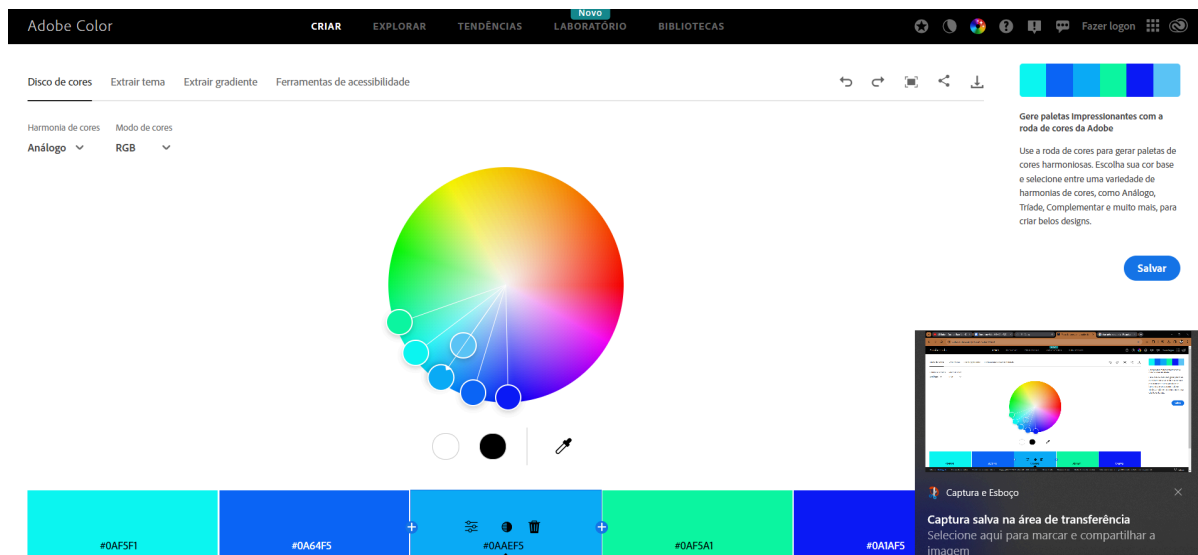
2 Módulo:

Cores:

Neste início de módulo aprendemos um pouco sobre pensamento humano sobre cores e o quanto que ela influencia na criação do nosso website sendo assim mais harmônico segue a tabela para consulta:

Cor	Associada a	Usar em
vermelho	amor, emoção, energia, raiva, perigo	comida, moda, entretenimento, serviços de emergência
amarelo	felicidade, alegria, otimismo, covardia	dar luz, dar calma e chamar atenção
laranja	divertimento, ambição, calor, cautela	comércio eletrônico, entretenimento, call center
verde	saúde, natureza, dinheiro, sorte, inveja	relaxamento, turismo, serviços financeiros, meio ambiente
azul	competência, sabedoria, calma, frio	tecnologia, medicina, governo
roxo	criatividade, poder, sabedoria, mistério	produtos de beleza, ioga, espiritualidade
marrom	terra, robustez, estabilidade, amizade	alimentação, imobiliária, finanças
preto	elegância, autoridade, mistério, morte	luxo, moda, marketing, cosméticos

Gustavo nos passou algumas ferramentas bem úteis para que pudéssemos criar sites mais harmônicos e de fácil utilização dos usuários. Como por exemplo:



<https://color.adobe.com/pt/create/color-wheel>

Isso nos permite descobrir paletas de cores mais harmônicas para o nosso site de maneira automática e gratuita.

Como podemos inserir essas cores em nosso código, existem diversos códigos e maneiras de demonstrar cores no nosso código alguma delas são:

```
background-color: white;
```

```
color: #000000;
```

Lembrando que todas as vezes que falamos sobre cores ou estilos estamos falando de CSS então deve estar incluso no style.css;

As cores podem ser definidas por nomes,, **#0000** ou **RGB (0 , 0 ,0)**

Podemos também fazer um **degradê** para as nossas cores utilizando um código.

O **degradê** tem uma funcionalidade bem simples ele é ativado pelo prefixo background-image: linear-gradient (Aqui dentro você deve afirmar pra onde você quer que o seu degradê siga se e pra cima pra baixo para a esquerda ou direita, logo após você coloca a cor de início e a cor de final como por exemplo:)

```
background-image: linear-gradient(to right, #042940, #005C53);
```

Temos o mesmo funcionamento para um degradê só que em círculo ficando assim:

```
background-image: radial-gradient(to right,  
#042940, #005C53);
```

Fonts:

Agora vamos falar sobre fontes,, primeiro podemos utilizar dois tipos de fontes aquelas que já possuem no nosso sistema sendo assim quando invocamos o código:

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

O VSCODE ira redirecionar para todas as fontes já existentes por padrão no sistema, porém podemos procurar qualquer fonte na internet sem precisar que ela esteja no nosso computador utilizando o Google Fonts.

‘Para isso podemos entrar no0 Google Fonts> Procurar a fonte> Clicar em Obter Fonte > Obtenha Código de incorporação > @import > E vamos copiar o código desta maneira

```
@import  
url('https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,100;0,200;0,300;0,4
```

```
00;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&family=Madimi+One&family=Noto+Sans+Khojki&display=swap')
```

Logo após vamos no nosso código, na parte de CSS e colamos o código acima, depois disso é só Escolher onde queremos aplicar nossa fonte como por exemplo:

```
@import
url('https://fonts.googleapis.com/css2?family=Kanit: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&family=Madimi+On
e&family=Noto+Sans+Khojki&display=swap');

body{
  font-family: 'Madimi One' , sans-serif;
```

É de suma importância escrevermos o nome da fonte no font-family de maneira correta para assim o sistema entender o que queremos.

Também podemos utilizar fontes instaladas no nosso próprio computador, para isso podemos baixar uma fonte como exemplo alguma do google fonts, baixar seu arquivo ou instalar e utilizar desta maneira no código.

```
@font-face {
  font-family:'Love' ;
  src: url('love larry.otf')  format('opentype');
}
```

Alguns pontos a serem ditos:

- O arquivo deve estar dentro da nossa pasta de projeto
- Devemos escrever o nome exatamente como está escrito
- Escrever seu format
- No font Family Podemos utilizar abreviações para facilitar
- Esta abreviação será a forma que utilizaremos até o fim do código como se fosse uma class

Tipos de Format:

OpenType (OTF)

TrueType (TTF)

As fontes seguem um padrão:

```
body{
  font-family: 'Madimi One' , sans-serif;
  font-size: 16px;
  font-weight: 400;
  font-style: italic;
}
```

font-family: É a família que nossa fonte participa como exemplo Arial.

font-size: É o seu tamanho no texto, no caso o tamanho da letra

font-weight: É a espessura do texto

font-style: é o estilo do texto como exemplo Itálico, negrito e etc.

Existe uma maneira de encurtamos esse código fazendo essa ordem

```
Posso utilizar um shorthand da seguinte maneira::after
  font-style -> font-weeight -> font-size -> font-family

font-family: 'Courier New', Courier, monospace;
font-size: 16px;
font-weight: 400;
font-style: italic;
```

Ficando assim:

```
font: italic 400 16px 'Courier New', Courier, monospace;
```

Também possuímos formas de alinhar nosso texto que são:

text-align: left; Para alinhar nosso texto a esquerda

text-align: right; Para alinhar nosso texto a direita

text-align: center; Para alinhar nosso texto no centro

text-align: justify; Para alinhar nosso texto de acordo com a página

Seletores Personalizados:

Aqui descobrimos uma ferramenta onde podemos organizar nosso código de maneira mais coerente e assim conseguindo resolver certas particularidades de maneira individual, muito útil para coisas específicas no CSS, um exemplo de seletor e o código **CLASS** e **ID**:

```
<h1 class="principal"> Criando Sites</h1>
```

```
<h1 id="principal"> Criando Sites</h1>
```


Qual a sua diferença? Eles possuem particularidades bem parecidas, porém o id irá identificar um seletor único em toda a página enquanto o class pode ser vários. Como falei, a principal funcionalidade destes seletores é utilizá-los no CSS, porém para isso devemos respeitar algumas regras.

Para **ID** sempre utilizaremos #

Já para **CLASS** utilizaremos .

```
#principal {  
    background-color: white;  
    text-align: center;  
}  
  
.Intermediario {  
    color: black;  
    font-family: 'Times New Roman', Times, serif;  
}
```

Uma grande vantagem de utilizarmos CLASS é que se modificarmos uma vez a CLASS todos aqueles objetos/Textos que estejam dentro de uma class são formatados de maneira padrão, sendo assim facilitando muito nosso código CSS.

Pseudos-Classes

Elas são responsáveis para executar algumas funções dentro da classe de tal forma melhorar a experiência do usuário e deixar algo mais harmônico e também quebrar alguns bugs deixando assim nosso site mais responsivo.

Como exemplo:

```
<style>  
    body{  
        font-family: Arial, Helvetica, sans-serif;  
    }  
    a {  
        color: red;  
        text-decoration: none;  
        font-weight: bold;  
    }  
    a:visited {  
        color: darkred;  
    }
```

```
    }

    a:hover {
        color: black;
        text-decoration: underline;
    }

    a:active {
        color: yellow;
    }

</style>
</head>
```

Neste exemplo podemos ver 3 tipos de pseudos classes, que são:

a:visited- faz com que nosso link ganhe uma cor como no exemplo vermelho escuro

a:hover- Significa que toda vez que passarmos o mouse pelo texto ele ficará preto e e vai fazer uma linha underline

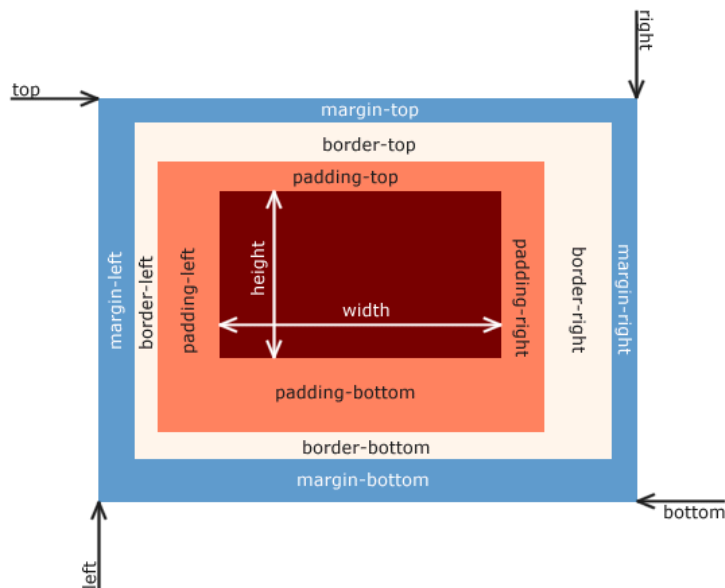
a:active- Significa que quando apertamos o botão de link ele ficará amarelo

Com isso podemos fazer diversas configurações e deixar nosso site de diversas maneiras, fazendo menus e etc

- Como vistos para pseudos classes precisamos utilizar o class: função para sim ele entender o que queremos

Box Model

Ou Modelo de caixa, este é um assunto extremamente importante de entendimento pois ele será significadamente responsável por tudo que vimos de designer em nosso site:



width- Largura
height- Altura
padding-Preenchimento
margin- Margem
border- Bordar

Como isso funciona na prática:

```
h1 {  
    background-color: lightgray;  
    height: 300px;  
    width: 300px;  
}
```

Aqui eu configurei a caixa de modo geral tanto de altura e largura eu quero que ela tenha.

Margin, existem algumas maneiras de configurarmos nossa margem a primeira e mais comum é

```
margin-top: 10px;  
margin-right: 10px;  
margin-bottom: 10px;  
margin-left: 10px;
```

Fazendo cada parte dela ou podemos simplificar de duas formas:

```
margin: 10px;
```

Caso todos os lados forem todos iguais ou:

```
margin: 10px 20px 10px 20px;
```

Sempre respeitando a ordem através do sentido de um relógio desta forma:

top> right> bottom > left ou por exemplo se possuímos lados iguais e parte de cima e baixo iguais podemos exemplificar também com:

```
margin: 10px 20px;
```

Uma função muito utilizada é a:

```
margin: auto;
```

Ele é responsável por centralizar seus blocos isso facilita bastante para ficar mais harmônico nosso site

A borda também tem suas propriedades que são:

```
border-width: 10px;  
border-style: solid;  
border-color: aqua;
```

Onde como já vimos antes:

border-width = Altura da borda

border-style= O estilo da nossa borda se será sólida, negrito e etc

border-color = a cor da nossa borda

Padding também segue as mesmas propriedades da Margin ficando assim:

```
padding-top: 10px;  
padding-right: 10px;  
padding-bottom: 10px;  
padding-left: 10px;
```

Como já disse ele segue a mesma propriedade da margin, sendo assim seu shorthand:

```
padding: 10px;
```

Já o OUTLINE é responsável por criar uma borda entre a borda e o conteúdo tendo esse código:

```
outline-width: 5px;
```

```
outline-style: dashed;  
outline-color: aquamarine;
```

Podendo ser simplificado desta forma:

```
outline: 5px dashed aquamarine;
```

TAGS Semânticas::

Como já vimos o HTML é uma linguagem de marcação de texto e com isso devemos seguir regras e uma delas é uma regra semântica, as tags elas foram criadas com o intuito de facilitar e deixar nosso código de maneira organizado e com ele temos diversas vantagens tais como uma melhor organização do HTML e CSS juntos, essas tags seguem ordens e significados

Header:

Cria áreas relativas a cabeçalhos. Pode ser o cabeçalho principal de um site ou até mesmo o cabeçalho de uma seção ou artigo. Normalmente inclui títulos

Pode conter também menus de navegação.

Nav

Define uma área que possui os links de navegação pela estrutura de páginas que vão compor o website. Um pode estar dentro de um <header>

Main

É um agrupador usado para delimitar o conteúdo principal do nosso site. Normalmente concentra as seções, artigos e conteúdos periféricos.

Section

Criar seções para sua página. Ela pode conter o conteúdo diretamente no seu corpo ou dividir os conteúdos em artigos com conteúdos específicos. Segundo a documentação oficial da W3C, “uma seção é um agrupamento temático de conteúdos, tipicamente com um cabeçalho”.

Article

Um artigo é um elemento que vai conter um conteúdo que pode ser lido de forma independente e dizem respeito a um mesmo assunto. Podemos usar um para delimitar um post de blog ou fórum, uma notícia, etc.

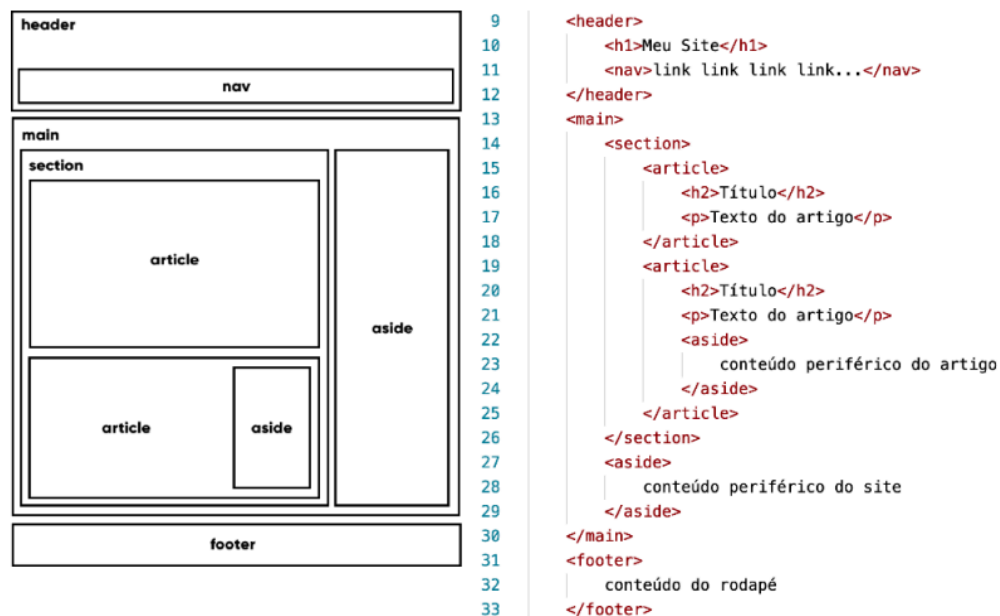
Aside

Delimita um conteúdo periférico e complementar ao conteúdo principal de um artigo ou seção. Normalmente um conteúdo está posicionado ao lado de um determinado texto ou até mesmo no meio dele, exatamente como fizemos no bloco de texto apresentado anteriormente, falando sobre “MÚLTIPLOS NÍVEIS”.

Footer

Criar um rodapé para o site inteiro, seção ou artigo. É um conteúdo que não faz parte diretamente do conteúdo nem é um conteúdo periférico (o que caracterizaria um), mas possui informações sobre autoria do conteúdo, links adicionais, mapa do site, documentos relacionados. A seguir, vou criar uma proposta de estrutura para um projeto de site. Não tome ela como a única possibilidade de criar o posicionamento de elementos de agrupamento semântico.

Segue aqui um bom exemplo do que disse:



Podemos também colocar sombras nas nossas caixas utilizando a seguinte função:

```
box-shadow: 1px 1px 1px black;
```

Funciona basicamente com deslocamento vertical, horizontal, opacidade e cor.

Podemos também fazer com que nossas bordas fiquem mais arredondadas usando o:

```
border-radius: 10px 10px 40px 40px ;
```

Tem o mesmo funcionamento do Padding e Margin no mapa de pixel.

Podemos também deixar nosso site adaptável para diversas telas utilizando essa configuração

```
min-width: 300px;
max-width: 1000px;
```

Isso pode ser aumentado ou baixado de acordo com o site

Para finalizarmos este módulo desenvolvemos um site juntos utilizando tudo isso que foi aprendido acima. O site está disponível aqui:

Módulo 3

Aqui começamos um novo módulo com foco em GIT, Github, Imagens de Fundo e Tabelas.

GIT

O **GIT** é uma ferramenta que funciona como um repositório local, basicamente o GIT funciona como um salvamento do nosso arquivo dentro da nossa máquina e através de alguns comandos ainda não aprendidos podemos invocar, upar, modificar arquivos na nossa máquina através do GIT.

Ele funciona com um APP já instalado.

GITHUB

O GITHUB é o irmão do GIT, os dois têm a mesma ideia porém o GitHub é um repositório online, ele funciona quase como fosse a nossa máquina sendo que online, através do GITHUB.com podemos enviar e deixar nossos projetos hospedados no seu site sem nos preocupar e levar o nosso computador pessoal em todos os lugares, de maneira online podemos mudar modificar ou até mesmo criar um novo projeto e deixá-lo hospedado no GITHUB.

Dentro do GITHUB podemos criar novos repositórios partindo da sua pasta raiz, sendo assim quando usarmos todas as pastas e arquivos dentro dela será upada junto.

Esses arquivos devem sempre estar em Atualização, para isso usamos uma ferramenta dentro do GITHUB desktop chamada COMMIT ela é responsável por atualizar nosso repositório remoto, fazendo com que não perdemos algo importante.

- Presta atenção ao commitar para não subir informações erradas ou não ter um backup do arquivo..

<https://github.com/DevElyShow> - Este é o meu GITHUB

O GITHUB também possui uma ferramenta muito interessante e importante para projetos básicos que é hospedar gratuitamente nosso projetos na internet para isso podemos fazer a seguinte coisa:

- Subir nosso repositório através do Github Desktop
- Dar um PUSH
- Ir até o site do GITHUB
- Entrar no repositório
- Ir at'a função de Settings
- Ir em PAGES
- Configurar desta forma

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://devellyshow.github.io/Projeto-Cordel/>

Last deployed by  DevElyShow yesterday

 Visit site

...


Build and deployment


Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the `main` branch. [Learn more about configuring the publishing source for your site.](#)

 main ▾

 / (root) ▾

Save

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.

[Learn more about deploying to GitHub Pages using custom workflows](#)

Logo após isso (o processo de upload pode demorar uns minutos) o site estará no ar e o GITHUB disponibiliza para você o link, compartilhar com amigos, empresas e etc

Como exemplo:

<https://devellyshow.github.io/Projeto-Cordel/>

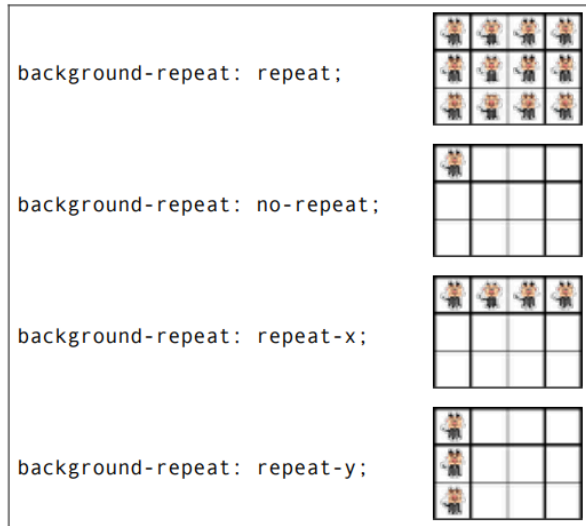
Imagens de Fundo

Agora aprenderemos uma maneira de colocar uma imagem de fundo no nosso site através da função `background-image`:

```
body{
  background-image: url(Images/wallpaper004.jpg);
  background-size: 100px 100px;
  background-repeat: no-repeat;
}
</style>
```

Podemos configurar dentro do nosso Style, essas funções responsáveis por colocar uma imagem de fundo em nosso site:

Background-image- Aqui podemos colocar uma imagem qualquer de fundo utilizando URL
Background-size: E a sua dimensão ao qual quero que apareça na tela
Background-repeat: É uma função para se quisermos repetir nossa imagem de diversas formas, tanto pra baixo, pra cima pro lado e etc.



Um exemplo de repeat.

Podemos também definir qual parte da imagem queremos que apareça em nosso site usando a seguinte referência:

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

No código:

```
div#q1 {  
    background-position: left top;  
}  
  
div#q2 {  
    background-position: left right;  
}  
  
div#q3 {  
    background-position: right center;  
}  
  
div#q4 {  
    background-position: right bottom;  
}
```

Podemos utilizar também dentro de Background-size:

- **auto (padrão):** a imagem de fundo será aplicada em seu tamanho original.
- **[length]px :** Redimensionar a largura da imagem e faz a altura se adaptar automaticamente. Podemos também informar as duas dimensões na sequência ou também usar valores percentuais.
- **cover:** Muda o tamanho da imagem para que ela seja sempre totalmente exibida na tela, sem nenhum corte.
- **contain:** Redimensiona a imagem para que ela cubra o contêiner, mesmo que para isso ocorram alguns eventuais cortes.

A outra propriedade é o attachment ela cuida da imagem de fundo com o resto do documento, principalmente se o conteúdo for maior do que a altura da página e seja necessário fazer uma rolagem vertical.

- **scroll (padrão):** a imagem de fundo vai rolar junto com o conteúdo.
- **fixed:** A imagem de fundo vai ficar fixada enquanto o conteúdo vai sendo rolado

Como ninguém gosta de linhas de código enorme podemos utilizar também algumas shorthands para melhorar nosso código da seguinte maneira:

```
/* height: 100vh;
   background-color: black;
   background-image: url(Images/wallpaper002.jpg);
   background-position: center center;
   background-repeat: no-repeat;
   background-size: cover;
   background-attachment: fixed;
   /* Shorthand
      Background-color > image > position > repeat >
attachment
      Background-size: cover
   */

   background: black url(Images/wallpaper002.jpg)
center center no-repeat cover fixed ;
   background-size: cover;
```

Pra finalizar o módulo nós fizemos um projeto de mini cordel, utilizando todas essas features aprendidas até aqui o site está disponibilizado em:

<https://devellyshow.github.io/Projeto-Cordel/>

E o repositório: <https://github.com/DevElyShow/Projeto-Cordel>

4 Módulo:

Dando agora início aos assuntos referentes ao 4 módulo do curso de CSS e HTML:

Neste capítulo tivemos uma introdução sobre iframes e seu uso, eles podem ser muito bem vindos para usos específicos no site porém hoje estão em desuso por ter suas limitações e não ser tão seguro, A cargo de curiosidade este é seu código:

```
<!--Tamanho Padrão 300x150-->
<h1>
    Testando iFrames
</h1>
<p>Acessando o site com iFrames desta forma
    <iframe src="https://www.instagram.com/_elyshow_"
frameborder="0" height="500"></iframe>

</p>
```

Logo após demos início ao assunto de formulário e suas propriedades, os formulários por características são descritos desta forma:

```
<h1> Meu primeiro Formulario </h1>
<form action="cadastro.php" method="get">
    <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>
```

Todo formulário é envelopado numa tag form e precisa primeiramente de um banco de dados para direcionar nossa resposta no caso o cadastro.php (iremos estudar isso no futuro). Logo após temos o nosso method que é dividido em 2 formas:

O GET é muito utilizado para acessar um recurso, uma informação, mas nunca para enviar dados como dados de um formulário, por exemplo, isso é responsabilidade do POST. Outras diferenças são que a informação POST não será salva no histórico do usuário, e as informações confidenciais (como senha, por exemplo) enviadas no formulário não serão visíveis na barra de URL (usando GET, elas serão visíveis no histórico e na barra de URL).

Logo após isso abrimos um <p> para definirmos o parágrafo de forma semântica, e o nosso formulário é envelopado numa tag label, onde precisamos definir nosso id, no caso desse exemplo utilizamos "nome"

Logo após usamos o input para definirmos qual será o valor daquele texto, existem diversos tipos de input, como email, senha, telefone, data de nascimento e etc. Cabe a nós definirmos qual o melhor pro nosso formulário.

Nele definindo a sua função de o id usado no for.

Logo após utilizando o label para também colocar nossa segunda etapa do formulário

Outro exemplo de input e o de botão de enviar definido desta forma:

```
<p><input type="submit" value="Enviar"></p>
```

Segue agora alguns exemplos de formulários:

Email

```
<p>
  <label for="iemail">E-mail</label>
  <input type="email" name="email" id="iemail"
required>
</p>
```

Telefone:

```
<p>
  <label for="itel">Telefone</label>
  <input type="tel" name="tel" id="itel"
autocomplete="tel" required pattern="^\(\d{2}\)\d{4},
5)-\d{4}$" required>
</p>
```

Senha:

```
  <label for="isen">Senha: </label>
  <input type="password" name="sen" id="isen"
required minlength="8">
```

Podemos fazer também botões de questionário desta forma utilizando a tag fieldset:

```
<fieldset>
  <legend>Sexo</legend>
  <input type="radio" name="Sexo" id="isexma"> <label
for="isexma">Masculino</label> <br>
  <input type="radio" name="Sexo" id="isexfem"> <label
for="isexfem">Feminino</label>
</fieldset>
```

Isso tudo dentro do FORM.

```
<fieldset>
  <legend>Esportes Favoritos</legend>
  <input type="checkbox" name="esbas" id="iesbas">
<label for="iesbas">Basquete</label><br>
  <input type="checkbox" name="esfut" id="iesfut">
<label for="iesfut">Futebol</label><br>
  <input type="checkbox" name="esfor" id="iesfor">
<label for="iesfor">Formula 1</label><br>
</fieldset>
```

Podemos também trabalhar com botões pre definidos, desta forma:

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

  <optgroup label="Região Sudeste">
    <option value="MG">Minas Gerais</option>
    <option value="RJ">Rio de Janeiro</option>
  </optgroup>
  <optgroup label="Região Nordeste">

    <option value="PE">Pernambuco</option>
    <option value="BA">Bahia</option>
  </optgroup>
</select>
```

Utilizando a tag optgroup, nós definimos grupos para nossas opções e isso fica evidenciado no botão

Trabalha com cores:

Exemplo de botões:

```
<p>
  <input type="submit" value="Enviar">
  <input type="reset" value="Limpar">
</p>
```

Cores

```
<p>
  <label for="icor">Cor:</label>
  <input type="color" name="Cor" id="icor"
value="#00ff00">
</p>
```

Nível de satisfação:

```
<p>
  <label for="inivel">Nível de Satisfação</label>
  <input type="range" name="Nivel" id="inivel"
min="1" max="5" value="3">
</p>
```

Upar arquivos como foto:

```
<p>
  <label for="ifoto">Foto de Perfil</label>
  <input type="file" name="Foto" id="ifoto">
</p>
```

Portanto os formulários são uma ferramenta bem legal principalmente com aqueles que trabalham com banco de dados e etc, temos diversas opções que podem ser trabalhadas e assim deixar da maneira que mais nos agrada.

Media-Query

Iremos agora falar sobre media-query, trecho muito importante principalmente para nós que visamos o desenvolvimento web, em resumo Media query, nada mais é do que uma estrutura do CSS que permite aplicar estilizações específicas para uma página web de acordo com certas condições, fazendo assim com esta página possa se adequar ao layout de tela em diferentes tamanhos e tipos de mídia.

Portanto é algo que devemos prestar atenção na construção do nosso código pois ela é a responsável pela responsividade do nosso site.

A media query é introduzida no código CSS desta forma:

Para impressão:

```
<link rel="stylesheet" href="estilo/impressora.css"
media="print">
```

Devemos escrever no nosso head um link falando que este link no caso, nosso papel será para impressão isso é muito comum para jornais como NY Times, escrevendo nosso código ficamos com isso

```
* {
  margin: 10px;
  padding: 10px;
}

html {
  font-family: 'Courier New', Courier, monospace;
  font-size: 1.5em;
  line-height: 1.5em;
}

menu {
  display: none;
}
```

Claro que isso é um exemplo devemos formatar isso de acordo a cada site e de maneira que deixe mais fluido pro usuário.

Da mesma forma que utilizamos um link podemos utilizar um link para um css normal como já vimos e estamos fazendo desde então:

```
<link rel="stylesheet" href="estilos/tela.css" media="screen">
```

E a partir deles que podemos e devemos fazer as adaptações de telas, podemos fazer css distinto para cada tela mas isso nós ira prejudicar e trazer muitos arquivos pro nosso projeto podemos resumir tudo em apenas um CSS intitulado de media-query desta forma:

```
<link rel="stylesheet" href="estilos/media-querie.css"
media="all">
```

***É IMPORTANTÍSSIMO O USO DE MEDIA="ALL"**

Usamos o **@MediaScreen** para definir nossa resolução

Com isso podemos criar um arquivo a parte CSS envolvendo todas as medias queries dividindo-as por resolução desta forma:

```
@media screen and (min-width:768px) and (max-width: 982px) {
/*TABLET*/
    body {
        background-image: url(../imagens/back-tablet.jpg);
    }

    img#fon {display: none;}
    img#tv {display: none;}
    img#tbl {display: block;}
    img#pc {display: none;}
    img#imp {display: none;}
}
```

Claro que dentro das nossa medias queries nós podemos configurar da maneira que nós desejarmos :

Resolução de cada aparelho:

```
/* Desktop */

@media (min-width: 1281px) {

    /* CSS */

}
```

```
/*Laptops */

@media (min-width: 1025px) and (max-width: 1280px) {

    /* CSS */

}
```

```
/* Tablets    ##Device = Tablets, Ipads (portrait)
##Screen = B/w 768px to 1024px*/

@media (min-width: 768px) and (max-width: 1024px) {

    /* CSS */

}
```

```
/*
##Device = Tablets, Ipads (landscape)
##Screen = B/w 768px to 1024px
*/
```

```
@media (min-width: 768px) and (max-width: 1024px) and
(orientation: landscape) {

    /* CSS */

}
```

```
/*
    ##Device = Low Resolution Tablets, Mobiles (Landscape)
    ##Screen = B/w 481px to 767px
*/

@media (min-width: 481px) and (max-width: 767px) {

    /* CSS */

}
```

```
/*
    ##Device = Most of the Smartphones Mobiles (Portrait)
    ##Screen = B/w 320px to 479px
*/

@media (min-width: 320px) and (max-width: 480px) {

    /* CSS */

}
```

Com estas resoluções posso tornar meu projeto mais responsivo para diversas telas e plataforma melhorando a experiência do usuário,

Lembrando que previamente já teremos o nosso código CSS original e a partir dele faremos as adaptações para que isso se chame.

Mobile First

Essa é uma linha de código que prioriza primeiro a sua versão mobile fazendo assim primeiro sua construção e só depois indo para as outras versões.

Para finalizar o módulo nós fizemos o último projeto do curso que foi uma tela de login;

Seu repositório- <https://github.com/DevElyShow/Projeto-Tela-de-Login>

Site hospedado- <https://develyslow.github.io/Projeto-Tela-de-Login/>