HTML5

Definição e estrutura básica

Em 1991 Tim Berners-Lee criou essa linguagem de marcação para melhorar a comunicação entre ele e seus colegas de trabalho no CERN, desde então já surgiram 5 versões e o HTML se tornou a base da web.

Com o HTML definimos o significado e a estrutura do conteúdo da web e, além de texto, nossas páginas precisam de imagens, vídeos e vários outros formatos e para isso temos os elementos HTML.

Um elemento HTML é formado pela tag de abertura e seus atributos, o conteúdo e uma tag de fechamento. E mais a frente veremos que existem elementos que não tem tag de fechamento.

Com esses elementos podemos agrupar tipos de conteúdo, alterar tamanho e forma de fontes e adicionar diferentes mídias ao nossa página na web.

E agora podemos ver como é a estrutura básica de um arquivo HTML.

A primeira linha do documento deve ser o <!DOCTYPE html>, apesar de parecer um elemento HTML ela apenas diz ao navegador que ele está lidando com um arquivo do tipo HTML5. Os elementos HTML vem logo abaixo.

<html>

A tag html é a raiz do seu documento, todos os elementos HTML devem estar dentro dela. E nela nós informamos ao navegador qual é o idioma desse nosso documento, através do atributo lang, para o português brasileiro usamos pt-BR.

<head>

A tag head contém elementos que serão lidos pelo navegador, como os metadados - um exemplo é o charset, que é a codificação de caracteres e a mais comum é a UTF-8, o JavaScript com a tag script, o CSS através das tags style e link - veremos a diferença quando falarmos sobre CSS - e o título da página com a tag title.

<body>

E dentro da tag body colocamos todo o conteúdo visível ao usuário: textos, imagens, vídeos.

Prática

Como exercício para esse curso iremos construir um site pessoal, e precisamos começar com a estrutura básica que acabamos de ver.

Vamos criar um arquivo index.html e adicionar o doctype e os elementos html, head e body.

Depois adicionaremos os elementos meta e title, no primeiro adicionamos o atributo charset com o valor UTF-8 para dizer ao navegador qual é a codificação dos caracteres e no segundo podemos colocar nosso nome.

E por último escreveremos nosso nome dentro do elemento body apenas para enxergarmos isso no navegador.

Semântica

Durante muitos anos o elemento padrão no HTML era a div, construíamos nosso conteúdo todo baseado nela, e assim nascia a sopa de divs.

Mas em 2014 saiu a quinta versão do HTML, e com ela vieram vários mudanças importantes, como performance e acessibilidade, mas nesse curso introdutório vamos focar na semântica.

A semântica nos permite descrever mais precisamente o nosso conteúdo, agora um bloco de texto não é apenas uma div, agora é um article e tem mais significado assim. E temos vários elementos para ressignificar as divs:

<section>

Representa uma seção genérica de conteúdo quando não houver um elemento mais específico para isso.

<header>

É o cabeçalho da página ou de uma seção da página e normalmente contém logotipos, menus, campos de busca.

<article>

Representa um conteúdo independente e de maior relevância dentro de uma página, como um post de blog, uma notícia em uma barra lateral ou um bloco de comentários. Um article pode conter outros elementos, como header, cabeçalhos, parágrafos e imagens.

<aside>

É uma seção que engloba conteúdos relacionados ao conteúdo principal, como artigos relacionados, biografia do autor e publicidade. Normalmente são representadas como barras laterais.

<footer>

Esse elemento representa o rodapé do conteúdo ou de parte dele, pois ele é aceito dentro de vários elementos, como article e section e até do body. Exemplos de conteúdo de um <footer> são informações de autor e *links* relacionados.

<h1>-<h6>

Eles não foram criados na versão 5 do HTML e nem são específicos para semântica, mas servem para esse propósito. São utilizados para marcar a importância dos títulos, sendo <h1> o mais importante e <h6> o menos. Uma dica: use apenas um <h1> por página, pois ele representa o objetivo da sua página.

Prática

Dando continuidade ao nosso site iremos montar sua estrutura. Pensei em adicionarmos um cabeçalho com nosso nome, uma lista de posts (como um blog) e um rodapé para nossos contatos.

Vamos abrir nosso arquivo index.html e começar pelo cabeçalho: criamos um <header> logo abaixo do <body> e colocamos o título da nossa página dentro de um <h1>.

Depois criaremos a lista de postagens: abrimos um elemento section e dentro dele adicionamos outro <header> contendo um <h2>. Notem que eu posso ter mais de um <header> na página.

Para criar nossa postagem adicionamos um <article> com um <header> e um <h3>.

O último passo desta etapa é criar um rodapé para nossas informações de contato: crie um elemento footer antes de fechar o </body>.

Não se preocupe com o layout e com conteúdo da página, nós vamos tratar isso mais a frente.

Textos e links

A criação do HTML foi motivada pela necessidade de compartilhar textos e documentos, e mesmo depois de quase 30 anos, com toda a evolução da web, isso ainda representa uma boa parte do conteúdo da web.

```
<h1>Título da página</h1>

<h2>Título de seção</h2>

<h3>Título de artigo</h3>

<p>Conteúdo do artigo.</p>
```

Já falamos anteriormente sobre os elementos h1-h6 e, eles são essenciais para nos indicar visualmente a importância e localização de seções de texto na página, mas para textos maiores e mais densos usamos o elemento p.

O <p> representa um parágrafo, mas ele não suporta apenas texto, podemos adicionar imagens, código, vídeos e vários outros tipos de conteúdo dentro dele.

Um outro elemento interessante e extremamente necessário na web é o <a> - que significa anchor/âncora, ele representa um hyperlink, é ele que interliga vários conteúdos e páginas na web.

```
<a>Link</a>

<a href="linkedin.com/in/vilaboim">LinkedIn</a>
<a href="mailto:lucas@vilaboim.com">E-mail</a>

<a target="_blank">Link</a>
```

O elemento a tem vários atributos, mas vamos focar em dois, o href e o target.

O href representa o *hyperlink* para onde sua âncora aponta, pode ser uma página do seu ou de outro site, um e-mail e até mesmo um telefone, os dois últimos precisam dos prefixos mailto: e tel:, respectivamente.

O target neste momento vai servir para nos ajudar a abrir nossos links em outra aba do navegador usando o valor _blank.

Prática

Vamos adicionar um texto fictício a nossa postagem: logo após o fechamento do </header> vamos adicionar um elemento p e inserir um texto que vamos retirar do site lipsum.com

E em alguma parte deste texto vamos adicionar um *hiperlink* para outra página e um para nosso e-mail.

Criarei um *hyperlink* para meu perfil no *LinkedIn:* adicione o *hyperlink* no atributo href e o valor _blank no atributo target, assim o *link* será aberto em outra aba. E em algum outro lugar do texto adicionarei meu e-mail e um link para ele, desta forma: lucas@vilaboim.com

Imagens

A web também é feita de imagens e para representá-las temos o elemento <img>, ele é um daqueles elementos sem tag de fechamento.

```
<img>

<img src="img/avatar.jpg">

<img alt="Foto de Lucas Vilaboim">
```

O elemento img é bem simples, contendo apenas 2 atributos próprios, o src e o alt.

O src é obrigatório e guarda o caminho para a imagem que você quer mostrar na página.

O alt não é obrigatório mas é altamente recomendado por melhorar a acessibilidade, ele mostra a descrição da imagem caso ela não carregue e leitores de tela usam esse atributo para descrever a imagem para o usuário saber o que ela significa.

Prática

Vamos adicionar uma imagem ao cabeçalho da página e uma imagem a postagem.

Primeiro vamos colocar as imagens na pasta do nosso projeto. Para a imagem do cabeçalho eu escolhi uma foto minha com 100 *pixels* de largura e 100 *pixels* de altura e para a imagem da postagem eu procurei por *html code* no site Unsplash, escolhi uma das imagens e deixei ela com 960 *pixels* de largura por 322 *pixels* de altura.

Dentro do primeiro <header> da página e antes do <h1> iremos adicionar um elemento img e no atributo src colocamos o caminho para a nossa foto, /lucas-vilaboim.jpg, e o atributo alt deve conter um significado para a imagem, como no meu caso é uma ilustração, colocarei *Ilustração do rosto de Lucas Vilaboim*.

E dentro do <header> do <article> vamos fazer a mesma coisa, mas agora depois do <h3>, e no atributo alt colocaremos *Editor de texto mostrando códigos HTML*.

Listas

Os últimos elementos que veremos neste módulo são os relacionados a listas: <ul>, <ol> e <li>.

Listas servem para agrupar uma coleção de itens, como uma lista de ingredientes ou, como será no nosso caso, uma lista com contatos.

O elemento ul cria uma lista não ordenada, onde a ordem dos elementos não é importante, e é representada com pontos, círculos ou quadrados.

O <ol> serve para criar lista ordenadas, nessas a ordem importa, portanto elas são representadas com números, algarismos romanos ou letras.

E o elemento li é um item dentro de uma dessas listas. Um <li> pode conter vários tipos de conteúdos, como parágrafos, imagens e até outras listas.

Prática

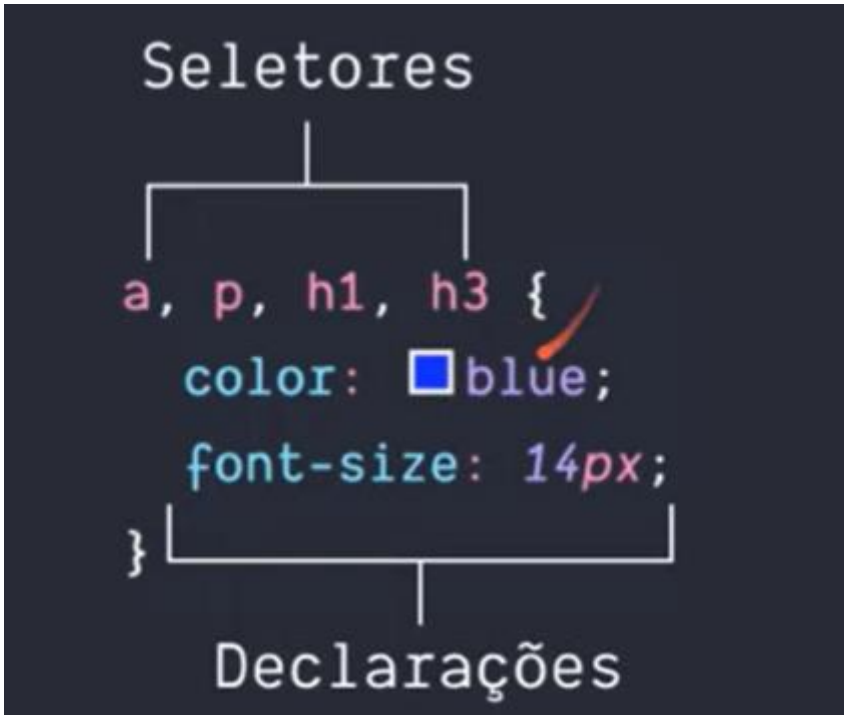
Adicionaremos uma lista de contatos ao rodapé da nossa página, e para isso usaremos também o elemento *a* que vimos anteriormente.

Crie um elemento ul e dentro dele adicione um <li> com um elemento a, no atributo href adicione o *link* de alguma rede social que você mantenha e, no conteúdo da âncora coloque o nome dessa rede.

CSS 3

Definição e seletores

Após a criação do HTML a necessidade de formatar as páginas ficou evidente, assim, em 1996, foi criada a linguagem de estilo que conhecemos por CSS.



A sintaxe é bem simples e pode ser explicada com a frase "você cria regras de estilo para elementos ou grupos de elementos".

Vamos usar um elemento HTML que vimos anteriormente, a âncora <a>, para exemplificar.

Uma regra CSS é representada por um seletor ou um grupo de seletores, no nosso caso é o <a>, então dentro de um par de chaves adicionamos as declarações, no exemplo acima estamos alterando cor e tamanho da fonte dessa âncora, as declarações são formadas por uma propriedade e um valor.

Percebam que podemos colocar vários seletores em uma regra separando-os por vírgula.

E há um último detalhe nesse exemplo: a pseudo-classe. Elementos HTML sofrem alterações causadas pela interação do usuário, como mover o mouse por cima ou clicar nesse elemento.

O *a:hover* do exemplo significa que a âncora também terá essa aparência quando o usuário passar o mouse por cima de um *hyperlink*.

ID x Classe

No exemplo anterior criamos uma regra que altera um elemento HTML diretamente, mas isso significa que todos os elementos <a> ficarão com aquela aparência, e normalmente temos sites mais complexos que precisam de várias regras diferentes para elementos iguais.

Para ficar mais tangível vamos relembrar um pouco o site que começamos a fazer no módulo passado, ele tinha vários elementos header, mas não vamos querer que o header principal tenha a mesma formatação que o header de uma postagem, é aí que entram os IDs e Classes.

O seletor que vimos no primeiro exemplo é um seletor de tipo, pois ele representa um elemento HTML, e com IDs e Classes podemos representar qualquer tipo de elemento mas há algumas diferenças entre eles:

CSS

```
<header id="header" class="header"></header>

<header class="header"></header>
```

HTML

```
.header {
  padding: 10px;
}

#header {
  padding: 15px;
}
```

ID: é representado pelo símbolo # (hash) seguido de um nome para esse ID.

Classe: a classe é representada de forma parecida do ID, mas é precedida por um ponto em vez do hash.

E a diferença mais importante entre eles é a forma como devem ser usados: o ID só pode ser usado uma vez em uma página HTML enquanto a classe não tem restrições.

Exercício

Vamos adicionar algumas classes no nosso site e alterar alguns elementos, mas antes precisamos adicionar um arquivo CSS a nossa página.

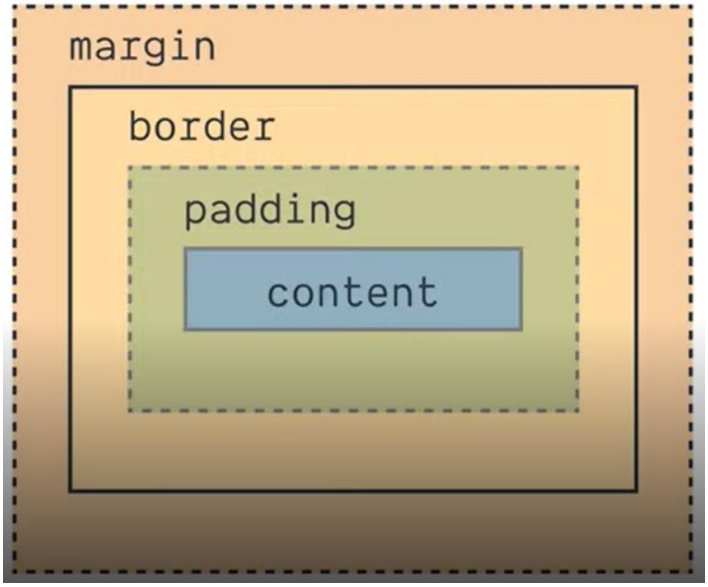
No módulo de HTML descobrimos que podemos adicionar CSS de duas formas, com o elemento *style*, e assim suas regras ficarão no arquivo HTML, ou podemos criar um arquivo CSS e adicioná-lo na página através do elemento *link*, e é essa forma que usaremos.

Crie um elemento *link* dentro do head do seu arquivo e adicione os atributos *rel="stylesheet"* e *href="style.css"*, o *rel* denota o tipo de arquivo que estamos incluindo na página e o *href* é o caminho para o arquivo. E na mesma pasta do arquivo *HTML* crie um arquivo chamado *style.css*.

Agora sim vamos ao CSS, adicione um ID #title ao h1 da página, pois queremos que ele seja único, e depois adicione as classes .subtitle e .post_title ao h2 e h3, respectivamente.

No arquivo CSS vamos mudar a cor desses três títulos, e depois alterar o tamanho da fonte do título da postagem.

Box-model



Quando estamos criando o layout de um site o navegador representa cada elemento HTML como uma caixa retangular, isso é o box-model. E com CSS nós alteramos a aparência dessa caixa (largura, altura, cor de fundo, etc.). Essa caixa é composta por 4 áreas: o conteúdo, o padding, a borda e a margem.

- As margens (margin) são espaçamentos entre elementos;
- As bordas (border) ;
- O padding é um espaçamento entre as bordas e o conteúdo, a diferença para as margens é que declarações de imagem de fundo funcionam nele;
- O conteúdo (content) é o que o seu bloco representa, um texto, uma imagem, um vídeo;

Exercício

Para enxergamos o box-model vamos adicionar cores e bordas a alguns elementos.

Primeiro adicionaremos uma cor de fundo para a visualização ficar mais fácil, usaremos a propriedade *background* com o valor *#fcfcfc* no elemento *body*.

Depois vamos adicionar uma classe ao <article>, pode ser .post, e então vamos colocar a cor branca de fundo com a propriedade *background* e o valor *#FFF*. Agora conseguimos enxergar o *content* do *box-model*.

Vamos adicionar um *padding* de 10 pixels neste mesmo *article*. Perceberam o espaçamento que surgiu em volta do nosso conteúdo?

Agora adicionamos um borda mais escura a ele com a propriedade border. Vou falar mais detalhadamente sobre border mais a frente, mas por enquanto vamos deixar essa borda com 3 pixels de largura, o contorno sólido e a cor azul.

E por último vamos adicionar uma margem do lado de fora do post com a propriedade *margin* e o valor 10 pixels.

E agora inspecionando o nosso elemento conseguimos todas aquelas camadas citadas antes: o conteúdo em azul, o *padding* em verde, as bordas em marrom e as margens em laranja.

E já que começamos a falar sobre bordas e cor de fundo, no próximo vídeo vamos nos aprofundar nessas propriedades.

Estilizando elementos

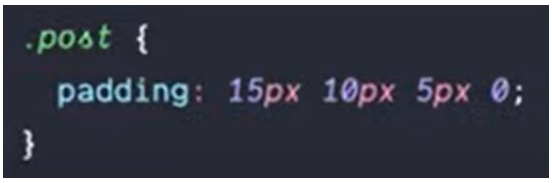
Agora que entendemos o box-model podemos focar em deixar nosso site mais bonito, então vamos repassar pelas propriedades já citadas:

Padding e Margin

Anteriormente usamos o *padding* e o *margin* da forma mais básica, com apenas um valor, mas eles são mais poderosos que isso. Se quisermos atribuir tamanhos diferentes para cada lado do *box* nós podemos, e vamos ver três formas de fazer isso.

A primeira é colocando um valor para as partes superior e inferior e depois para os lados esquerdo e direito.

O valor de 10 *pixels* se refere ao eixo Y, ou partes superior e inferior, e os 5 *pixels* se referem aos lados esquerdo e direito.



A segunda forma é dando valores para cada lado do *box*.

Então começamos pelo topo com 15 pixels, passamos o lado direito com 10 pixels, depois para a parte inferior com 5 pixels e por último o lado esquerdo com 0, e sempre nessa ordem.

Uma boa dica também é que quando o valor for 0 não precisamos não precisamos colocar a unidade.

A terceira forma é com as propriedades específicas para cada lado, até agora tínhamos visto atalhos para essas propriedades.

Essa opção é mais usada quando temos o mesmo valor para 3 lados, e o quarto precisa ter um valor diferente, então usamos o padding com apenas um valor e uma dessas opções para representar o lado diferente.

Background

A propriedade *background* também é um atalho para várias propriedades, mas isso vocês podem absorver aos poucos, e uma boa opção de leitura é a documentação do MDN.

Por enquanto veremos apenas como mudar a cor de fundo.

```css
.post {
  background-color: green;
  background-image: url("bg.png");
  background-position: top;
}
```

```css
.post {
  background-color: green;
  background-color: #008800;
  background: #008000;
}
```

E aqui temos 3 formas de colocar uma cor de fundo, e ainda existem outras.

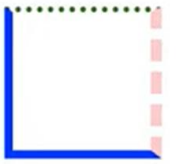
A primeira é pelo nome da cor em inglês, a segunda é pelo código hexadecimal e a terceira é usando apenas o atalho *background*.

Border

Vimos que a propriedade *border* pode ter 3 valores: a largura, a cor e o estilo, mas existem algumas particularidades nisso.

A largura pode ser usada com várias unidades, como px, em e mm. A cor pode ser atribuída pelo nome ou por um código hexadecimal, assim como fizemos com o *background*, e o estilo é representada por palavras-chave, vamos ver algumas delas:

```css
.post {
  border: 3px solid blue;
  border-top: 2px dotted green;
  border-right: 4px dashed pink;
}
```



solid: mostra uma borda simples e reta;

dotted: são bolinhas com um pequeno espaçamento entre elas;

dashed: forma uma linha tracejada.

E aproveitando que mostrei esse código temos que falar sobre como separar a estilização dos lados de uma borda.

E se você não quiser usar a propriedade *border* existem as propriedade específicas para cada aspecto de uma borda, são elas *border-width* para a largura, *border-color* para a cor e *border-style* para o estilo.

Aqui temos o mesmo código anterior de duas formas diferentes, a primeira com o atalho *border* e a segunda com cada propriedade específica.

E depois disso podemos juntar os lados com os aspectos de uma borda e criar uma regra mais específica ainda.

Border-radius

E a última propriedade é o *border-radius*, ele permite arredondar os cantos de um elemento. Podemos usar várias unidades, mas as mais comuns são os pixels e a porcentagem.

Colocando apenas um valor mudamos todos os cantos do elemento, mas seguindo aquela mesma ordem que vimos no *padding* e *margin* - topo, direita, inferior e esquerda - conseguimos alterar cada canto separadamente.

Exercício

Neste exercício vamos deixar o nosso site um pouco mais bonito usando as propriedades que acabamos de ver.

Vamos aumentar o padding para 15 pixels e colocar uma margem de também de 15 pixels só na parte de baixo do post.

Quando olhamos para os textos percebemos que os espaçamentos estão diferentes do restante do post, então vamos padronizar isso.

No título do post vamos retirar todas as margens para depois colocar apenas uma margem inferior de 15 pixels. E no corpo do post precisamos adicionar uma classe e remover todas as margens para depois adicionar uma margem superior de 15 pixels.

Podemos manter o background branco, mas vamos diminuir a largura das bordas para 2 pixels e mudar a cor para a mesma do texto - #505050 - e por último adicionaremos um border-radius, 5 pixels são suficientes. Podemos adicionar esse mesmo de valor de border-radius na imagem, para isso vamos acrescentar uma class a imagem antes.

Estilizando textos

Já sabemos que podemos mudar cor e tamanho de algumas fontes, e agora vamos nos aprofundar nisso.

font-family

```
#title {
  font-family: Verdana;
}

.post_title {
  font-family: Verdana, Arial;
}
```

Com o font-family podemos alterar a fonte dos nossos textos, como uma fonte da internet ou uma que esteja instalada no nosso computador, mas vamos nos ater às fontes seguras, chamadas de web safe fonts.

Essas fontes são chamadas assim pois são encontradas em quases todos os sistemas e podem ser usadas sem preocupação.

font-size

```
#title {
  font-size: 30px;
}

.post_title {
  font-size: 18px;
}
```

O font-size nos ajuda a mudar o tamanho do texto, existem algumas unidades de medida para ele mas por enquanto os pixels são suficientes para nós.

font-style

```
#title {
  font-style: normal;
}

.subtitle {
  font-style: italic;
}
```

Usamos o font-style para tornar um texto itálico, na maioria das vezes você usará apenas o valor *italic* para ele, mas se precisar tirar o itálico de um texto você pode usar o valor *normal*.

Font-Weight

```
#title {
  font-weight: normal;
}

.subtitle {
  font-weight: bold;
}
```

Altera o peso do texto, bold é negrito.

Text-transform

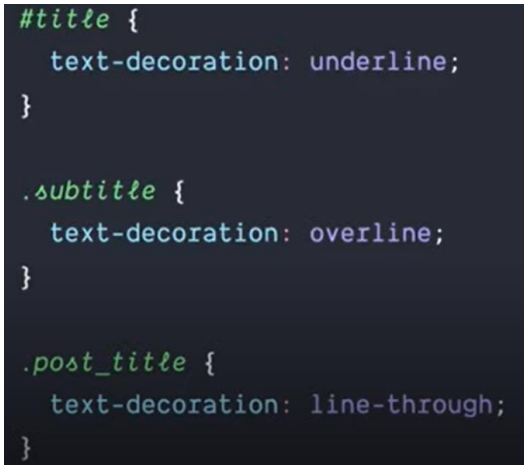
```
#title {
  text-transform: uppercase;
}

.subtitle {
  text-transform: lowercase;
}

.post_title {
  text-transform: capitalize;
}
```

Uppercase: COLOCA TODO O TEXTO EM CAIXA ALTA

Lowercase: coloca todo o texto em caixa baixa

Capitalize: Coloca Todas As Primeiras Letras De Cada Palavra Em Maiúsculo

Text-decoration



```
#title {
  text-decoration: underline;
}

.subtitle {
  text-decoration: overline;
}

.post_title {
  text-decoration: line-through;
}
```

Serve para destacar textos - Coloca linhas no texto

Underline coloca linha abaixo

Overline coloca linha acima

E line-through coloca uma linha ao centro da palavra, cortando essa

Estilizando listas

```css
ul {
  list-style-type: square;
}

ol {
  list-style-type: upper-roman;
}

ul {
  list-style-type: "\1F44D";
}
```

```css
ul {
  list-style-image: url("rocket.png");
}
```

Dimensões e alinhamento

Width
Height

Max-width
Max-height

Margin

Text align