



*DÊ UM START EM
SUA CARREIRA DEV.*

A JORNADA COMPLETA PARA VOCÊ ENTRAR NA
CARREIRA DO FUTURO, **SER DISPUTADO PELAS
EMPRESAS E GANHAR 5X MAIS.**

ORIENTAÇÕES GERAIS

O CURSO

- ➡ O curso está estruturado em módulos. Cada um deles irá te levar para um próximo nível de conhecimento progressivamente.
- ➡ **Módulo A - HTML Básico**
O HTML é utilizado para estruturar a página. Nesse módulo, vamos aprender um pouco mais sobre essa linguagem e suas principais tags.
- ➡ **Módulo B - CSS Básico**
Nesse módulo vamos ver uma introdução ao CSS e sua utilização mais básica, desde a sua definição, passando por cores, fontes até dimensionamento e posicionamento simples de elementos.

ORIENTAÇÕES GERAIS

O CURSO

- ➡ **Módulo C - CSS Avançado**
Nesse módulo vamos nos aprofundar um pouco mais em CSS. Vamos avançar no posicionamento e ver conceitos responsividade e utilização de variáveis no CSS.
- ➡ **Módulo D - Publicando seu website**
Esse é o módulo final. Aqui temos uma revisão de todo o conteúdo e um tutorial básico para você publicar seu website.
- ➡ Alguns vídeos estão em inglês. Porém não se preocupe, caso não domine a língua. Você pode ativar as legendas clicando no ícone CC na barra de ferramentas do vídeo. Caso tenha dúvidas, segue link explicativo sobre a ativação das legendas: [Como ativar as legendas](#)

ORIENTAÇÕES GERAIS

AS FERRAMENTAS

➡ Para que você aproveite e aprenda muito durante o curso, selecionamos algumas ferramentas(sem custo para você) que servirão para a visualização das aulas, execução dos exercícios e entrega das atividades.

➡ Essas são as ferramentas que irá utilizar:

[GitHub: Ferramenta de versionamento](#)

[VS CODE: editor que vamos utilizar para editar os códigos.](#)

[Page Ruler: ferramenta para tirar medidas do site](#)

[Code Pen: editor online](#)

MÓDULO A - HTML BÁSICO

AULA 1 - HTML

[ASSISTA A AULA](#)

INTRODUÇÃO


Vamos mergulhar no HTML (Linguagem de Marcação de Hipertexto). O HTML é uma linguagem de marcação que fornece um modo estruturado de criar páginas web. Ao usarmos as tags que o HTML nos disponibiliza, podemos controlar como quase tudo é apresentado no navegador do usuário. É o navegador quem interpreta as tags para apresentar o que o usuário final vê.

[Referência de Elementos/Tags HTML](#)

[Referência do HTML 5](#)

HTML E SUAS TAGS

O HTML é uma linguagem de marcação interpretada pelo navegador. Ela usa tags ou elementos específicos para dizer ao navegador como as coisas devem ser apresentadas visualmente ao usuário. Vamos dar uma olhada em alguns conceitos por trás de como essas tags funcionam antes de vermos as tags em si.

 Nem todas as tags desses exemplos são tags HTML reais. Elas estão sendo usadas para explicar os conceitos.

As tags são colocadas dentro de sinais de menor que (<) e maior que (>). Perceba que a tag de fechamento leva o caractere /. Isto diz ao navegador que esta é a tag de fechamento correspondente. As tags são quase sempre seguidas de tags de fechamento correspondentes.

```
<tag>conteúdo</tag>
```

As tags também podem ser aninhadas umas dentro das outras.


```
<div>  
  <p>abacaxi</p>  
</div>
```

```
<div>  
  <p>queijo</p>  
  <p>pizza</p>  
  <p>presunto</p>  
</div>
```

As tags usam o que chamamos de atributos para adicionar funcionalidades ou descrever comportamentos.

Algumas tags têm fechamento próprio. Essas tags normalmente não envolvem algum conteúdo. A tag `` é uma tag de fechamento próprio bem comum. Como você pode ver, ainda há uma / no final da tag. Observe também o uso dos atributos.

1. ``

ESTRUTURA BÁSICA DE PÁGINAS HTML

ASSISTA A AULA

Nesta seção vamos dar uma olhada na estrutura de um arquivo html. Vamos discutir as tags básicas necessárias para ter uma página bem formada para exibição na internet. A partir de agora, vamos falar no contexto do HTML5.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title></title>`
5. `</head>`
6. `<body>`

7. `</body>`
8. `</html>`

Vamos dar uma olhada em algumas dessas tags/elementos.

- Doctype

`<!DOCTYPE html>` informa ao navegador qual versão do HTML será usada neste documento. Neste caso, estamos declarando o HTML5. Doctype é especial no sentido de que ele não é considerado uma tag de verdade, mas sim uma declaração.

O que você realmente precisa entender sobre esta tag/declaração é que ela deve estar presente, e deve estar especificamente na primeira linha do seu arquivo HTML.

- html

<html> é a tag que representa o elemento raiz (elemento de nível superior) de um documento HTML, por isso ela também é chamada de elemento raiz. Todos os outros elementos devem ser descendentes deste elemento. Dito isto, observe a tag de fechamento **</html>** na última linha.

- head

<head> é a tag que contém informações descritivas sobre a sua página, como o título

<title> das páginas. Também é aqui onde você insere coisas como **<meta>** e planilhas de estilo. Falaremos mais sobre essas coisas mais para frente nesse curso. Observe que a tag **<title>** possui uma tag de fechamento, **</title>**, assim como a tag **<html>**. Acredito que você esteja percebendo o padrão.

[Leitura Recomendada](#)

- title

<title> é a tag que define o título do documento, ele é exibido na barra de título do navegador ou na aba da página. Ela pode conter apenas texto, e todas as tags colocadas aqui são ignoradas. Ela é seguida da tag de fechamento **</title>**.

- body

`<body>` representa o conteúdo de um documento HTML. Pode haver apenas um elemento `<body>` em um documento HTML. É aqui onde ficam as coisas que queremos que as pessoas vejam. Vou parar de apontar as tags de fechamento. Acho que você já pegou a ideia.



Leitura adicional

[Introdução ao HTML no MDN](#)

[O que há no head? no MDN](#)

[O elemento meta no MDN](#)

[Lista de códigos ISO 639-1 no Wikipedia](#)

AULA 2 - HTML ELEMENTOS DE BLOCO E EM LINHA

[ASSISTA A AULA](#)



Leitura adicional

[Elementos inline no MDN](#)

[Elementos em nível de bloco no MDN](#)

TRANSCRIÇÃO DO VÍDEO

Olá, a todos. Sejam bem-vindos à Kenzie Academy. Meu nome é Davey Strus.

Aninhamos tranquilamente elementos uns dentro dos outros, e até agora tudo certo.

Desde que você feche suas tags adequadamente, você pode colocar os elementos uns dentro dos outros.


```
<p>
  Estou <em>muito</em> entediado!
</p>
```

Ao fazer isso, o elemento externo é chamado de pai do elemento interno, e o elemento interno é o filho do elemento externo.

Ah, mas e se o filho tiver seus próprios filhos?

```
<body>
  <p>
    Vamos ler <cite>Carlos Drummond de
Andrade!</cite>
  </p>
</body>
```

O parágrafo é filho do body, e a citação é filha do parágrafo. Portanto qual a relação entre o body e cite? Ué, a cite é o neto do body!

Vamos mais fundo.

```
<html>
  <body>
    <p>
      Vamos ler <cite>Carlos Drummond de
Andrade!</cite>
    </p>
  </body>
</html>
```


O parágrafo é filho do body, e a citação é filha do parágrafo. Portanto qual a relação entre o body e cite? Ué, a cite é o neto do body!

Vamos mais fundo.

```
<html>
  <body>
    <p>
      Vamos ler <cite>Carlos Drummond de
Andrade!</cite>
    </p>
  </body>
</html>
```

Termos como bisavô não são muito úteis em termos de marcação. Neste ponto, os chamamos apenas de ancestrais e

descendentes. O elemento html é um ancestral de todos os elementos da página, e todos os outros elementos são descendentes do elemento html. Falaremos mais sobre isso e sobre a estrutura de árvore que isso cria mais para frente. Isso é tudo que você precisa saber por enquanto.

Então agora sabemos tudo sobre aninhar elementos uns dentro dos outros, e você provavelmente pode aninhar qualquer elemento dentro de outro, certo? Não! Há regras.

Para determinar quais elementos podem ser aninhados dentro de outro, primeiro temos que entender as diferenças entre elementos de nível de bloco e elementos em linha. O HTML 5 reclassificou os elementos

HTML usando um sistema muito mais complicado que, sinceramente, não é muito útil no começo, e entender as antigas classificações de nível de bloco e em linha são úteis, então vamos por este caminho para ajudar no aprendizado.

A versão super curta é que os elementos de nível de bloco começam uma nova linha e preenchem toda a largura de seu container. Elementos em linha podem iniciar e terminar no meio de uma linha.

Exemplos de elementos de nível de bloco que você já conhece incluem h1 até h6 e os parágrafos. Os elementos em linha que você já está familiarizado incluem em, strong, cite, i, b, e u.

Mas quais são as restrições de quais elementos podem ser aninhados dentro de outros elementos? Bom, como você deve ter imaginado, você não pode colocar um elemento de nível de bloco dentro de um elemento em linha. Você pode colocar um elemento em linha dentro de outro elemento em linha. E você pode colocar um elemento de nível de bloco dentro de outro elemento de nível de bloco... às vezes. Há algumas exceções bem grandes. Na verdade, você não pode colocar elementos de nível de bloco dentro de um parágrafo ou dentro de títulos, h1 até h6. Isso significa que ainda não falamos sobre nenhum elemento que possa conter outros elementos de nível de bloco.

Conheça o `<div>`. É uma abreviação de divisão de conteúdo, e é um elemento sem significado semântico. Ele não tem função a não ser servir de container para outros elementos. É um elemento de nível de bloco e pode conter qualquer outro elemento que pertence ao corpo. Não é muito útil até que você comece a mexer com CSS, mas é bom saber que ele existe para aquelas ocasiões onde você precisa de um tipo de container e não há um elemento mais apropriado. Ele é essencialmente o elemento de nível de bloco genérico.

E como você deve estar imaginando, há um equivalente em linha: ``. Ele não significa nada. É apenas um container em linha genérico. Mas às vezes é disso que você precisa, porque é necessário aplicar

algum CSS em linha, e não há um elemento semântico melhor que este.

E é isso. Até a próxima, meu nome é Davey Strus. Booom desenvolvimento!

AULA 3 - HTML LISTAS

[ASSISTA A AULA](#)



Leitura adicional

[Listas em HTML no w3schools](#)

[Listas não ordenadas no MDN](#)

[Listas ordenadas no MDN](#)

[Lista de itens no MDN](#)

[Lista de itens no MDN](#)

TRANSCRIÇÃO DO VÍDEO

Olá, pessoal. Sejam bem-vindos à Kenzie Academy. Meu nome é Davey Strus.

Estamos progredindo bem com o HTML, mas nossas páginas ainda estão um pouco tediosas, apenas com títulos e parágrafos, e com algumas frases em negrito ou itálico. Vamos agitar um pouco as coisas com listas. Um pequeno agito, mas não deixa de ser um agito.

Estamos interessados em dois tipos de listas: Listas não ordenadas e listas ordenadas. Ambas são constituídas de itens de lista. Há um terceiro tipo de lista em HTML chamado de lista de definições, mas ela não é constituída de itens de lista, e geralmente não é tão parecida (nem tão comum) com as outras duas, então iremos focar nas listas não ordenadas e ordenadas por enquanto. Sinta-se à vontade para seguir o link e ler mais sobre listas de definição quando quiser.

Tanto as listas não ordenadas quanto as listas ordenadas são constituídas de itens de lista, e a marcação é virtualmente idêntica. Então qual a diferença? Bem, está no nome. Em uma lista não ordenada, a ordem dos itens não importa. Você pode

reorganizar a lista e ela ainda fará sentido. O navegador não irá reorganizá-la para você de maneira inesperada, mas ao decidir qual tipo de lista usar, a questão é: A ordem dos itens é importante? Se sim, então você precisa de uma lista ordenada. Vamos dar uma olhada na marcação.

```
<h1>Filmes de 1978</h1>
```

```
<h2>Filmes de maior bilheteria (EUA)</h2>
```

```
<ol>
```

```
  <li>Grease — Nos Tempos da Brilhantina</li>
```

```
  <li>Superman</li>
```

```
  <li>Clube dos Cafajestes</li>
```

```
  <li>Doido para Brigar... Louco para Amar</li>
```

```
  <li>O Diabo Disse Não</li>
```


<h2>Indicados para melhor filme</h2>

Expresso da Meia-noite

O paraíso Pode Esperar

O Franco Atirador

Amargo Regresso

Uma Mulher Descasada

À esquerda, há uma lista ordenada, marcada com o elemento À direita, há uma lista não ordenada, . Ambas são constituídas de itens de lista, marcadas com . a lista da esquerda, a ordem é importante. São os dez filmes de maior bilheteria do ano, em ordem. Na lista da direita, a ordem não interessa. São as indicações de melhor filme sem ordem

específica.

Então como elas serão apresentadas no navegador?

Bem, devo admitir que eu trapaceei um pouquinho aqui. Eu usei alguns divs e um pouco de CSS para fazer as listas aparecerem lado a lado. As suas aparecerão uma embaixo da outra se você copiar o HTML que eu acabei de mostrar. Isso facilita na hora de encontrar as diferenças. Em suma, a lista ordenada está numerada, e a lista não ordenada está em tópicos. Faz sentido, certo? Já que a ordem faz diferença em uma lista ordenada, a numeração faz sentido. E na lista não ordenada, os pontos de tópico deixam claro que isso é, de fato, uma lista, mesmo que não tenha uma ordem específica. E esta é a diferença básica.

As listas ordenadas também possuem alguns atributos opcionais que as listas não ordenadas não têm. Se você quiser começar uma lista ordenada, digamos, no 6º item, simplesmente adicione `start="6"`.

```
<h2>Filmes de maior bilheteria (EUA)</h2>
<ol start="6">
  <li>Hooper, o Homem das Mil Façanhas</li>
  <li>Tubarão 2</li>
  <li>A Vingança da Pantera Cor-de-Rosa</li>
  <li>O Franco Atirador</li>
  <li>Halloween - O Início</li>
</ol>
```

Quer listar as coisas em ordem invertida? Use o atributo booleano `reversed`.

```
<h2>Top 5</h2>
<ol reversed>
  <li>O Paraíso Pode Esperar</li>
  <li>Doido para Brigar... Louco para Amar</li>
  <li>Clube dos Cafajestes</li>
  <li>Superman</li>
  <li>Grease — Nos Tempos da Brilhantina</li>
</ol>
```

Lembra dos atributos booleanos? Os valores deles são opcionais. Se você incluir o valor, ele deve ser o mesmo do nome do atributo. Neste caso, `reversed="reversed"`.

```
<h2>Top 5</h2>
<ol reversed="reversed">
  <li>O Paraíso Pode Esperar</li>
```



```
<li>Grease — Nos Tempos da  
Brilhantina</li>  
</ol>
```

Você também pode mudar a numeração para usar letras maiúsculas ou minúsculas, ou algarismos romanos, usando o atributo `type`.

```
<h1>Review de Roger Ebert's sobre Grease  
— Nos Tempos da Brilhantina</h1>  
<ol type="I">  
  <li>Introdução</li>  
  <li>Elenco</li>  
  <li>Resumo do enredo</li>  
  <li>Pontos Fortes</li>  
  <li>Pontos Fracos</li>  
  <li>Conclusão</li>  
</ol>
```

Coloque `type="I"` para algarismos romanos maiúsculos, `"i"` para algarismos romanos minúsculos, `"A"` para letras maiúsculas, `"a"` para letras minúsculas, ou `"1"` para números, que é o padrão.

Portanto as listas ordenadas possuem alguns atributos que podemos usar: `start`, `reversed`, e `type`. As listas não ordenadas não possuem atributos que são parte da especificação atual. Se você quiser exibi-las de maneira diferente, você precisará usar CSS.

E quanto aos itens de lista? Eles possuem um atributo muito útil: `value`, que funciona apenas em listas ordenadas. Ele especifica o valor de um item de lista específico, e depois dele a lista continuará sendo

numerada automaticamente. Isso é útil se houver um empate, por exemplo.

```
<h1>Filmes com a maioria dos prêmios da Academia</h1>
```

```
<ol>
```

```
  <li>Ben-Hur (empate)</li>
```

```
  <li value="1">Titanic (empate)</li>
```

```
  <li value="1">O Retorno do Rei
```

```
(empate)</li>
```

```
  <li value="4">Amor, Sublime Amor</li>
```

```
</ol>
```

E adivinha só. Você pode aninhar listas dentro de listas! Eita! Simplesmente inclua outra lista dentro de um item de lista antes de fechar a tag.

```
<h1>Review de Roger Ebert's sobre Grease  
— Nos Tempos da Brilhantina</h1>
```

```
<ol type="I">
```

```
  <li>Introdução</li>
```

```
  <li>Elenco
```

```
    <ul>
```

```
      <li>Olivia Newton-John</li>
```

```
      <li>John Travolta</li>
```

```
    </ul>
```

```
  </li>
```

```
  <li>Resumo do enredo</li>
```

```
  <li>Pontos Fortes</li>
```

```
  <li>Pontos Fracos</li>
```

```
  <li>Conclusão</li>
```

```
</ol>
```


Aqui eu aninhei uma lista não ordenada dentro de uma lista ordenada. É possível fazer isso e vice-versa. Naturalmente, você também pode aninhar uma lista dentro de outra lista do mesmo tipo.

Na verdade, você pode colocar quase qualquer outro tipo de elemento dentro de um item de lista: parágrafos, títulos, divs... como quiser! E claro, você pode colocar elementos em linha também.

```
<h1>Review de Roger Ebert's sobre Grease  
— Nos Tempos da Brilhantina</h1>
```

```
<ol type="I">  
  <li>Introdução
```

```
    <p><cite>Grease — Nos Tempos  
da Brilhantina</cite> vale a pena ver  
pela nostalgia, mas é apenas um musical  
mediano.</p>
```

```
</li>  
<li>Elenco</li>  
<li>Resumo do enredo</li>  
<li>Pontos Fortes</li>  
<li>Pontos Fracos</li>  
<li>Conclusão</li>  
</ol>
```

E aí, você está pronto para temperar um pouco suas páginas com listas? Sim, é claro que sim, seu fazedor de listas! Até a próxima, meu nome é Davey Strus. Boooooas codificações!

Aqui eu aninhei uma lista não ordenada dentro de uma lista ordenada. É possível fazer isso e vice-versa. Naturalmente, você também pode aninhar uma lista dentro de outra lista do mesmo tipo.

Na verdade, você pode colocar quase qualquer outro tipo de elemento dentro de um item de lista: parágrafos, títulos, divs... como quiser! E claro, você pode colocar elementos em linha também.

```
<h1>Review de Roger Ebert's sobre Grease  
— Nos Tempos da Brilhantina</h1>
```

```
<ol type="I">  
  <li>Introdução
```

```
    <p><cite>Grease — Nos Tempos  
da Brilhantina</cite> vale a pena ver  
pela nostalgia, mas é apenas um musical  
mediano.</p>
```


EXERCÍCIOS: LISTAS ORDENADAS E NÃO ORDENADAS

VISÃO GERAL

Construa duas listas, uma ordenada e outra não ordenada

REQUISITOS

Fazer uma lista ordenada com seus FILMES favoritos. Começando com os mais favoritos seguindo para os menos favoritos.

Fazer uma lista não ordenada com seus HOBBIES.

Acesse o [exercício](#)

AULA 4 - HTML HIPERLINKS

[ASSISTA A AULA](#)



Leitura adicional

[HTML tag <a> on w3schools](#)

[O elemento âncora on MDN](#)

[sobre rel=noopener by Mathias Bynens](#)

TRANSCRIÇÃO DO VÍDEO

Olá, pessoal. Sejam bem-vindos à Kenzie Academy. Meu nome é Davey Strus.

Aprendemos como fazer todo tipo de coisa chique com textos, mas ainda é só texto. Queremos hipertexto! Para isso, precisamos de hiperlinks.

Um hiperlink, ou apenas link, se você preferir assim, nos envia para outro lugar na internet. Pode ser outra página do mesmo site, outro site, ou para uma parte diferente da página onde estamos. Pode

até mesmo ser um link para baixar um arquivo. Se o objeto tiver uma URL, você pode criar um link até ele.

Você faz isso com o elemento `<a>`.

`<a>`

É um a de âncora, apesar de seu uso como âncora agora ser obsoleto. Hoje em dia, ele é usado apenas para links. O atributo crucial aqui é o href.

`Sobre nós`

Quando um href estiver presente, o elemento se torna um hiperlink, e o conteúdo entre as tags de abertura e de

fechamento podem ser ativados para seguir o link. O href aponta para uma URL, ou para um fragmento de URL.

Um fragmento de URL começa com uma cerquilha (#) e aponta para um elemento específico da página atual, usando o id do elemento. Você pode dar um id a qualquer elemento usando, com antecedência suficiente, o atributo id.

`Produtos`

`<h2 id="products">Nossos produtos</h2>`

Este link aponta para aquele h2, pois o nome que segue a cerquilha do fragmento de URL bate com o id do h2. Ao ativar aquele link, o navegador irá rolar a página

até onde o h2 está. Você também pode usar um fragmento no final de uma URL para linkar a um id específico de outra página.

```
<a href="about.html#founders">  
Fundadores  
</a>
```

Este link nos enviaria para o about.html e em seguida iria rolar a nova página até o elemento com o id "founders".

Você também pode linkar ao topo da página atual usando #top ou apenas #. Você não precisa ter um elemento com o id "top" para que isso funcione.

O conteúdo de um link, ou seja, as coisas

que aparecem entre as tags de abertura e fechamento, e as coisas que se tornam clicáveis, podem ser praticamente qualquer coisa, exceto outro link. Nada de links dentro de links. Você pode pensar em links como elementos em linha. Como mencionei antes, o HTML redefiniu os tipos de conteúdo, pois precisávamos de mais confusão em nossas vidas, mas os links não inserem novas linhas automaticamente nem preenchem a largura de seus containers por padrão. Portanto, eles agem como elementos em linha. Apesar disso, você pode colocar um elemento de nível de bloco dentro de um link, por mais esquisito que isso se pareça para caretões como eu, que se lembram de quando não se podia fazer isso.


```
<h1>  
  <a href="https://kenzie.com.br/">Kenzie  
Academy Brasil</a>  
</h1>
```

```
<a href="https://kenzie.com.br/">  
  <h1>Kenzie Academy Brasil</h1>  
</a>
```

Aqui, fiz das duas formas. Aninhei um link dentro de um h1 e um h1 dentro de um link. Qual a diferença prática?

No primeiro link, apenas o texto é clicável. Observe o ponteiro do mouse. Eu preciso colocá-lo diretamente sobre a palavra. No segundo link, todo o h1, e não apenas o texto dentro do h1, é clicável. Já que o h1 é um elemento de nível de bloco, ele

ocupa toda a largura. Então todo o espaço à direita da palavra ainda faz parte do h1 e é clicável.

Os links têm vários atributos além do href, mas o que você provavelmente irá usar mais é o target. A maioria dos valores possíveis para o target tem a ver com frames, os quais não mexeremos agora. Mas o uso mais comum do target é o de dar o valor _blank para abrir um link em uma nova aba ou janela.

```
<a href="https://kenzie.com.br/" target="_  
blank">  
  Kenzie Academy Brasil  
</a>
```


Isso é comum ao linkar a recursos externos, tipo ao linkar a outro site, sob certas circunstâncias em que você acredita que o usuário provavelmente irá desejar manter seu site aberto. Por exemplo, usamos isso nas páginas deste curso ao linkar a um recurso externo, pois imaginamos que você provavelmente quer manter a aula aberta enquanto lê outra coisa. Mas cuidado para não exagerar com o `target="_blank"`. Ele está fazendo uma suposição sobre a intenção do usuário, e ao passo que ele sempre pode clicar com o botão direito ou com a tecla CTRL pressionada para abrir qualquer link em uma nova aba, se você marcá-lo com `target="_blank"`, o usuário não terá escolha a não ser abri-lo em uma nova aba. Eles não têm a opção de ignorar este

comportamento.

Ao usar `target="_blank"`, é uma boa ideia adicionar `rel="noopener"` ao link.

```
<a  
  href="https://kenzie.com.br/"  
  target="_blank"  
  rel="noopener"  
>  
  Kenzie Academy Brasil  
</a>
```

O atributo `rel` especifica a relação entre a página atual e a página para qual o link está apontando. Você pode não usá-lo muito, e deve sentir-se à vontade para ler mais sobre isso por conta própria; mas neste caso, você pode usar isso para lidar com um possível problema de segurança ao

usar `target="_blank"`. Você pode ler mais sobre a natureza exata deste problema de segurança seguindo um link nesta aula. Não vou falar sobre isso neste vídeo.

Você deve ter notado que eu coloquei algumas linhas em branco no meio da tag de abertura aqui, pois o número de atributos estava tornando a tag muito grande. Isso é totalmente válido. Se você fizer isso, é uma boa ideia indentar seus atributos e alinhá-los verticalmente, como eu fiz, para mantê-los agradáveis de ler. Em seguida, eu alinhei o símbolo de fechamento com o símbolo de abertura para ficar fácil identificar onde a tag de abertura começa e termina.

Agora você tem o poder de tornar seu

texto bom e hiper. Use este novo poder com responsabilidade. Até a próxima, meu nome é Davey Strus. Cá entre nós, eu provavelmente continuarei sendo o Davey Strus até mesmo depois da próxima. Mas até lá, boooooas codificações!

AULA 5 - HTML INCORPORANDO IMAGENS

[ASSISTA A AULA](#)



Leitura adicional

[Imagens no HTML no MDN](#)

[O elemento de incorporação de imagem no MDN](#)

[Guia de tipo e formato de arquivo de imagem no MDN \(em inglês\)](#)

[HTML images no w3schools \(em inglês\)](#)

[HTML tag no w3schools \(em inglês\)](#)

TRANSCRIÇÃO DO VÍDEO

Olá, pessoal. Sejam bem-vindos à Kenzie Academy. Meu nome é Davey Strus.

Vamos falar de imagens. É claro que HTML significa Linguagem de Marcação de Hipertextos, mas nossas páginas não estão limitadas a apenas textos. Nós podemos inserir outras mídias em nossas páginas. E qual o exemplo mais simples? Uma simples imagem. Para inserir uma imagem, use o elemento de inserção de imagem, .

``

O `` é diferente de todos os outros elementos que usamos até agora no corpo, no sentido de que é um elemento vazio. Quer dizer, ele não tem tag de fechamento. Usamos outro elemento vazio no cabeçalho: `<meta>`. Mas esta é a primeira vez que usamos um no corpo. E a tag de fechamento não é meramente opcional, ela é proibida. Ele simplesmente não possui tag de fechamento.

Contudo, ele possui atributos. Primeiro, o atributo source, src.

``

É aqui que você coloca a URL da imagem. Facinho. Falaremos mais sobre formatos de imagem depois, mas de longe os formatos de imagem mais comuns que você encontrará na web são JPGs, GIFs e PNGs.

Em seguida, temos o atributo alt.

``

O atributo alt fornece uma alternativa textual, caso o usuário esteja impossibilitado de ver a imagem por algum motivo. Pode ser porque está usando um leitor de tela, ou porque a URL está errada, ou porque a conexão está muito lenta. Seja como for, o valor aqui deve ser o texto que você

gostaria que aparecesse caso a imagem não possa ser vista. Não seja preguiçoso de colocar o nome do arquivo de imagem ou algo assim. Isso não ajuda muito. Pense sobre o que a imagem está tentando comunicar e converta isso em palavras.

```

```

Se a imagem for puramente decorativa, você pode dar um valor vazio ao atributo alt. Os leitores de tela irão ignorá-lo e as pessoas com conexões lentas simplesmente não verão nada. Isso é provavelmente mais útil do que ver ou ouvir as palavras "fronteira florida" ou algo do tipo.

Entretanto, você não deveria deixar o atributo alt totalmente desligado. Coloque algo útil nele ou aponte-o para uma string vazia se ele for puramente decorativo. Mais especificamente, a especificação do HTML 5 permite que o alt seja omitido sob certas circunstâncias, mas isso tem a ver com ferramentas que geram HTML automaticamente e coisas do tipo, e não com cenários que você mesmo encontrará ao escrever HTML. Você deveria considerar o alt como algo obrigatório.

Você também pode especificar a largura e a altura da imagem usando esses atributos.


```

```

Espera-se que os valores numéricos que você usar aqui sejam em pixels. Por padrão, a imagem aparecerá no tamanho que ela de fato é. Você não precisa dizer isso ao navegador. Ele irá entender. A vantagem de especificar a largura e altura é que um placeholder do tamanho correto irá aparecer enquanto a imagem está carregando. Sem isso, você verá o conteúdo mudando de lugar na página enquanto a imagem carrega. Não é o fim

do mundo, mas não é o adequado.

Isso é o básico. E como sempre, sinta-se à vontade para seguir os links nesta aula para aprender mais. Até a próxima, meu nome é Davey Strus. Boooooas codificações!

EXERCÍCIOS: IMAGENS E LINKS

VISÃO GERAL

Nesta atividade você irá inserir algumas imagens com subtítulos e criar links nelas.

REQUISITOS

Você tem bastante liberdade para escolher imagens sobre o que gosta, aqui estão as informações que são necessárias:

Um título de nível um (h1) com um nome apropriado.

Ao menos três imagens com subtítulos (h2-h6). Cada nível de título possui tamanhos diferentes, escolha com sabedoria qual usar.

Acesse o [exercício](#)

AULA 6: HTML TÍTULOS

Os elementos h1 - h6 são títulos.

Esses elementos têm uma classificação dada pelo número em seu nome. O <h1> elemento é dito ter o posto mais alto, o <h6> elemento tem o menor grau, e dois elementos com o mesmo nome têm o mesmo valor.

```
<h1>This is heading 1</h1>  
<h2>Esse é um cabeçalho 2</h2>  
<h3>Esse é um cabeçalho 3</h3>  
<h4>Esse é um cabeçalho 4</h4>  
<h5>Esse é um cabeçalho 5</h5>  
<h6>Esse é um cabeçalho 6</h6>
```

O código a seguir mostra alguns títulos com algum conteúdo abaixo deles.

```
<h1>Elementos de cabeçalho</h1>  
<h2>Sumário</h2>  
<p>Algum texto aqui...</p>
```

```
<h2>Exemplos</h2>  
<h3>Exemplo 1</h3>  
<p>Algum texto aqui...</p>
```

```
<h3>Exemplo 2</h3>  
<p>Algum texto aqui...</p>
```

```
<h2>Veja também</h2>  
<p>Algum texto aqui...</p>
```


QUESTÕES DE ACESSIBILIDADE

Uma técnica de navegação comum para usuários de software de leitura de tela é pular de um título para outro para determinar rapidamente o conteúdo da página. Por isso, é importante não pular um ou mais níveis de título. Fazer isso pode criar confusão, pois a pessoa que está navegando dessa forma pode ficar se perguntando onde está o rumo que está faltando.

Não fazer

```
<h1>Esse é um cabeçalho 1</h1>  
<h3>Esse é um cabeçalho 3</h3>  
<h4>Esse é um cabeçalho 4</h4>
```

Fazer

```
<h1>Esse é um cabeçalho 1</h1>  
<h2>Esse é um cabeçalho 2</h2>  
<h3>Esse é um cabeçalho 3</h3>
```

Os títulos podem ser aninhados como subseções para refletir a organização do conteúdo da página. A maioria dos leitores de tela também pode gerar uma lista ordenada de todos os títulos em uma página, o que pode ajudar uma pessoa a determinar rapidamente a hierarquia do conteúdo:

1. h1 Besouros
 1. h2 Etimologia
 2. h2 Distribuição e Diversidade
 3. h2 Evolução

- 1. h3 Paleozóico tardio
- 2. h3 jurássico
- 3. h3 Cretáceo
- 4. h3 Cenozóico
- 4. h2 Morfologia Externa
 - 1. h3 Cabeça
 - 1. h4 Partes da boca
 - 2. h3 Tórax
 - 1. h4 Protórax
 - 2. h4 Pterotórax
 - 3. h3 Pernas
 - 4. h3 Asas
 - 5. h3 Abdômen

Quando os títulos são aninhados, os níveis dos títulos podem ser “ignorados” ao fechar uma subseção.

EXEMPLOS:

[Tags de título w3schools](#)

[Exemplo](#)

AULA 7: HTML TAGS DE DIVISÃO DE CONTEÚDO

TAG ARTICLE

A tag `<article>` representa uma composição completa ou independente em uma página. Pode ser uma postagem num fórum, um artigo de revista ou jornal, um post de um blog ou mesmo um comentário de um usuário.

Quando as tags `<article>` são aninhados, os elementos internos representam artigos que estão relacionados ao conteúdo do artigo externo. Por exemplo, os comentários do post de um blog, podem ser implementados com a

tag article.

Este exemplo mostra uma postagem de blog usando a tag article:

```
<article>
```

```
  <header>
```

```
    <h1>A Primeira Regra da Vida</h1>
```

```
    <p><time>3 dias atrás</time></p>
```

```
  </header>
```

```
  <p>Se houver um microfone perto de  
  você, assumo-o</p>
```



```
<footer>  
  <a href="#">Mostrar comentários...</  
a>  
</footer>  
  
</article>
```

Você pode ver um pouco mais sobre a tag <article> nesse vídeo:

[Tag article do HTML 5](#)

TAG SECTION

A tag section representa uma seção genérica de uma página. Uma seção, neste contexto, é um agrupamento temático de conteúdo, por exemplo,

uma seção de serviços.

Exemplos

Exemplos de seções, seriam capítulos de um livro, ou seções numeradas de uma tese. Uma página de web pode ser dividida em seções, como por exemplo, uma para a introdução, uma para posts e outra para informações de contato.

No exemplo a seguir, vemos um artigo (parte de uma página da web) sobre maçãs, contendo duas seções curtas.

```
<article>  
  <p>A maçã é o fruto da macieira.</p>  
  <section>  
    <h1>Maça vermelha</h1>
```



```
<p>Essas maçãs vermelhas brilhantes  
são as mais comumente encontradas na  
maioria dos supermercados.</p>
```

```
</section>
```

```
<section>
```

```
<h1>Maça-verde</h1>
```

```
<p>Essas maçãs verdes e suculentas  
fazem um ótimo recheio para tortas de  
maça.</p>
```

```
</section>
```

```
</article>
```

Aqui está um programa de graduação
com duas seções, uma para a lista de
formandos e outra para a descrição da
cerimônia.

```
<!DOCTYPE Html>
```

```
<html Lang=En>
```

```
<head>
```

```
<title>Cerimônia de Graduação Verão  
2022</title>
```

```
</head>
```

```
<body>
```

```
<h1>Graduação</h1>
```

```
<section>
```

```
<h1>Cerimônia</h1>
```

```
<p>Procissão de Abertura</p>
```

```
<p>Discurso do Orador da Turma</p>
```

```
<p>Discurso do Presidente da Turma</
```

```
p>
```

```
<p>Apresentação dos Diplomas</p>
```

```
<p>Discurso de Fechamento do  
Diretor</p>
```

```
</section>
```



```
<section>
  <h1>Graduados</h1>

  <ul>
    <li>Molly Carpenter</li>
    <li>Anastasia Luccio</li>
    <li>Ebenazar McCoy</li>
    <li>Karrin Murphy</li>
    <li>Thomas Raith</li>
    <li>Susan Rodriguez</li>
  </ul>
</section>

</body>
</html>
```

Exemplo de uso simples

Antes

```
<div>
  <h2>Cabeçalho</h2>
  <p>Amontoado de conteúdos maneiros</p>
</div>
```

Depois

```
<section>
  <h2>Cabeçalho</h2>
  <p>Amontoado de conteúdos maneiros</p>
</section>
```

Conforme mencionado acima, `<section>` é um elemento de corte genérico e só deve ser usado se não houver um elemento

mais específico para representá-lo. Por exemplo, um menu de navegação deve ser envolvido em uma tag `<nav>`, mas uma lista de resultados de pesquisa ou uma exibição de mapa e seus controles não possuem elementos específicos e podem ser colocados dentro de uma `<section>`.

[Tag section do HTML 5](#)

Considere também estes casos:

Se o conteúdo do elemento representa uma unidade atômica autônoma de conteúdo como, por exemplo, uma postagem de blog, um comentário ou um artigo de jornal), o `<article>` seria uma escolha melhor.



Você pode ver um pouco mais sobre a tag `<section>` nesse vídeo:

PORTFÓLIO BÁSICO

VISÃO GERAL

Nesta atividade, você irá criar uma página de portfólio básica.

REQUISITOS

Para realizar esse exercício você vai acessar a plataforma codepen. O link está no fim desse página.

Em sua biografia, você tem bastante liberdade em relação ao conteúdo que deseja compartilhar. Aqui estão os elementos que devem ser incluídos:
Um título de nível um(h1) com um nome apropriado para a sua biografia.

Ao menos três títulos de níveis inferiores

(h2-h6), fazendo com que na sequência de cada um seja introduzido no mínimo 3 parágrafos(p). Cada nível de título possui tamanhos diferentes, escolha com sabedoria qual usar.

Um link para um site que você visita com frequência.

Acesse o [exercício](#)

MÓDULO B - CSS BÁSICO

AULA 1 - INTRODUÇÃO AO CSS

Neste módulo, você irá aprender a usar o CSS para manipular a aparência do conteúdo das suas páginas web.

O QUE REALMENTE É CSS?

CSS (Folha de Estilo em Cascata) é o código que você usa para dar estilo à sua página.

O CSS não é uma linguagem de programação mas sim uma linguagem de estilização. Isso significa que o CSS permite aplicar estilos seletivamente a elementos

em documentos HTML. Por exemplo, para selecionar todos os elementos parágrafo de uma página HTML e tornar o texto dentro deles vermelho, você escreveria este CSS:

```
p {  
  color: red;  
}
```

Anatomia de um conjunto de regras CSS

Vamos dar uma olhada no CSS acima com mais detalhes:



Toda essa estrutura é chamada de conjunto de regras, mas geralmente usamos o termo “regra”, por ser mais curto. Note os nomes das partes individuais:

Seletor(Selector)

É, nesse caso, o nome do elemento HTML. Ele seleciona o(s) elemento(s) a serem

estilizados(elementos `<p>`). Para dar estilo a um outro elemento, é só mudar o seletor.

Propriedades(Property)

Forma pela qual você altera um determinado estilo de um elemento HTML. No CSS, você escolhe quais propriedades você deseja alterar.

Valor da propriedade(Property value)

À direita da propriedade, depois dos dois pontos, nós temos o valor de propriedade, que escolhe uma dentre muitas aparências possíveis para uma determinada propriedade.

Declaração(Declaration)

Uma regra simples como `color: red;` especificando quais das propriedades do

elemento você quer estilizar.

Note outras partes importantes da sintaxe:

Cada linha de comando deve ser envolvida em chaves({})

Dentro de cada declaração, você deve usar dois pontos(:) para separar a propriedade de seus valores

Dentro de cada conjunto de regras, você deve usar um ponto e vírgula(;) para separar cada declaração da próxima

Então para modificar múltiplos valores de propriedades de uma vez, você deve escrevê-los separados por ponto e vírgula, desse modo:

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

Selecionando múltiplos elementos

Você também pode selecionar vários tipos de elementos e aplicar um único conjunto de regras a todos eles. Inclua múltiplos seletores separados por vírgulas. Por exemplo:

```
p, li, h1 {  
  color: red;  
}
```


AULA 2 - SELETORES NO CSS

Neste módulo, nós vamos nos aprofundar um pouco mais nos seletores de CSS.

SELETORES CSS

Em CSS, os seletores são usados para direcionar os elementos HTML em nossas páginas da web que queremos estilizar. Há uma grande variedade de seletores CSS disponíveis, permitindo uma precisão refinada ao selecionar os elementos a serem estilizados.

Um seletor CSS é a primeira parte de uma regra CSS. É um padrão de elementos e outros termos que informam ao

navegador quais elementos HTML devem ser selecionados para que os valores de propriedade dentro da regra sejam aplicados a eles.

```
h1 {  
    color: blue;  
    background-color: yellow;  
}  
  
p {  
    color: red;  
}
```


Tipos de seletores

Existem alguns agrupamentos diferentes de seletores e saber qual tipo de seletor você pode precisar o ajudará a encontrar a ferramenta certa para o trabalho.

Seletores por nome de elemento
Seleciona elementos através do seu nome:

```
/* Todos os elementos <a> */  
a {  
  color: red;  
}
```

Seletores por classe

Perceba que para selecionar um elemento

através da sua classe, você precisa adicionar um ponto(.) antes do nome da classe:

```
.titulo{  
  /* declarações aqui */  
}
```

Seletores por ID

Assim como na classe você precisa de um ponto, para utilizar o seletor por ID, você precisa adicionar uma cerquilha(#) antes do nome do ID:

```
#tituloUnico{  
  /* declarações aqui */  
}
```


Seletores por atributo

Os elementos HTML podem ter atributos que fornecem mais detalhes sobre o elemento que está sendo marcado. Em CSS, você pode usar seletores de atributo para estilizar elementos com determinados atributos. Esta lição mostrará como usar esses seletores.

Seletores de presença e valor
Esses seletores permitem a seleção de um elemento com base na presença de um atributo sozinho(por exemplo, href) ou mesmo através do atributo juntamente com o valor desse.

Seletor	Exemplo	Descrição
[attr]	a[title]	Corresponde a elementos que possuem um atributo " attr ", cujo nome é o valor entre colchetes(title)
[attr=value]	a[href="https://example.com"]	Corresponde a elementos com um atributo " attr " cujo valor é exatamente o " value ", a string entre aspas
[attr~=value]	p[class~="special"]	Corresponde a elementos com um atributo " attr " cujo valor é exatamente o " value " ou contém o valor em sua lista de valores, separados por espaço
[attr =value]	div[lang =“zh”]	Corresponde a elementos com um atributo " attr " cujo valor é exatamente o " value " ou começa com " value " imediatamente seguido por um hífen.

No exemplo abaixo você pode ver esses seletores sendo usados:

Usando li[class] podemos combinar qualquer seletor com um atributo de classe. Isso corresponde a todos os itens da lista, exceto o primeiro.
li[class="a"] corresponde a um seletor com uma classe de a, mas não a um seletor com uma classe de a com outra classe separada

por espaço como parte do valor. Ele seleciona o segundo item da lista. `li[class~="a"]` corresponderá a uma classe de a, mas também a um valor que contém a classe de a como parte de uma lista separada por espaços em branco. Ele seleciona o segundo e o terceiro itens da lista.

Seletores de presença e valor

Esses seletores permitem a seleção de um elemento com base na presença de um atributo sozinho (por exemplo, href) ou mesmo através do atributo juntamente com o valor desse.

Attribute presence and value selectors

- Item 1
- Item 2
- Item 3
- Item 4

```
li[class] {  
  font-size: 200%;  
}  
  
li[class="a"] {  
  background-color: yellow;  
}  
  
li[class~="a"] {  
  color: red;  
}
```

```
<h1>Attribute presence and value selectors</h1>  
<ul>  
  <li>Item 1</li>  
  <li class="a">Item 2</li>  
  <li class="a b">Item 3</li>  
  <li class="ab">Item 4</li>  
</ul>
```


Seletores de substring

Esses seletores permitem uma correspondência mais avançada de substrings dentro do valor do seu atributo. Por exemplo, se você tivesse classes de box-warning e box-error e quisesse combinar tudo que começou com a string "box-", você poderia usar [class^="box-"] para selecionar os dois(ou [class|="box"] como descrito abaixo)

Seletor	Exemplo	Descrição
[attr^=value]	li[class^="box-"]	Corresponde a elementos com um atributo attr (cujo nome é o valor entre colchetes), cujo valor começa com valor.
[attr\$=value]	li[class\$="-box"]	Corresponde a elementos com um atributo attr cujo valor termina com valor.
[attr*=value]	li[class*="box"]	Corresponde a elementos com um atributo attr cujo valor contém o valor em qualquer lugar dentro da string.

O próximo exemplo mostra o uso desses seletores:

li[class^="a"] corresponde a qualquer valor de atributo que comece com a, portanto, corresponde aos dois primeiros itens da lista.

li[class\$="a"] corresponde a qualquer valor de atributo que termina com a, portanto, corresponde ao primeiro e terceiro item da lista.

li[class*="a"] corresponde a qualquer valor de atributo onde a apareça em qualquer lugar na string, portanto, corresponde a todos os itens da nossa lista

Attribute substring matching selectors

- Item 1
- Item 2
- Item 3
- Item 4

```
li[class^="a"] {  
    font-size: 200%;  
}  
  
li[class$="a"] {  
    background-color: yellow;  
}  
  
li[class*="a"] {  
    color: red;  
}
```

```
<h1>Attribute substring matching selectors</h1>  
<ul>  
    <li class="a">Item 1</li>  
    <li class="ab">Item 2</li>  
    <li class="bca">Item 3</li>  
    <li class="bcabc">Item 4</li>  
</ul>
```

Seletores de substring

Se você deseja combinar valores de atributo sem distinção entre maiúsculas e minúsculas, você pode usar o valor `i` antes do colchete de fechamento. Este sinalizador informa ao navegador para corresponder caracteres ASCII sem distinção entre maiúsculas e minúsculas. Sem o sinalizador, os valores serão correspondidos de acordo com a distinção entre maiúsculas e minúsculas do idioma do documento - no caso do HTML, será sensível a maiúsculas e minúsculas.

No exemplo a seguir, o primeiro seletor corresponderá a um valor que começa

com um - ele corresponde apenas ao primeiro item da lista porque os outros dois itens da lista começam com um A maiúsculo. O segundo seletor usa o sinalizador que não diferencia maiúsculas de minúsculas e, portanto, corresponde a todos os itens da lista.

Case-insensitivity

- Item 1
- Item 2
- Item 3

```
li[class^="a"] {  
    background-color: yellow;  
}  
  
li[class^="a" i] {  
    color: red;  
}
```

```
<h1>Case-insensitivity</h1>  
<ul>  
    <li class="a">Item 1</li>  
    <li class="A">Item 2</li>  
    <li class="Ab">Item 3</li>  
</ul>
```

 Para saber mais

Leia: [Seletores CSS](#)

EXERCÍCIO - UTILIZANDO OS SELETORES CSS

VISÃO GERAL

Nos conteúdos anteriores você aprendeu a utilizar alguns seletores. Hora de praticar!

Abra o arquivo [CodePen](#) e faça as alterações necessárias no arquivo HTML e CSS para obter o seguinte resultado:

Kenzie Academy Brasil

Sobre:

A Kenzie Academy é uma escola de codificação e design em Indianapolis, Indiana. Nosso programa de 6 meses a 2 anos com aprendizado pago de 1 ano é uma nova alternativa às faculdades tradicionais e aos bootcamps de codificação de curto prazo.

Programas oferecidos:

1. [Software Engineering](#)
2. [UX Engineering](#)
3. [Digital Marketing](#)

Kenzie está empolgada com os [alunos](#).

Kenzie Academy Website

Saiba mais sobre a Kenzie Academy [here](#).

EXERCÍCIOS:

1. Adicione uma classe 'mainTitle' no seletor h1.
2. Adicione uma classe 'text' no primeiro parágrafo.
3. Altere a cor do item "3. Digital Marketing" para rosa(dica: você pode utilizar a keyword 'pink').
4. Encontre o erro no seletor css 'knowMore'; (dica: declaração com classe).
5. Atribua um id 'excited' a tag span dentro do elemento div 'knowMore'.

AULA 3 - UNIDADES DE MEDIDA

[ASSISTA A AULA](#)

VISÃO GERAL

CSS tem várias opções de unidades para se usar ao determinar o tamanho de alguma propriedade no CSS.

Aprender todas as suas opções para unidades CSS pode ser fundamental para estilizar de uma melhor forma e o layout tenha uma ótima aparência em qualquer tela.

O que é uma unidade de medida?

O que é uma unidade de medida?

Uma unidade CSS determina o tamanho de uma propriedade que você está definindo para um elemento ou conteúdo. Por exemplo, se você deseja definir a margem de propriedade de um parágrafo, você deve dar a ela um valor específico. Este valor inclui a unidade CSS.

Vamos ver um exemplo:

```
p {  
  margin: 20px;  
}
```

Nesse caso, margin é uma propriedade, e 20px é o valor, onde px é a unidade de medida.

Embora seja comum ver unidades como o px sendo usadas, a grande questão é: "Qual é a melhor unidade para se usar aqui?"

Aqui estão algumas considerações a serem feitas ao escolher um tipo de

unidade e exemplos de casos de uso:

Unidades absolutas vs. Unidades relativas

Ao considerar todas as opções para quais unidades usar, é importante considerar as duas categorias de unidades: absolutas e relativas.

Unidades absolutas

As unidades que são "absolutas" têm o mesmo tamanho, independentemente do elemento pai ou do tamanho da janela. Isso significa que uma propriedade definida com um valor

que tem uma unidade absoluta terá esse tamanho quando vista em um dispositivo móvel ou computador.

As unidades absolutas podem ser úteis ao trabalhar em um projeto onde a capacidade de resposta não está sendo considerada. Por exemplo, aplicativos de desktop que não podem ser redimensionados podem ser estilizados para as dimensões padrão. Se a janela não escalar, você também não precisa que o conteúdo escale.

Dica: unidades absolutas podem ser menos favoráveis para sites responsivos porque não escalam

Unidade	Descrição	Exemplo
px	1/96 de 1 polegada (96px = 1 polegada)	font-size: 12px;
pt	1/72 de 1 polegada (72pt = 1 polegada)	font-size: 12pt;
pc	12pt = 1pc	font-size: 1.2pc;
cm	centímetro	font-size: 0.6cm;
mm	milímetro (10 mm = 1 cm)	font-size: 4mm;
in	polegadas	font-size: 0.2in;

Pixels (px) são normalmente a unidade absoluta mais popular para telas. Centímetros, milímetros e polegadas são mais comuns para impressão e você pode nem saber que eram opções!

Unidades relativas

As unidades relativas são úteis para estilizar sites responsivos porque são dimensionadas em relação ao pai ou ao tamanho da janela(dependendo da unidade).

Como regra geral, as unidades relativas podem ser usadas como padrão para sites responsivos. Isso pode ajudá-lo a evitar a necessidade de atualizar estilos para diferentes tamanhos de tela.

As unidades relativas podem ser um pouco mais difíceis do que as unidades absolutas para determinar qual usar,

então vamos examinar suas opções em detalhes.

Unidade	Descrição
%	Relativo ao valor do elemento pai dessa propriedade
em	Relativo ao tamanho da fonte atual do elemento
rem	Relativo ao tamanho da fonte do root da página(por exemplo, o elemento <code><html></code>). "Rem" = root em"
vh	Relativo à altura da janela de visualização(desktop ou mobile). 1vh = 1/100 da altura da janela de visualização
vw	Relativo à largura da janela de visualização. 1vw = 1/100 da largura da janela de visualização.
vmin	Relativo à menor dimensão da janela de visualização(por exemplo, para orientação de retrato, a largura é menor do que a altura, portanto, é relativa à largura). 1vmin = 1/100 da dimensão menor da janela de visualização.
vmax	Relativo à maior dimensão da janela de visualização(por exemplo, altura para orientação de retrato). 1vmax = 1/100 da dimensão maior da janela de visualização.

Nem sempre está claro qual dessas opções é melhor usar para cada tipo de propriedade CSS. Por exemplo, % é geralmente mais apropriado para propriedades relacionadas ao layout, como largura, do que para tamanho da

fonte.

Aqui estão alguns exemplos de quando você usaria cada unidade relativa:

%: Você deseja que um elemento filho tenha 10% da largura do pai como margem para que nunca preencha todo o elemento pai. Se o tamanho do pai mudar, a margem também será atualizada.

```
.filho {  
    margin: 10%;  
}
```

em: Você deseja que a fonte de um elemento filho tenha metade do tamanho da fonte de seu pai (por exemplo, o parágrafo sob o título de uma seção).

```
.filho {  
    font-size: 0.5em;  
}
```

rem: O tamanho da fonte deve ser o dobro do tamanho da fonte do elemento raiz. Pode ser assim que você dimensiona seus cabeçalhos, porque todos eles devem ter o mesmo tamanho, independentemente do contêiner pai.


```
header {  
  font-size: 2rem;  
}
```

vh: Sua página deve sempre ter a altura da janela de visualização.

```
body {  
  height: 100vh;  
}
```

vw: Você tem uma seção com texto que deve ter a metade da largura da janela de visualização.

```
section {  
  width: 50vw;  
}
```

vmin: Você tem uma imagem que deve ser sempre tão larga quanto a menor dimensão da janela de visualização. Em um dispositivo móvel no modo retrato, a imagem será tão ampla quanto a largura da janela de visualização.

```
img {  
  width: 100vmin;  
}
```

vmax: Você não se importa se uma imagem for cortada porque você deseja que ela preencha completamente a dimensão maior da janela de visualização. Por exemplo, se a imagem de um padrão for usada como plano de fundo.


```
img {  
  width: 100vmax;  
}
```

No geral, quando e como você escolhe suas unidades CSS se resumem a algumas perguntas:

Eu quero que o meu estilo seja dimensionado quando o tamanho da janela de visualização mudar?
Se eu quiser que ele seja escalonado, ao que eu quero que ele seja relativo?
Depois de responder a essas perguntas, é muito mais fácil definir qual unidade usar!



Para saber mais

Leia: [Conheça as Unidades de CSS](#)

AULA 4 - APLICANDO COR A ELEMENTOS COM CSS

Nessa aula vamos ver um pouco sobre as cores no css, propriedades como color, background, que podem ser útil nas próximas atividades, e também páginas para seleção de paleta de cores.

COISAS QUE PODEM TER COR

No nível do elemento, tudo em HTML pode ter uma cor aplicada a ele. Em vez disso, vamos olhar para as coisas em termos dos tipos de coisas

desenhadas nos elementos, como texto e bordas e assim por diante. Para cada um, veremos uma lista das propriedades CSS que aplicam cores a eles.

PROPRIEDADES

Sempre que um elemento é renderizado, essas propriedades são usadas para determinar a cor do texto, seu plano de fundo e quaisquer decorações no texto.

color

Propriedade utilizada para definir a cor de um texto

A cor a ser usada ao desenhar o texto e quaisquer decorações de texto, como a adição de sublinhados ou sobrelinhados e assim por diante.

background-color

A cor de fundo de um elemento, seja ele do tipo em linha ou em bloco.

text-shadow

Define um efeito de sombra para o texto. Entre as opções de configuração para a sombra, está a cor dessa sombra.

border

Essa propriedade é utilizada para definir uma borda para um elemento, dentre as suas configuração está a cor dessa borda.

opacity

Propriedade utilizar para regular a transparência de um elemento, ou seja, deixá-lo mais opaco ou mais transparente.

VALORES

As cores no CSS podem ser aplicadas tanto através de uma palavra chave, ou seja, o nome da cor, quanto através de valores numéricos, utilizando as

unidades hexadecimal ou rgb, há também outras formas, porém essas são as mais comuns.

Vamos utilizar como exemplo a cor preta:

Palava chave: **black**

Hexadecimal: **#000000**

RGB: **rgb(0, 0, 0)**



Para saber mais

Leia: [Propriedade color do CSS - MDN](#)

Leia: opacity - [MDN](#)

Leia (em inglês): [CSS Colors - W3Schools](#)

Leia (em inglês): [CSS RGB Colors - W3Schools](#)

Leia (em inglês): [CSS HEX Colors - W3Schools](#)

Paletas

[Paletton](#)

[Coolors](#)

[Adobe Color](#)

Conteúdo opcional

Leia (em inglês): [Applying color to HTML elements using CSS](#)

Leia (em inglês): [CSS Color Module Level 3](#)

Leia (em inglês): [color - MDN](#)

Leia(em inglês): [opacity - MDN](#)

AULA 5 - FONTES CSS

[ASSISTA A AULA](#)

ESTILIZANDO FONTES

Nós temos algumas opções de estilização de fontes, desde o seu tamanho até família dessa fontes. Vamos ver algumas propriedades e como utilizá-las!

font-family

A propriedade font-family especifica uma lista de um ou mais nomes de famílias de fontes e / ou categorias de

família para o elemento selecionado:

```
p {  
    font-family: Georgia, serif;  
}
```

Esse código resultará no seguinte:

Almost before we knew it, we had
left the ground.

As aspas são necessária quando o nome da família é composto, ou seja, tem mais um nome

Os valores são separados por vírgulas para indicar que são alternativas. O navegador selecionará a primeira fonte da lista, se não encontrar a primeira ele utiliza a segunda e assim por diante.

Você deve sempre incluir pelo menos uma categoria de família em uma lista de famílias de fontes, uma vez que não há garantia de que qualquer fonte esteja disponível. Isso permite que o navegador selecione uma fonte alternativa quando necessário.

Valores

Há duas opções de valores, definir apenas a categoria de família da fonte ou definir uma fonte específica.

Categorias de famílias

Quando definimos apenas a categoria, o navegador aplica uma fonte dessa categoria que ela tenha por padrão em sua memória

Temos algumas opções de categorias de fontes. As principais são:

serif

A fonte do tipo serif é aquela que tem uma 'perninha' no início e final da letra.

Alguns exemplos de fonte do tipo serif são Times New Roman e Georgia, como vimos no exemplo acima.

sans-serif

A fonte do tipo sans-serif é o contrário da serif, ou seja, 'sem serifa', como o próprio nome já mostra.

Alguns exemplos, são Arial, Roboto e Gill Sans, que vimos no exemplo acima.

monospace

Na fonte do tipo monospace, todos os caracteres passam a ocupar exatamente o mesmo espaço. Muito utilizada em IDE's.

Um exemplo de fonte monospace é a Roboto Mono:

```
Almost before we  
knew it, we had left  
the ground.
```

cursive

A fonte do tipo cursive é aquela que possui um design mais sofisticado, com estilo de manuscrito.

Um exemplo de fonte cursive é a Pacifico:

*Almost before we knew it,
we had left the ground.*

font-size

A propriedade CSS font-size define o tamanho da fonte.

Valores

Há algumas opções de valores para essa propriedade. A mais utilizada é um valor numérico, seguido de uma unidade de medida como px, em ou rem, que são as mais utilizadas se tratando de fontes.

font-weight

A propriedade CSS font-weight define a espessura da fonte. Os pesos disponíveis dependem da família de fontes atualmente definida.

Valores

normal

Espessura normal, equivalente aos valores 400 ou mesmo 500

bold

Negrito. Equivalente aos valores 600 ou mesmo 700.

lighter

Um peso de fonte relativo mais leve do que o elemento pai.

bolder

Um peso de fonte relativo mais pesado que o elemento pai.

valor numérico que varia de 100 a 900,

sem a utilização de unidades de medida

line-height

A propriedade CSS line-height define a altura de uma caixa de linha. É comumente usado para definir a distância entre as linhas de texto.

Em elementos de nível de bloco, ele especifica a altura mínima das caixas de linha dentro do

Valores

Principalmente valores numéricos, seguido ou não de uma unidade de medida. Quando é aplicado um valor sem unidade, ele fica relativo ao tamanho da fonte do texto no qual

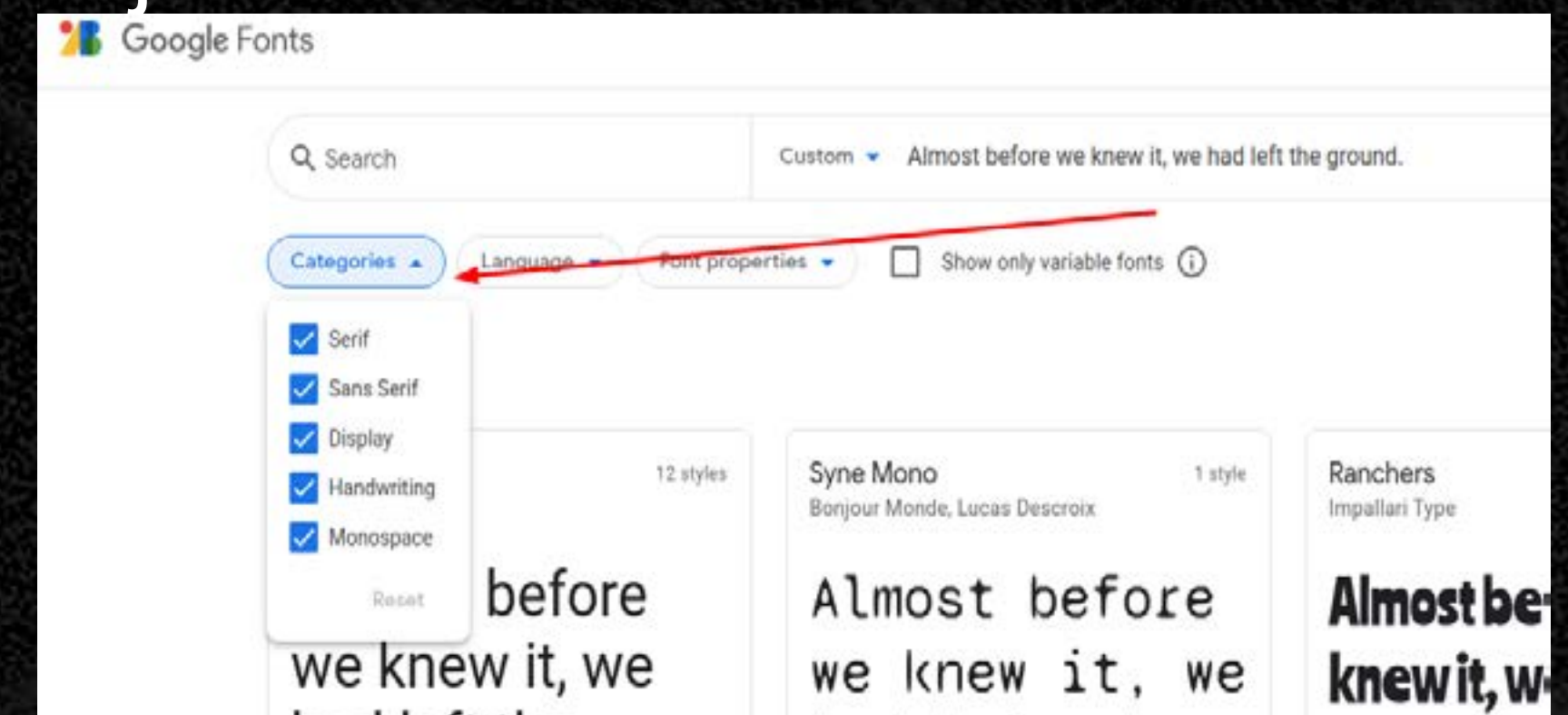
está aplicando o line-height

IMPORTANDO FONTES

Como utilizar o Google Fonts

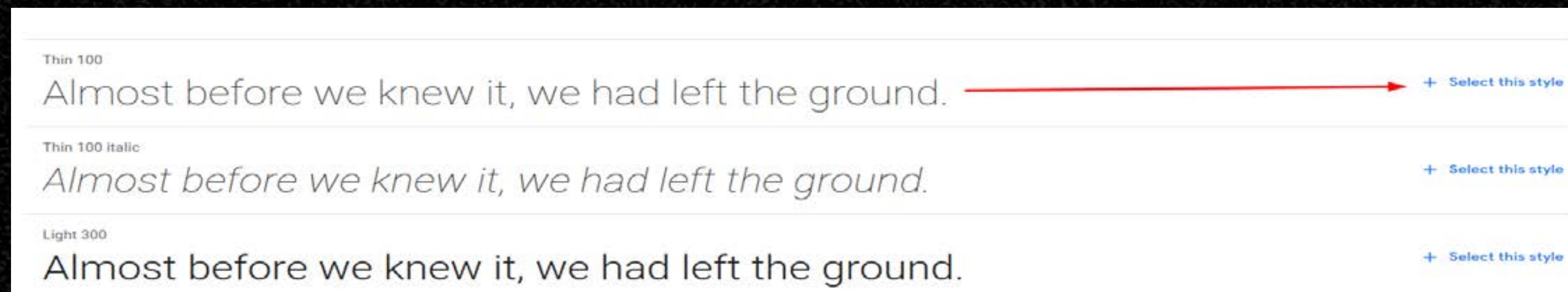
Acesse o website da Google Fonts, através da url fonts.google.com

Pesquise pela fonte desejada ou procure por alguma rolando a página inicial. Você também pode filtrar pela categoria desejada.



1. Selecione a fonte desejada.

2. Selecione os estilos dessa fonte que deseja utilizar, clicando em “Select this style”, ao lado do estilo.



1. Depois é só importar no seu website, utilizando o código disponibilizado pela plataforma, e então, aplica no seu CSS.

Review

Roboto ^

Thin 100 ⊖

[Add more styles](#) [Remove all](#)

Use on the web

To embed a font, copy the code into the <head> of your html

☒ <link> ☐ @import

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Roboto', sans-serif;
```



Para saber mais

Leia: [Estilizando Fontes com CSS](#)

EXERCÍCIO - TRABALHANDO COM FONTES

Neste exercício vamos trabalhar com a importação de fontes e edição das suas propriedades:

Acesse o [exercício](#)

AULA 6 - DIMENSIONAMENTO

Nessa aula vamos ver um pouco sobre o dimensionamento no css, e as principais propriedades utilizadas para dimensionar elementos, como width, height, max-width, min-width, max-height e min-height

PROPRIEDADES

width

A propriedade width define a largura de um elemento.

min-width

A propriedade min-width define uma largura mínima para um

elemento. Impede que o valor usado da propriedade width se torne menor do que o valor especificado para largura mínima.

max-width

A propriedade max-width define a largura máxima de um elemento. Impede que o valor usado da propriedade width se torne maior do que o valor especificado para a largura máxima.

height

A propriedade height especifica a altura de um elemento.

min-height

A propriedade min-height define uma altura mínima para um elemento.

Isso evita que o valor usado na propriedade height se torne menor do que o valor especificado para a altura mínima.

max-height

A propriedade max-height define a altura máxima de um elemento.

Isso evita que o valor usado da propriedade height se torne maior do que o valor especificado para a altura máxima.

VALORES

Para as propriedades relacionadas

a largura: Normalmente, valores numéricos, seguido de uma unidade de medida. A mais utilizada são px, % ou vw.

Para as propriedades relacionadas a altura: Normalmente, valores numéricos, seguido de uma unidade de medida. A mais utilizada são px ou vh.



Para saber mais

Leia: [Propriedade width](#)

Leia: [Propriedade height](#)

AULA 7 - POSICIONAMENTO

Nessa aula vamos ver um pouco sobre posicionamento utilizando a propriedade display.

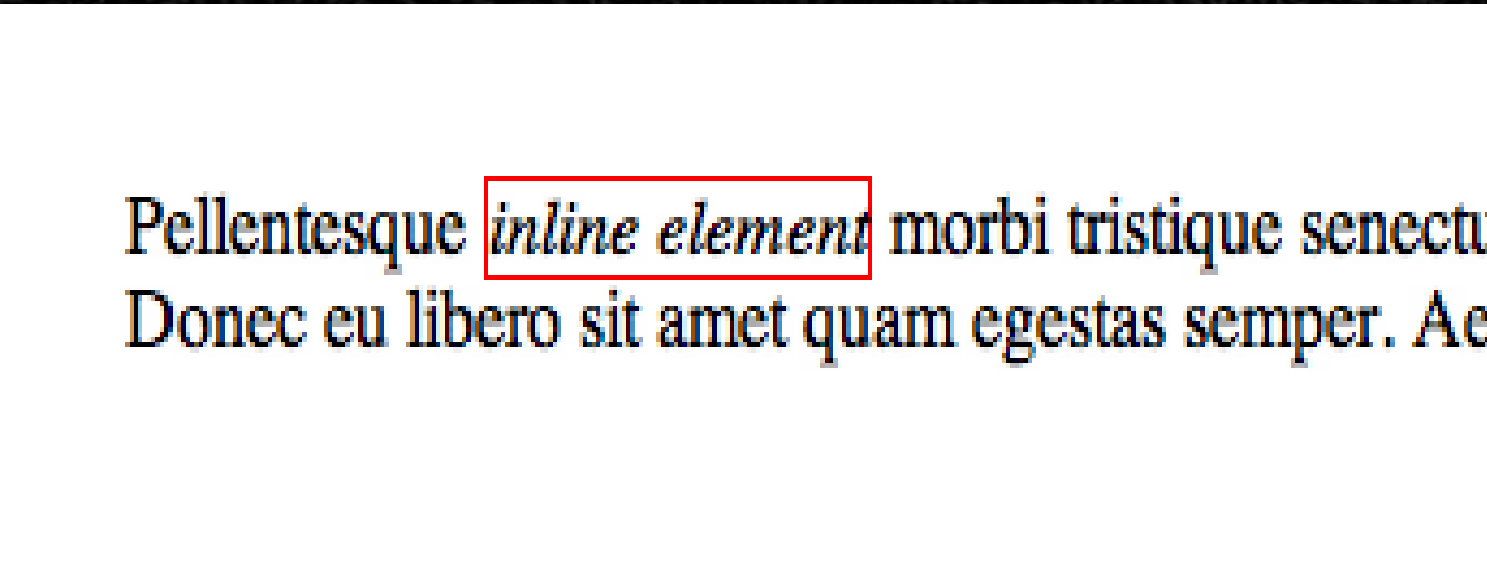
O DISPLAY

Todo elemento em uma página da web é uma caixa retangular. A propriedade display no CSS determina apenas como essa caixa se comporta. Existem alguns valores que são comumente usados:

inline

Pense em elementos como ``, `` ou `` e como envolver o

texto nesses elementos em uma string de texto não interrompe o fluxo do texto.



Pellentesque *inline element* morbi tristique senectus
Donec eu libero sit amet quam egestas semper. Aenean

O elemento `` possui uma borda vermelha de 1px. Observe que ele está alinhado com o resto do texto.

Um elemento inline irá aceitar margin e padding, mas o elemento ainda fica

embutido como você pode esperar. A margem e o preenchimento apenas afastarão outros elementos horizontalmente, não verticalmente.

Pellentesque *inline element* morbi tristique senectus
Donec eu libero sit amet quam egestas semper. Aenean

Um elemento inline não aceita height e width. Ele simplesmente irá ignorar.

inline-block

Um elemento definido como inline-block é muito semelhante ao inline no

sentido de que será definido em linha com o fluxo natural do texto (na “base da linha”). A diferença é que você pode definir uma largura e altura.

Pellentesque

*inline
block*

*inline
block*

*inline
block*

morbi tristique

senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

block

Vários elementos são definidos como block pela folha de estilo do navegador. Eles geralmente são elementos de contêiner, como <div>, <section> e . Também “blocos

de texto” como `<p>` e `<h1>`. Os elementos de nível de bloco não ficam em linha, mas passam por eles. Por padrão(mesmo sem definir uma largura), eles ocupam todo o espaço horizontal disponível.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

hi

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.



Para saber mais

Leia: [Pare de chutar e aprenda como funciona o `display: block`](#)



**Kenzie
Academy**