

## 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
  </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
    </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: `<span>`. 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.