

```
// O mesmo que const sin = math.sin, cos = math.cos, tan = math.tan
```

```
const {sin, cos, tan} = matemática;
```

Observe no código aqui que o objeto de matemática tem muitas propriedades outras do que os três que são destruídos em variáveis ??individuais. Aqueles que não são nomeados são simplesmente ignorados. Se o lado esquerdo deste

A tarefa incluiu uma variável cujo nome não era uma propriedade de Matemática, essa variável seria simplesmente designada indefinida.

Em cada um desses exemplos de destruição de objetos, escolhemos nomes variáveis ??que correspondem aos nomes de propriedades do objeto que somos destruição. Isso mantém a sintaxe simples e fácil de entender, mas

não é necessário. Cada um dos identificadores no lado esquerdo de um

A atribuição de destruição de objetos também pode ser um par de separação de cólon de identificadores, onde o primeiro é o nome da propriedade cujo valor é ser atribuído e o segundo é o nome da variável para atribuí -la a:

```
// O mesmo que const cosseno = math.cos, tangente = math.tan;
```

```
const {cos: cosseno, tan: tangente} = matemática;
```

Acho que a sintaxe de destruição de objetos se torna muito complicada para ser

Útil quando os nomes de variáveis ??e nomes de propriedades não são os mesmos,

E eu tendem a evitar a abreviação neste caso. Se você optar por usá -lo,

Lembre -se de que os nomes de propriedades estão sempre à esquerda do cólon, em ambos os literais de objeto e à esquerda de um objeto destrutivo atribuição.

A tarefa de destruição se torna ainda mais complicada quando é

usado com objetos aninhados, ou matrizes de objetos, ou objetos de matrizes, mas é legal: