

```
vamos [x, ... y] = [1,2,3,4];// y == [2,3,4]
```

Veremos três pontos usados ?? dessa maneira novamente no §8.3.2, onde eles são usados para indicar que todos os argumentos de função restantes devem ser coletados em uma única matriz.

A tarefa de destruição pode ser usada com matrizes aninhadas. Nesse caso,

O lado esquerdo da tarefa deve parecer uma matriz aninhada

literal:

```
vamos [a, [b, c]] = [1, [2,2.5], 3];// a == 1;b == 2;c ==
```

2.5

Uma característica poderosa da destruição de matrizes é que ela realmente não

requer uma matriz! Você pode usar qualquer objeto iterável (capítulo 12) no

Lado RightHands da tarefa; qualquer objeto que possa ser usado com um

para/de loop (§5.4.4) também pode ser destruturado:

```
vamos [primeiro, ... descansar] = "olá";// primeiro == "h";descanso ==
```

```
["E", "L", "L", "O"]
```

A tarefa de destruição também pode ser realizada quando a direita

lado é um valor de objeto. Nesse caso, o lado esquerdo da tarefa

Parece algo como um objeto literal: uma lista separada por vírgula

nomes variáveis ?? dentro de aparelhos encaracolados:

```
Seja transparente = {r: 0.0, g: 0,0, b: 0,0, a: 1,0};// A RGBA
```

cor

```
Seja {r, g, b} = transparente;// r == 0,0;g == 0,0;b == 0.0
```

O próximo exemplo copia as funções globais do objeto de matemática em

Variáveis, que podem simplificar o código que faz muita trigonometria: