O método flatmap () funciona como o método map () (ver
"Map ()"), exceto que a matriz retornada é achatada automaticamente como se
passou para o plano ().Isto é, chamar A.flatmap (f) é o mesmo que
(mas mais eficiente que) a.map (f) .flat ():
Deixe frases = ["Hello World", "The Definitive Guide"];
deixe palavras = phrases.flatmap (frase => phrase.split (""));
palavras // => ["Olá", "mundo", "o", "definitivo", "guia"];
Você pode pensar em Flatmap () como uma generalização de mapa () que
permite cada elemento da matriz de entrada para mapear para qualquer número de
elementos da matriz de saída.Em particular, Flatmap () permite que você
mapear elementos de entrada para uma matriz vazia, que se achata para nada no
Matriz de saída:
// mapear números não negativos para suas raízes quadradas
[-2, -1, 1, 2] .flatmap (x => x <0? []: Math.sqrt (x)) // =>
[1, 2 ** 0,5]
7.8.3 Adicionando matrizes com concat ()
O método concat () cria e retorna uma nova matriz que contém
os elementos da matriz original em que concat () foi invocado,
seguido por cada um dos argumentos para concat ().Se algum deles
argumentos é uma matriz, então são os elementos da matriz que são
Concatenado, não a própria matriz.Observe, no entanto, que concat () faz
Não achatar recursivamente matrizes de matrizes.concat () não modifica
A matriz em que é invocada:
Seja a = [1,2,3];
A.Concat (4, 5) // => [1,2,3,4,5]
a.Concat ([4,5], [6,7]) // => [1,2,3,4,5,6,7];Matrizes são