

Todos os elementos da matriz:

Seja `a = [1,2,3,4,5]`;

`a.every (x => x < 10) // => true`: Todos os valores são <10.

`a.every (x => x % 2 === 0) // => false`: nem todos os valores são até.

O método de `alguns ()` é como o quantificador matemático "existe"

?: ele retorna verdadeiro se houver pelo menos um elemento na matriz para que o predicado retorne verdadeiro e retorna falso se e somente se o

Retornos predicados falsos para todos os elementos da matriz:

Seja `a = [1,2,3,4,5]`;

`A., (x => x%2 === 0) // => true`; A tem alguns números uniformes.

`A. alguma (isnan) // => false`; A não tem não números.

Observe que todos () e alguns () param de iterar os elementos da matriz como

Assim que eles sabem que valor retornar. `alguns ()` retorna verdadeiro o

Primeira vez que seu predicado retorna `<code> true </code>` e apenas itera

Durante toda a matriz, se o seu predicado sempre retornar FALSE.

`todo ()` é o oposto: retorna falsa na primeira vez que você

Retornos predicados falsos e apenas itera todos os elementos se o seu

O predicado sempre retorna verdadeiro. Observe também que, por matemática

Convenção, `todo ()` retorna verdadeiro e `alguns` retornam falsos quando

invocado em uma matriz vazia.

`Reduce ()` e `ReduceRight ()`

Os métodos `Reduce ()` e `ReduceRight ()` combinam o

elementos de uma matriz, usando a função que você especifica, para produzir um

valor único. Esta é uma operação comum na programação funcional