retornar x * x;// Calcule o valor da função
};// semicolon marca o fim do
atribuição.
quadrado (mais1 (y)) // => 16: Invoque duas funções em
uma expressão
No ES6 e mais tarde, há uma sintaxe abreviada para definir funções.
Esta sintaxe concisa usa => para separar a lista de argumentos do
Corpo de função, então as funções definidas dessa maneira são conhecidas como seta
funções.As funções de seta são mais comumente usadas quando você quer
Passe uma função sem nome como argumento para outra função.O
O código anterior se parece com isso quando reescrito para usar funções de seta:
const Plus1 = x => x + 1;// A entrada X mapeia para a saída
x + 1
const square = x => x * x;// A entrada X mapeia para a saída
x * x
Plus1 (y) // => 4: A invocação da função é
o mesmo
quadrado (mais1 (y)) // => 16
Quando usamos funções com objetos, obtemos métodos:
// quando as funções são atribuídas às propriedades de um
objeto, nós chamamos
// eles "métodos".Todos os objetos JavaScript (incluindo matrizes)
tem métodos:
deixe A = [];// Crie uma matriz vazia
a.push (1,2,3);// o método push () adiciona elementos
para uma matriz
a. reverse ();// Outro método: reverter o
ordem dos elementos
// Também podemos definir nossos próprios métodos.A palavra -chave "isto"
refere -se ao objeto
// no qual o método é definido: neste caso, os pontos
Array de antes.
pontos.dist = function () {// define um método para calcular