

Você deve ter notado que a terceira chamada para reduzir () neste exemplo tem apenas um argumento: não há valor inicial especificado. Quando você chama Reduce () como este sem valor inicial, ele usa o primeiro elemento da matriz como o valor inicial. Isso significa que a primeira chamada para a função de redução terá o primeiro e o segundo elementos da matriz como seus primeiros e segundos argumentos. Nos exemplos de soma e produto, nós poderíamos ter omitido o argumento do valor inicial.

Chamando Reduce () em uma matriz vazia sem argumento de valor inicial causa um TypeError. Se você ligar com apenas um valor - uma matriz com um elemento e nenhum valor inicial ou uma matriz vazia e um inicial valor - simplesmente retorna esse valor sem nunca chamar a função de redução.

ReduceRight () funciona como Reduce (), exceto que processa a matriz do índice mais alto para o mais baixo (da direita para a esquerda), mas sim do que do mais baixo ao mais alto. Você pode querer fazer isso se a redução

A operação tem associatividade da direita para a esquerda, por exemplo:

// calcular $2^{(3^4)}$. A exponenciação tem o direito para a esquerda precedência

Seja a = [2, 3, 4];

A. Reduteright ((ACC, Val) => Math.pow (val, acc)) // =>
2.4178516392292583E+24

Observe que nem reduz () nem reduteright () aceita um argumento opcional que especifica esse valor em que o

A função de redução deve ser invocada. O argumento de valor inicial opcional toma seu lugar. Consulte o método function.bind () (§8.7.5) se você

Precisa da sua função de redução invocada como um método de um determinado