

Mesmo que eles tenham as mesmas propriedades e valores. E dois distintos matrizes não são iguais, mesmo que tenham os mesmos elementos no mesmo ordem:

Seja o = {x: 1}, p = {x: 1}; // dois objetos com o mesmo propriedades

o === p // => false: objetos distintos

nunca são iguais

Seja a = [], b = []; // duas matrizes distintas e vazias

a === b // => false: matrizes distintas são

nunca igual

Os objetos às vezes são chamados de tipos de referência para distingui-los de Tipos primitivos de JavaScript. Usando essa terminologia, os valores dos objetos são referências, e dizemos que os objetos são comparados por referência: dois

Os valores dos objetos são os mesmos se e somente se eles se referirem ao mesmo objeto subjacente.

deixe A = []; // A variável A se refere a uma matriz vazia.

Seja b = a; // Agora B refere-se à mesma matriz.

b[0] = 1; // MATATE A matriz referida pela variável b.

a[0] // => 1: A mudança também é visível variável a.

a === b // => true: a e b referem-se ao mesmo objeto,

Então eles são iguais.

Como você pode ver neste código, atribuindo um objeto (ou matriz) a um A variável simplesmente atribui a referência: não cria uma nova cópia de o objeto. Se você quiser fazer uma nova cópia de um objeto ou matriz, você deve copiar explicitamente as propriedades do objeto ou os elementos do variedade. Este exemplo demonstra o uso de um loop for (§5.4.3):

deixe A = ["A", "B", "C"]; // Uma matriz que queremos

cópia

Seja b = []; // Uma matriz distinta nós vamos