

```
copiar em
para (vamos i = 0; i < A.Length; i++) { // para cada índice de um []
  b[i] = a[i]; // copie um elemento de um
  em b
}
```

Seja c = Array.From (B); // em ES6, copiar matrizes
com Array.From ()

Da mesma forma, se queremos comparar dois objetos ou matrizes distintos, nós
deve comparar suas propriedades ou elementos. Este código define uma função

Para comparar duas matrizes:

```
função igualArrays (a, b) {
  if (a === b) retorna true; // idêntico
  Matrizes são iguais
  if (a.Length !== B.Length) retornar FALSO; // Diferente-
  Matrizes de tamanho não são iguais
  para (vamos i = 0; i < A.Length; i++) { // loop
    todos os elementos
    if (a[i] !== b[i]) retorna false; // se houver
    diferem, matrizes não são iguais
  }
  retornar true; // De outra forma
  Eles são iguais
}
```

3.9 Conversões de tipo

O JavaScript é muito flexível sobre os tipos de valores necessários. Nós temos

Vi isso para booleanos: quando JavaScript espera um valor booleano, você
pode fornecer um valor de qualquer tipo, e JavaScript o converterá como
necessário. Alguns valores (valores "verdadeiros") se convertem para verdadeiros e outros
(Valores ?falsamente?) convertem para false. O mesmo vale para outros tipos: se
JavaScript quer uma string, ele converterá qualquer valor que você der em um
corda. Se JavaScript quiser um número, tentará converter o valor que você