

Puede almacenar varias variables del mismo tipo en una estructura de datos de matriz. Puede declarar una matriz mediante la especificación del tipo de sus elementos.

```
type[] arrayName;
```

Los ejemplos siguientes crean matrices unidimensionales, multidimensionales y escalonadas:

```
class TestArraysClass
{
    static void Main()
    {
        // Declare a single-dimensional array.
        int[] array1 = new int[5];

        // Declare and set array element values.
        int[] array2 = new int[] { 1, 3, 5, 7, 9 };

        // Alternative syntax.
        int[] array3 = { 1, 2, 3, 4, 5, 6 };

        // Declare a two dimensional array.
        int[,] multiDimensionalArray1 = new int[2, 3];

        // Declare and set array element values.
        int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };

        // Declare a jagged array.
        int[][] jaggedArray = new int[6][];

        // Set the values of the first array in the jagged array structure.
        jaggedArray[0] = new int[4] { 1, 2, 3, 4 };
    }
}
```

Información general de las matrices

Una matriz tiene las propiedades siguientes:

- Puede ser una matriz unidimensional, multidimensional o escalonada.
- El número de dimensiones y la longitud de cada dimensión se establecen al crear la instancia de matriz. No se pueden cambiar estos valores durante la vigencia de la instancia.
- Los valores predeterminados de los elementos numéricos de matriz se establecen en cero y los elementos de referencia se establecen en null.

- Una matriz escalonada es una matriz de matrices y, por consiguiente, sus elementos son tipos de referencia y se inicializan en null.
- Las matrices se indexan con cero: una matriz con n elementos se indexa de 0 a n-1.
- Los elementos de una matriz puede ser cualquier tipo, incluido un tipo de matriz.
- Los tipos de matriz son tipos de referencia que proceden del tipo base abstracto Array. Como este tipo implementa IEnumerable y IEnumerable<T>, puede usar la iteración foreach en todas las matrices de C#.