

Cet article a fait l'objet d'une traduction manuelle. Déplacez votre pointeur sur les phrases de l'article pour voir la version originale de ce texte. [Informations supplémentaires.](#)

Traduction  Source

# Tableaux (guide de programmation C#)

## Visual Studio 2015

Vous pouvez stocker plusieurs variables du même type dans une structure de données de tableau. Vous déclarez un tableau en spécifiant le type de ses éléments.

```
type[] arrayName;
```

Les exemples suivants créent des tableaux unidimensionnels, multidimensionnels et en escalier :

**C#**

```
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 };
    }
}
```

## Vue d'ensemble des tableaux

Un tableau possède les propriétés suivantes :

- Un tableau peut être [Unidimensionnel](#), [Multidimensionnel](#) ou [En escalier](#).
- Le nombre de dimensions et la longueur de chaque dimension sont des valeurs établies lorsque l'instance de tableau est créée. Ces valeurs ne peuvent pas être modifiées pendant la durée de vie de l'instance.
- Les valeurs par défaut des éléments de tableau numériques sont égales à zéro et les éléments de référence ont la valeur Null.
- Un tableau en escalier est un tableau de tableaux, et par conséquent ses éléments sont des types référence et sont initialisés à **null**.
- Les tableaux sont indexés à partir de zéro : un tableau avec *n* éléments est indexé de **0** à **n-1**.
- Les éléments du tableau peuvent être de n'importe quel autre type, y compris un type tableau.
- Les types de tableaux sont des [types référence](#) dérivés du type de base abstrait [Array](#). Puisque ce type implémente [IEnumerable](#) et [IEnumerable<T>](#), vous pouvez utiliser l'itération [foreach](#) sur tous les tableaux en C#.

## Rubriques connexes

- [Tableaux en tant qu'objets \(guide de programmation C#\)](#)
- [Utilisation de foreach avec des tableaux \(Guide de programmation C#\)](#)
- [Passage de tableaux en tant qu'arguments \(Guide de programmation C#\)](#)
- [Passage de tableaux à l'aide de paramètres ref et out \(guide de programmation C#\)](#)
- [En savoir plus sur les variables](#) dans [Débuter avec Visual C# 2010](#) (en anglais)

## Spécification du langage C#

Pour plus d'informations, voir la [Spécification du langage C#](#). La spécification du langage est la source de référence pour la syntaxe C# et son utilisation.

## Voir aussi

[Guide de programmation C#](#)  
[Collections \(C# et Visual Basic\)](#)  
[Array Collection Type](#)

© 2016 Microsoft