

# apprendre le c#

- Objet : apprendre le c#
- Niveau requis :  
débutant, avisé
- Commentaires : *Contexte d'utilisation du sujet du tuto.*

## GLOSSAIRE

<http://www.dotnet-france.com/Documents/IntroCSharpVBNET/Notions%20avanc%C3%A9es%20du%20langage%20CSharp.pdf>

<https://www.dotnetdojo.com/tableau-csharp/>

## Matrices

### Quelques rappels

### Schémas de matrices

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Structuré
{
    class RemplirEtAfficherMatrice
    {
        public const int L = 10;
        public const int C = 15;

        static void Main(string[] args) // début du programme principal
        {
            do // Début du "do ...while" proposant à l'utilisateur de
recommencer le programme
            {
                int[,] Ma_matrice;
                // matrice = new int [ L , C ]; // Allocation d'espace
en mémoire

                //(pas encore l'affectation des cases :
//Mais c'est la fonction qui le fait et qui remplit les
cases

                Ma_matrice = SaisirMatriceEntiers(); // appel de la
```

```

fonction qui remplit la matrice "Ma_matrice"
        AfficherMatriceEntiers(Ma_matrice); // appel de la
fonction qui permet d'afficher "Ma_matrice"
        Console.WriteLine();
    } while(veutContinuer("Voulez-vous remplir et afficher une autre
matrice ? (o/n)"));
    Console.WriteLine("Tapez sur Entree pour Terminer");
    Console.ReadLine();          // fin du programme principal
}

/*****Remarques*****/
* toutes les fonctions ci-dessous pourraient être placées dans un
autre fichier
* par exemple, une class Mes fonctions
* dans ce cas, il faudrait placer le nom du main suivi d'un point
devant le nom de la fonction lors de son appel
* par exemple class MesFonctions
* appel Ma_matrice = MesFonctions.SaisirMatriceEntiers();
* C'est ce qui est fait pour l'appel de la fonction
"Lire.Caractere" utilisée dans les fonctions ci-dessous.*****/

// Fonction qui permet de recommencer le programme
private static bool veutContinuer(string invite)
{
    char cara;
    do
    {
        cara = Char.ToLower(Lire.Caractere(invite));
        if (((cara != 'n') && (cara != 'o')))
            //if (!( (cara == 'n') || (cara == 'o') )) // expression
            identique selon règle de De Morgan
        {
            Console.WriteLine("Répondez oui ou non (o/n)");
        }
    } while (!((cara == 'o') || (cara == 'n')));
    return ((cara == 'o'));
}

//fonction qui permet de remplir une matrice
private static int[,] SaisirMatriceEntiers()
{
    int nbLig, nbCol; // dimensions utiles
    int[,] matrice;

    // Saisie de la taille utile de la matrice
    do // Nombre de lignes
    {
        nbLig = Lire.Entier("Indiquez le nombre de lignes de la
matrice (entre 1 et " + L + ") ");
        // Arrêt quand la valeur saisie est acceptable (càd entre 1

```

```

et MAX_NB_LIG incluses).
    } while ((nbLig < 1) || (L < nbLig));

    do // Nombre de colonnes
    {
        nbCol = Lire.Entier("Indiquez le nombre de colonnes de la
matrice (entre 1 et " + C + ") ");
        // Arrêt quand la valeur saisie est acceptable (càd entre 1
et MAX_NB_COL incluses).
    } while ((nbCol < 1) || (C < nbCol));

    // Allocation d'espace pour la matrice à saisir :
matrice = new int[nbLig, nbCol];

    // Saisie de la matrice
for (int i = 0; i < nbLig; i++) // Pour chaque ligne
{
    for (int j = 0; j < nbCol; j++) // Pour chaque colonne
dans la ligne
        // Arrêt avant de dépasser la taille max d'une ligne (càd le
nombre de colonnes).
        {
            matrice[i, j] = Lire.Entier("Matrice [" + i + ", " + j +
"] =");
        }
    }
return matrice;
}
// Fonction qui permet d'afficher une matrice :
private static void AfficherMatriceEntiers(int[,] matrice)
{
    int nbLig = matrice.GetLength(0); // nb lignes de la matrice
:
    // "GetLength(0)" longueur de la première dimension, la première
c'est zéro
    int nbCol = matrice.GetLength(1); // nb colonnes :
    // "GetLength(1)" : GetLength(1) longueur de la deuxième
dimension, la deuxième c'est 1

    for (int i = 0; i < nbLig; i++) // Pour chaque ligne
    {
        Console.WriteLine(); //saut de ligne avant chaque ligne
de la matrice
        for (int j = 0; j < nbCol; j++) // Pour chaque colonne
dans la ligne
        {
            Console.Write(matrice[i, j] + " | ");
        }
    }
    Console.WriteLine(); //saut de ligne après la matrice
}

```

```
}  
}
```

## Utilisation

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/utilisateurs:hyathie:config:apprendre-le-c>



Last update: **21/12/2014 06:42**