

Fiche de Procédure : Les Bases du C#

1. Qu'est-ce que C# ?

C# est un langage de programmation orienté objet développé par Microsoft. Il est principalement utilisé pour créer des applications Windows, des sites web (avec ASP.NET), des jeux (avec Unity), et bien plus encore.

2. Structure de base d'un programme C#

Un programme C# commence généralement par la déclaration d'un espace de noms (*namespace*), suivi d'une classe, puis d'une méthode `Main` qui est le point d'entrée du programme.

```
using System; // Importation de l'espace de noms System

namespace MonPremierProgramme
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Bonjour, C#!");
        }
    }
}
```

3. Les variables et types de base

Type	Description	Exemple
int	Nombre entier	int age = 30;
double	Nombre décimal	double prix = 9.99;
string	Chaîne de caractères	string nom = "Marie";
bool	Booléen (vrai/faux)	bool actif = true;

4. Les structures de contrôle

Condition (if, else, else if)

```
int age = 20;
if (age >= 18)
{
    Console.WriteLine("Majeur");
}
else
{
    Console.WriteLine("Mineur");
}
```

Boucle (for)

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine("Tour numéro " + i);
}
```

Boucle (while)

```
int compteur = 0;
while (compteur < 3)
{
    Console.WriteLine("Compteur : " + compteur);
    compteur++;
}
```

5. Les fonctions (méthodes)

```
static void DireBonjour(string prenom)
{
    Console.WriteLine("Bonjour " + prenom + " !");
}
```

```
// Appel de la méthode
DireBonjour("Lucie");
```

6. Les classes et objets

```
class Personne
{
    public string Nom;
    public int Age;

    public void SePresenter()
    {
        Console.WriteLine("Je m'appelle " + Nom + " et j'ai " + Age
+ " ans.");
    }
}
```

```
// Utilisation
Personne p = new Personne();
p.Nom = "Marie";
p.Age = 25;
p.SePresenter();
```

7. Lire et écrire dans la console

Afficher un message

```
Console.WriteLine("Ceci est un message.");
```

Lire une saisie utilisateur

```
Console.WriteLine("Quel est votre prénom ?");  
string prenom = Console.ReadLine();  
Console.WriteLine("Bonjour " + prenom + " !");
```

8. Exemple complet

```
using System;  
  
namespace ExempleCSharp  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Entrez votre nom :");  
            string nom = Console.ReadLine();  
  
            Console.WriteLine("Entrez votre âge :");  
            int age = int.Parse(Console.ReadLine());  
  
            Console.WriteLine("Bonjour " + nom + ", vous avez " +  
age + " ans.");  
        }  
    }  
}
```

9. Bonnes pratiques

- Toujours nommer les classes et méthodes de façon explicite.
- Indenter et commenter le code pour la lisibilité.
- Utiliser des types adaptés aux données.
- Tester et valider les entrées utilisateur (ex : avec `int.TryParse`).
- Organiser le code en plusieurs méthodes et classes si besoin.

10. Pour aller plus loin

- Découvrir les collections (`List`, `Dictionary`, etc.).
- Explorer la programmation orientée objet avancée (héritage, interfaces).
- Manipuler des fichiers, des bases de données.
- Créer des interfaces graphiques (WPF, WinForms) ou des applications web (ASP.NET).
- Développer des jeux avec Unity.

Astuce :

Pour tester rapidement du C#, utilisez un IDE comme Visual Studio, Visual Studio Code (avec l'extension C#), JetBrains Rider ou un éditeur en ligne comme dotnetfiddle.net !