

Ce qu'il faut retenir

Le langage C# est exécuté de haut en bas. Chaque instruction se termine obligatoirement **par un point virgule**.

Concernant Visual Studio

A l'intérieur de Visual Studio, vous allez retrouver plusieurs fenêtres qui vont composer la vue principale de votre logiciel. La vue principale correspond au fichier sélectionné (en cours de consultation) et vous pouvez y voir le code qu'il contient. Si vous utilisez Visual Studio Code, les fenêtres sont disposées d'une autre manière mais il s'agit sensiblement de la même chose.

Par défaut, vous avez sur la droite une fenêtre qu'on appelle **l'explorateur de solution**. C'est dans cette fenêtre que vous allez retrouver **l'ensemble des fichiers qui vont constituer vos programmes (ou projets)** et l'ensemble de vos programmes (ou projets) qui vont constituer un grand ensemble qu'on appelle **une solution**. D'où le nom **"Explorateur de solution"**.

Concernant une solution

"Quelle est la différence entre un programme et une solution ?" , "Je ne comprends pas pourquoi je vais avoir plusieurs programmes dans ma solution ?".

Ce sont des questions qu'il est tout à fait légitime de se poser. Vous imaginez peut être qu'un "programme" c'est le logiciel final que vous êtes en train de programmer. En soit, vous avez raison. Sauf que lorsque nous souhaitons réaliser une application informatique de qualité professionnelle, nous avons plutôt tendance à décomposer notre programme en plusieurs sous parties avec des fonctions spécifiques. Par exemple, une partie sera capable d'afficher les informations à l'utilisateur, une autre partie se chargera de récupérer les données dans une base de donnée, etc. Notre programme est en réalité composé de sous programme qu'on peut également appeler projet. Et c'est cet ensemble de projets que l'on va retrouver à l'intérieur de la "Solution".

Si vous n'avez pas compris, ce n'est pas bien grave, vous allez comprendre par vous même au fur et à mesure de votre apprentissage.

Concernant les quelques instructions à retenir.

Voici un récapitulatif des instructions que nous avons vu.

- `Console.WriteLine()` permet d'afficher une phrase (qui va être comprise entre les parenthèses) dans la console et d'effectuer un retour à la ligne.
- `Console.Write()` fait la même chose que l'instruction du dessus à ceci près qu'elle n'effectue pas de retours à la ligne dans la console.
- `Console.ReadKey()` est une instruction qui attend une saisie de l'utilisateur. Une fois que l'utilisateur aura appuyé sur n'importe quelle touche du clavier (mais une seule) le programme continuera avec l'instruction suivante. Il est évidemment possible de récupérer la touche du clavier saisie par l'utilisateur puisque la fonction `Console.ReadKey` retourne cette information. Nous verrons tout cela un peu plus loin.
- `Console.ReadLine()` est une instruction qui attend une saisie de l'utilisateur. Mais cette fois-ci, l'utilisateur peut saisir autant de texte qu'il le souhaite. Il suffira d'appuyer sur la touche Entrée pour que la saisie soit considérée comme finie et que le code continue avec l'instruction suivante. Evidemment il est possible récupérer la saisie de l'utilisateur comme nous le verrons plus loin.

Les bouts de code sont inclus dans des "dossiers" qu'on appelle des **namespaces** et il est possible d'utiliser ces **namespaces** avec l'instruction **using** que l'on met en haut du fichier .cs. Si les morceaux de code que l'on souhaite utiliser se situe dans le même namespace (même s'ils sont dans des fichiers différents), il n'est pas nécessaire d'utiliser un using.

Pour finir, la fonction **Main** est le point d'entrée (par défaut) d'une application console en c#. Vous ne savez pas encore ce que c'est qu'une fonction (tout du moins si vous êtes débutant en programmation) mais ce n'est pas très important dans la mesure où nous le verrons plus tard. Gardez simplement à l'esprit que l'instruction **Main** que vous voyez, c'est le nom de la fonction et c'est par là que **le programme va démarrer**. Cette fonction main prend **en "paramètres"**

(dans les parenthèses qui suivent) le nom de la fonction des arguments obligatoires à savoir **string[] args**. Il s'agit, comme nous le verrons par la suite d'un tableau de string qui s'appelle args.

```
public static void Main (string[] args) {  
    Console.WriteLine ("Hello World");  
}
```