

# Les Bases du langage C#

## A. Les variables , constantes et les reste :

### Qu'est ce que les « variables » ?

Les variables vont vous permettre de mémoriser , pendant que votre application tourne différentes valeurs utiles pour le fonctionnement de votre programme, une variable doit obligatoirement être déclaré avant d'être utilisé mais ce n'est pas tout il faut qu'elle soit initialisée ne l'oubliez jamais.

### Le nom d'une variable ?

Bah bien évidemment une variable à un nom pour se différencier des autres , mais il y à quelques règles à respecter !

Listons ces règles :

- Le nom de la variable doit commencer par une lettre
- Il peut être constitué de lettres de chiffres ou encore du caractère « \_ »
- Il peut contenir 1000 caractère a peut près (Faut être malade pour avoir une variable de cette longueur)
- La Case est importante
- Les mots clé du langage ne doivent pas être utilisé (Exemple : if while ou autres ...) Sauf si on ajoute @ devant le nom de la variable.

### Type des variables ?

Oui une type permet de définir ce que contiendra comme information notre variable qui possède déjà un nom.

Il y à deux type de variables qui sont disponibles:

- Les types valeur : la variable contient réellement l'information.
- Les types référence : la variable contient l'adresse en mémoire ou sont situer les informations.

Les différents types de variables sont définis au niveau du Framework oui le Framework on en parle dans le pack que je vous ai fournis avec la présentation des nouvelles technologies de Microsoft.

On peut par exemple utiliser les alias définis au niveau de visual C# , le type System.Int32 peut remplacer int.

Maintenant qu'on sait ce qu'est un « type » combien il y en a de types pour définir nos variables ?

Il y en a en tous 6 nous allons les lister ci-dessous :

Les types numériques entiers :

- sbyte
- short
- int
- long
- byte
- ushort
- uint
- ulong

Les types décimaux :

- float
- double
- decimal

Les types caractères :

Le type char est utilisé pour stocker un caractère unique, une variable de type char utilise deux octets pour stocker l'Unicode du caractère . Les 128 premiers caractères sont identiques au jeu de caractère ASCII, les caractères suivant jusqu'à 255 correspondent au caractères spéciaux et le reste c'est les symboles ( ouarf j'ai lutter pour apprendre ça).

L'affectation d'une valeur à une variable de type char doit être effectuée en encadrant la valeur par « caractère »

Il y à aussi les séquences d'échappements je pense que vous en avez déjà entendu parler

Je vais vous en donner quelques exemples et pas tous vous les lister car j'en ai pour un moment et j'ai la flemme de faire un tableau.

- \n correspond à Saut de Ligne
- \f correspond à Saut de Page
- \r correspond à Retour Chariot
- et bien d'autres...

Si vous en voulez la liste complète je vous conseil d'aller faire un tour sur le net, j'ai pas réussi à trouver un site les réunissant tous.

On a vu ce qu'était un caractère et que son type était le « char » alors quel est le type d'une chaine de caractères hum et bien c'est un string, oui un string c'est la traduction en anglais de chaine de caractères.

Les chaînes de caractères sont invariables car lors de l'affectation d'une valeur l'espace est réservé dans la mémoire et si on affecte plus tard une nouvelle valeur à cette chaîne de caractères la taille augmentera automatiquement c'est pratique hein ?

Plus besoin de s'ennuyer comme en C avec ces tableaux de caractères sur lequel on devait définir une taille pour qu'on puisse la stocker dans la mémoire.

Par exemple on va déclarer une chaîne de caractères :

- `string NomDuDéveloppeur = « God »;`

Que de révolutions avec le C# mais vous n'avez pas fini d'être étonné.

Le type `bool` :

Le type booléen permet d'utiliser une variable qui peut prendre deux états différents comme « oui / non » « vrai / faux » ou encore « on / off » rien de bien compliqué n'est ce pas ?

Exemple :

- `bool Disponible = false;`

Si on affecte une variable numérique à un booléen toutes valeurs différentes de 0 sera considéré comme un booléen `true`.

Le type `Object` :

C'est le type le plus utilisé je pense dans Visual C#, dans une variable de type `Object` vous pouvez mettre ce que vous voulez, En réalité ce type de variable ne stock rien.

La variable va pas contenir une valeur mais l'adresse où on pourra trouver la valeur de la variable

Vous vous rappelez l'utilisation des références eh bien voilà un cas d'application.

Il reste le type `Nullable` mais celui-ci est peut-être utilisé quoique, mais je vais pas non plus tout vous donner donc je vais vous laisser chercher ce qu'est le type `Nullable` en C# à quoi sert-il et comment l'utiliser.

Hé bien vous avez pris connaissance des différents types de variables, mais imaginons que un moment dans votre script vous vous êtes trompés de type et que vous voulez le changer.

Je vous voit venir avec vos grosses têtes « On a qu'à réécrire le type de la variable ?!! », hé bien non moi je veux pas, je vais vous apprendre une autre technique qui s'appelle le « `Cast` » ou « `Transtypage` »

Allez on va éclaircir tout ça, c'est parti sortez l'aspirine !

