Compétence 6 : Organiser son développement professionnel

Sous-Compétence : Mettre en place son environnement d'apprentissage personnel

Langage

- Types de donnée
- Variables
- Macros
- Opérateurs
- Instructions conditionnelles
- Instructions répétitives
- Instructions pour objets
- Fonctions utilisateur
- Commentaires
- Directives

Variables - Référence du Langage

Une variable est juste un emplacement mémoire où vous stockez une donnée de telle sorte qu'elle soit accessible rapidement. Pensez à une boîte aux lettres dans la mémoire où vous pouvez placer ou retirer une information. Par exemple, vous pourriez créer une variable pour stocker le nombre de réponses d'un utilisateur, ou le résultat d'une équation mathématique.

Chaque variable a un nom (comme une boite aux lettres), ce nom doit commencer par le caractère **$** et ne doit contenir que des **lettres**, des **chiffres** et le caractère de soulignement '_' . Voici quelques exemples de noms :

$Var1

$vVariable

$ma_Variable

Notez que tous les noms de variables sont insensibles à la casse: $maVariable est la même que $MavARiAblE.

Chaque variable est stockée comme une information 'variant' .

Déclaration de Variables

Les variables sont déclarées et créées avec les mots clé Local et Global - Dim peut aussi être utilisé, mais ce n'est pas recommandé.

```
Local $vVariable
```

Ou bien, vous pouvez déclarer plusieurs variables simultanément:

```
Global $vVariable1, $vVariable2
```

Vous pouvez aussi assigner une variable **sans** la déclarer auparavant, mais beaucoup préfèrent les déclarations explicites.

```
$vVariable = "Create and Assign"
```

Déclaration des Constantes

Les constantes sont déclarées et créées en utilisant le mot-clé Const et doivent être initialisées avec une valeur:

```
Const $iConst1 = 1, $iConst2 = 12
```

Fonctions Utilisateur - Référence du Langage

Une fonction est une partie de code qui peut être appelée dans le script pour effectuer une certaine "function". Il existe deux sortes de fonctions dans AutoIt, **Fonctions intégrées** et **Fonctions utilisateur**.

Notez que les noms de toutes les fonctions sont insensibles à la casse: msgbox() est identique à MsgBox() et MyFunc() est identique à MyfunC().

Fonctions intégrées

La liste complète des fonctions intégrées est disponible ici et l'explication de leur utilisation ici.

Fonctions Utilisateur (UDF)

Les fonctions Utilisateur sont déclarées en utilisant l'instruction Func...EndFunc.

Les fonctions peuvent accepter des paramètres et fournir des valeurs de retour.

Les noms de fonctions doivent commencer par une lettre ou le caractère de soulignement (underscore) et le reste du nom peut contenir n'importe quelle combinaison de lettres, de chiffres ou du caractère de soulignement. Exemple de fonctions valides :

MaFonct

Fonct1

_Ma_Fonct1

Voici un exemple d'utilisation d'une fonction qui double un nombre et qui est appelée 10 fois:

```
#include <Constants.au3>

Local $iNumber = 10
Local $iDoubled = 0

For $i = 1 To 10
    $iDoubled = MyDouble($iNumber)
    MsgBox($MB_OK, "", $iNumber & " doubled is " & $iDoubled)
    $iNumber = $iDoubled
Next
Exit

Func MyDouble($iValue)
    $iValue = $iValue * 2
    Return $iValue
EndFunc   ;==>MyDouble
```