Programme : suite d'instructions manipulant des données

LANGAGE VISUAL BASIC

Données typées. Visual Basic propose les types usuels de la programmation : entier, réels, booléens, chaîne de caractères.

Structures avancées de données. Gestion des collections de valeurs (énumérations, tableaux) et des objets structurés (enregistrements, classes).

Séquences d'instructions, c'est la base même de la programmation, pouvoir écrire et exécuter une série de commandes sans avoir à intervenir entre les instructions.

Structures algorithmiques : les branchements conditionnels et les boucles.

Les outils de la programmation structurée : pouvoir regrouper du code dans des procédures et des fonctions. Organisation du code en modules et possibilité de distribuer ces dernières.

Visual Basic n'est pas « case sensitive », il ne différencie pas les termes écrits en minuscule et majuscule.

Le type de données définit le type d'opérateurs qu'on peut leur appliquer.

- **Numérique** qui peut être réel (double) ou entier (long). Les opérateurs applicables sont : +, -, *, / (division réelle), \ (division entière), mod (modulo)

Exemple : 5 / 2 → 2.5 ; 5 \ 2 → 2 ; 5 mod 2 → 1

- **Booléen** (boolean) qui ne prend que deux valeurs possibles : True et False. Les opérateurs sont : not, and, or.

Exemple : True and False → False

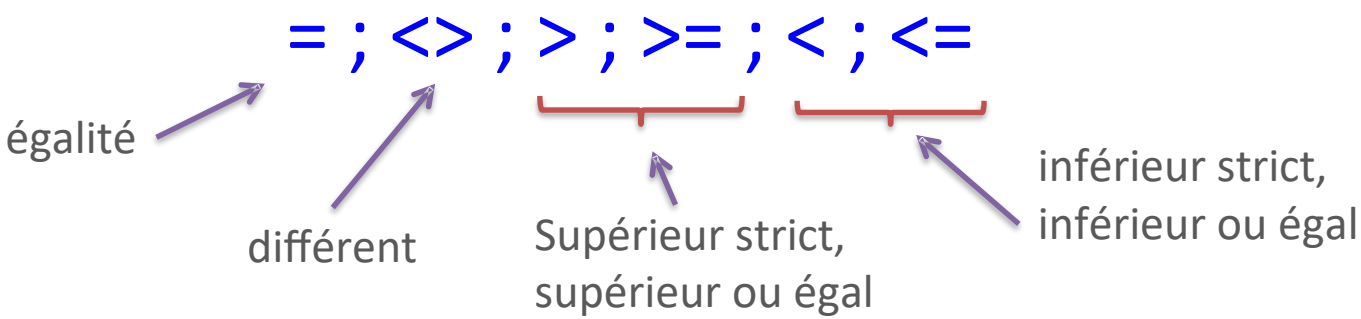
- **Chaîne de caractères** (string) qui correspond à une suite de caractères délimitée par des guillemets '' ''. Les opérateurs possibles sont la concaténation, la suppression d'une sous-partie, la copie d'une sous-partie, etc.

Exemple : ''toto'' est une chaîne de caractères, toto on ne sait pas ce que c'est (pour l'instant)

Habituellement, les opérations font intervenir des données de type identique et renvoie un résultat du même type.

• Type	• Valeurs
• Boolean	• Vrai, faux
• Integer	• Entiers
• Long	• Entiers
• Single	• Réels
• Double	• Réels
• Currency	• 4 chiffres après la ,
• Date	• 1/1/100 à 31/12/9999
• String	• Chaines de caractères
• Object	• Tout objet
• Variant	• N'importe quel type

Les opérateurs de comparaison confrontent des données de même type, mais le résultat est un booléen



Exemples

5 > 2 → True

5 > ''toto'' → illicite

5 <> 5 → False

''toto'' > ''tata'' → True

Licite. Comparaison de gauche à droite basée sur le code ASCII. Arrêt des comparaisons dès que l'indécision est levée.

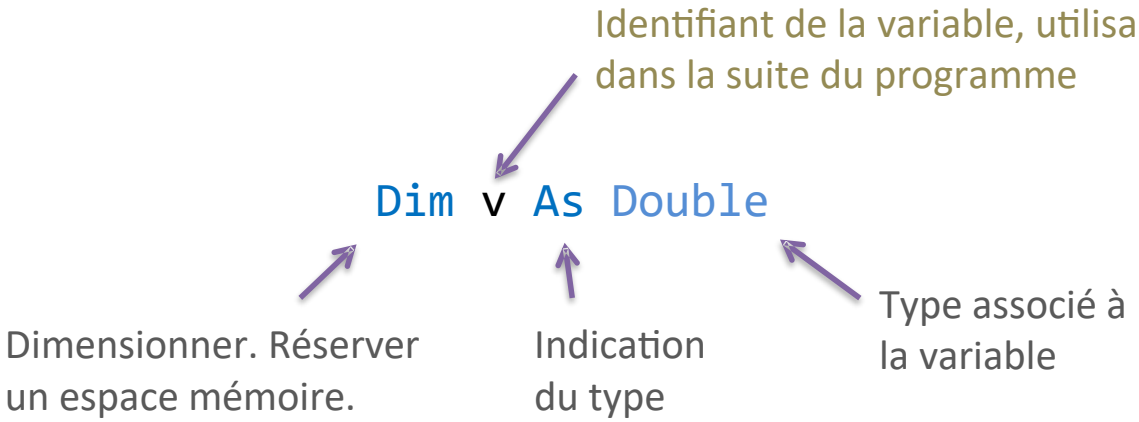
- Valeur absolue: Abs(-9) retourne 9
- Signe: Sgn(-18) retourne –1 (ou 0 ou 1)
- Troncature à l'unité : Fix(-18.3) = -18
  Fix(18.3) = 18
  - Tronque la partie décimale
- Partie entière: Int(13.12) retourne 13
  Int(-14.8) retourne –15
  - $E(x) \leq x < E(x) + 1$
  - Tronque à l'entier inférieur le plus proche.

- Sqr, Exp, Log
  - Sqr(4) retourne 2, Exp(5) retourne 148.413…, Log(9) retourne 2.197224… (en base e)
- Nombres aléatoires
  - Rnd retourne un nombre aléatoire entre 0 (compris) et 1 (non compris)
  - a = Rnd a peut valoir 0.12131441
  - Int((b – a + 1) * Rnd + a) retourne un nombre aléatoire entier entre a et b
- Sin, Cos, Tan, Atn (arc-tangente)

- Date retourne la date actuelle
- Time retourne l'heure courante
  - Date et Time peuvent retourner des chaînes de carctères String
- DateSerial retourne une valeur unique pour une date donnée, sous forme Variant
  - dv1 = DateSerial(2003, 4, 22)
    dv2 = DateSerial(1928, 5, 3)
    dv1 – dv2 représente le nombre de jours entre ces deux dates
- Day, Month et Year retourne respectivement le jour, le mois et l'année d'une date.
  - Year(Date) retourne 2019 cette année (en entier)

Les **variables** correspondent à des identifiants auxquels sont associés des valeurs d'un type donné. Elles matérialisent un espace mémoire avec un contenu que l'on peut lire ou écrire.

Déclaration d'une variable



Affectation. Attribuer une valeur à la variable

```
v = 2.5
```

= est le symbole d'affectation. A gauche de = on modifie le contenu dans une variable, à droite on lit le contenu d'une variable. C'est pour cette raison que l'instruction v = v + 1 est licite.

Opération et affectation

```
x = v * 2
```

La valeur 5 est écrite dans la variable x qui doit être déclarée au préalable.