

## 1. Les variables

Les variables sont désignent un emplacement dans la mémoire d'un ordinateur. Elles sont caractérisées par leur type et leur identificateur.

### a. Les types de base

Le type de base caractérise la taille mémoire d'une variable et l'ensemble des valeurs qu'elle peut prendre.

Booléen :	type logique de valeur <b>VRAI</b> ou <b>FAUX</b>
Octet :	entier non signé sur 8 bits (0 .. 255)
Caractère :	entier non signé sur 8 bits, codage des symboles ASCII
Entier :	entier signé sur 16 bits (-32 768 .. +32 767)
Entier court :	entier signé sur 8 bits (-128 .. +127)
Entier long :	entier signé sur 32 bits (-2 147 483 648 .. +2 147 483 647)
Mot :	entier non signé sur 16 bits (0 .. 65 535)
Mot long :	entier non signé sur 32 bits (0 .. 4 294 967 295)
Réel :	valeur réelle simple précision (codage IEEE sur 32 bits)
Réel double :	valeur réelle double précision (codage IEEE sur 64 bits)

### b. L'identificateur

Permet de donner un nom à une variable. Ce nom ne doit pas commencer par un chiffre et les seuls caractères autorisés sont les lettres, les chiffres et le caractère de soulignement (\_).

### c. Les opérateurs

Opérateurs arithmétiques	
+	addition
-	soustraction
*	multiplication
/	division
%	modulo
Opérateurs de comparaison	
==	égalité
<	infériorité stricte
<=	infériorité
>	supériorité stricte
>=	supériorité
!=	différence
Opérateurs logiques	
	OU logique
&&	ET logique
!	négation
Opérateur d'accès	
=	affectation

Principaux opérateurs (les symboles peuvent varier d'un langage ou pseudo-langage à l'autre)

Tous les langages de programmation sont capables d'effectuer des opérations arithmétiques ou logiques et des comparaisons.

L'opérateur d'affectation possède deux opérandes : une à droite et une à gauche. Son usage signifie que l'on stocke le résultat de l'expression de droite à l'emplacement mémoire spécifié par la variable de gauche.

- Il faut bien faire attention à ne pas confondre une affectation et une égalité. Exemple dans le cas du Robotino :

```
if (bumper.value() == true) {
    setVelocity(0, 0, 0);
}
```

Ici il s'agit d'une égalité : on cherche à déterminer si bumper.value() est égal ou pas à la valeur true.

Ici il s'agit d'une affectation : on augmente de 50 la valeur de la variable **speed**. **speed** est un identificateur et **speed** est de type réel.

```
public void actionPerformed(ActionEvent arg0) {
    speed = speed + 50;
    setVelocity(speed, 0, 0);
}
```

## 2. Les conditions

Les conditions permettent de prendre des décisions lors de l'exécution d'un programme. Une expression est construite autour d'un ou plusieurs opérateurs. Les conditions sont des expressions qui ne peuvent prendre que deux valeurs à savoir vrai ou faux. Les conditions interviennent dans toutes les structures fondamentales.

speed = 50	■■■■■
Expression	Valeur
speed == 50	vraie
speed != 51	vraie
speed <= 40	faux
speed == 2*15 + 20	vraie

Valeur d'une expression

