

Lecture et Ecriture dans un algorithme

En programmation, « lire » une valeur, ou « écrire » une valeur, est toujours utilisé dans le sens de la machine (ordinateur ou calculatrice).

Lorsque la machine va afficher une phrase, on utilisera donc la commande « écrire ». Dans cette situation, l'utilisateur est amené à lire la phrase affichée à l'écran.

Lorsque la machine va récupérer une information donnée par l'utilisateur, on utilisera donc la commande « lire ». Dans cette situation, l'utilisateur est amené à écrire une information et à la valider en appuyant sur Entrée.

Exemple :

```
Début.  
| Ecrire (« Comment tu t'appelles ? »).  
| Lire (prenom).  
| Ecrire (« Bonjour », prenom).  
Fin.
```

L'écran affiche « comment tu t'appelles ? »
L'utilisateur écrit son prénom, mémorisé dans la variable prenom.
L'écran affiche bonjour suivi du prénom de l'utilisateur.

On peut choisir d'incrémenter légèrement les instructions, ce qui permet d'identifier plus rapidement la longueur du programme.

Pour permettre de comprendre mieux un algorithme, ou un programme, on peut décrire en quoi consiste chaque instruction à côté, en commentaire. La police utilisée doit clairement permettre de comprendre que ce sont des commentaires, et non des instructions.

Remarques :

- Avez-vous observé que l'instruction en troisième ligne du programme contient un espace juste après le Bonjour avant de fermer les guillemets ? A votre avis, pourquoi ?
- Lorsque l'utilisateur écrit son prénom et appuie sur entrée, pour mémoriser le prénom, le programme doit nécessairement créer une « boîte » dans laquelle il va mémoriser la chaîne de caractères rentrée par l'utilisateur, afin de pouvoir ensuite utiliser à nouveau sa valeur.

Cet exemple nous permet de comprendre que nous avons besoin de mémoriser des informations à l'intérieur de notre programme. D'où la notion de variable.

Utiliser des variables dans un algorithme

Bien souvent, pour faire un algorithme qui puisse interagir avec l'utilisateur, nous aurons besoin de lire des informations données par l'utilisateur, et de stocker ces informations (comme le prénom dans l'exemple précédent).

Parfois, on devra faire des calculs internes avec ces valeurs. Ces calculs seront totalement invisibles par l'utilisateur : seul l'analyste programmeur a connaissance de comment fonctionne le programme.

Les valeurs que l'on doit stocker sont rangées dans des « variables » (un espace mémoire, une « boîte »).

Exemple : dans l'algorithme du prénom de la page précédente, on « range » le prénom de la personne dans une variable nommée *prenom*, ainsi, on peut utiliser cette valeur un peu plus tard, en disant bonjour à l'utilisateur.

Le nom de la variable peut n'avoir qu'une seule lettre.

Le nom de la variable doit permettre de retrouver facilement la valeur qui est dans la variable.

Selon le langage de programmation, les variables doivent être déclarées, ainsi que leur nature (est-ce un nombre entier ? un nombre réel ? une chaîne de caractères ? un unique caractère ? un booléen ? etc...)

Exemple :

Variables :

a, b, c, d, e sont des nombres. } Variables de type nombre

Début :

Lire a }
Lire b } Initialisation
 $c \leftarrow a + b$ }
 $d \leftarrow a - b$ }
 $e \leftarrow c \times d + b^2$ } Traitement
Afficher e } Sortie

Fin.

Pour comprendre ce qu'exécute ce programme, on peut utiliser un tableau de variables :
Exécutons ce programme pour $a = 5$ et $b = 3$.

	a	b	c	d	e
Initialisation	5	3			
Traitement			8		
				2	
					25
Sortie					25

Tester ce programme avec les valeurs 7 et 1, puis avec les valeurs (-2) et 4.
Que fait, en réalité, ce programme ?