Algorithmique et Programmation
Chap 2 : Variable et Affectation

E. M. Souidi

Faculté des Sciences - Rabat
SVI4 –STU4 2013-14

Lors du traitement d'informations par un programme (ou algorithme), les données et résultats sont des grandeurs susceptibles de varier.

Lors du traitement d'informations par un programme (ou algorithme), les données et résultats sont des grandeurs susceptibles de varier.
On les appelle pour cette raison des variables.

Lors du traitement d'informations par un programme (ou algorithme), les données et résultats sont des grandeurs susceptibles de varier.
On les appelle pour cette raison des variables.

On leur donne des noms pour les identifier.

On leur donne des noms pour les identifier.
Les variables permettent de stocker provisoirement des valeurs. On trouve aussi dans les algorithmes des données constantes. Par exemple Pi=3,14, e=2,73. On les appelle constantes.

Les variables et les constantes peuvent provenir de l'utilisateur (entré au clavier), d'un périphérique, des valeurs intermédiaire ou définitives obtenues par le programme.

identificateur

Une variable ou une constante doit toujours posséder un nom appelé identificateur, qui doit être choisi aussi explicite que possible.

identificateur

Une variable ou une constante doit toujours posséder un nom appelé identificateur, qui doit être choisi aussi explicite que possible.
Aucune raison d'appeler 3,1415 x ou y mais pi ou Pi.

La règle consiste à construire les identificateurs de variables et constantes en n'utilisant que des caractères alphabétiques d'abord suivi de chiffres et/ou du caractère de soulignement _

Un identificateur ne doit jamais contenir d'espace. Par exemple : prix_ttc, p_rayon, g_rayon, ou rayon1 rayon2

Types de variables et constantes

Les variables et constantes peuvent être de types différents.
Les types les plus simples sont :

Types de variables et constantes

Les variables et constantes peuvent être de types différents.
Les types les plus simples sont :
- numériques : réels (float) ou entiers (integer).

Types de variables et constantes

Les variables et constantes peuvent être de types différents.
Les types les plus simples sont :
- numériques : réels (float) ou entiers (integer).
- chaîne de caractères (string).

- booléen (2 valeurs possibles Vrai (true) et Faux (false))

- booléen (2 valeurs possibles Vrai (true) et Faux (false))
Il existe d'autres types : date (j/m/a), monétaire (2 chiffres après la virgule) etc.

Déclaration des variables et constantes

Certains langages de programmation (à typage statique) nécessitent la déclaration des variables et constantes ainsi que leurs types avant de les utiliser.

Déclaration en pseudo code

```
Constantes
constante1=..., constante2=...
Variables
Variable1, Variable2 : entiers
Variable3, Variable4, Variable5 : réels
```

Exemple

```
Calcul de la surface d'un disque
Constantes
Pi=3,1415
Variables
Rayon, Surface : Réels
Début
lire(Rayon)
Surface =Pi*Rayon2
écrire (surface)
Fin
```

D'autres langages (typage dynamique) les types de constantes et variables sont reconnus automatiquement.

```
Calcul de la surface d'un disque
Pi=3,1415
Début
lire(Rayon)
Surface =Pi*Rayon2
écrire (surface)
Fin
```

Chaîne en pseudo-code

En pseudo-code, une chaîne de caractères est toujours notée entre guillemets anglaises " " ou apostrophes ’ ’ 567 est un nombre "567" est de type caractère représente la suite 5,6,7

Booléens en pseudo-code

En pseudo-code, on représente les variables de types booléens par VRAI et FAUX.

Booléens en pseudo-code

En pseudo-code, on représente les variables de types booléens par VRAI et FAUX.
le type booléen est très économique en termes de place mémoire occupée, puisque pour stocker une telle information binaire, un seul bit suffit.

Affectation

L'affectation consiste à attribuer une valeur à une variable. Elle traduit bien le "devient égale à ". En pseudo-code, l'instruction d'affectation se note avec le signe $\leftarrow$

Par exemple

r $\leftarrow$ 24 attribut 24 à la variable r.

Par exemple

r $\leftarrow$ 24 attribut 24 à la variable r.
sous-entend impérativement que r soit une variable de type numérique. Si r a été défini dans un autre type, cette instruction provoquera alors une erreur.

On peut bien attribuer à une variable la valeur d'une autre variable. Par exemple r $\leftarrow$ t

On peut bien attribuer à une variable la valeur d'une autre variable. Par exemple r $\leftarrow$ t
Une variable peut bien être affectée du résultat d'une expression qui contient cette même variable.

On peut bien attribuer à une variable la valeur d'une autre variable. Par exemple r $\leftarrow$ t
Une variable peut bien être affectée du résultat d'une expression qui contient cette même variable.
Par exemple i $\leftarrow$ i+1 (ce qui est faux mathématiquement)

Opérateurs arithmétiques

Un opérateur est un signe qui relie deux valeurs, pour produire un résultat :

Opérateurs arithmétiques

Un opérateur est un signe qui relie deux valeurs, pour produire un résultat :

+ : addition
- : soustraction
* : multiplication
/ : division
** : puissance
mod : reste de la division euclidienne

on a le droit d'utiliser les parenthèses, avec les mêmes règles de priorité qu'en mathématiques. Opérateur alphanumérique : & (ou +).
Cet opérateur permet de concaténer deux chaînes de caractères.

Opérateurs logiques (ou booléens)

Il s'agit du ET, OU, NON (et de XOR. Nous y reviendrons plus tard).
En pseudo code on utilise ces mêmes opérateurs. C'est à dire +, -, * &, ET, OU, etc

Expression

Une expression est un ensemble de valeurs de même type, reliées par des opérateurs, et équivalent à une seule valeur de ce même type.
Dans une affectation, à gauche de la flèche, il n'y a que le nom d'une variable, et uniquement ce nom ; à droite il y a une expression.

Exemple

A $\leftarrow$ "Extra"
B $\leftarrow$ "ordinaire"
C $\leftarrow$ A & B

Ordre des instructions

Dans un algorithme l'ordre des instructions est très important.

En PYTHON, on a pas besoin de déclarer une variable avant de s'en servir. Python reconnaît automatiquement le type de chaque variable.

En PYTHON, on a pas besoin de déclarer une variable avant de s'en servir. Python reconnaît automatiquement le type de chaque variable. Python est à typage dynamique

En PYTHON, on a pas besoin de déclarer une variable avant de s'en servir. Python reconnaît automatiquement le type de chaque variable. Python est à typage dynamique
Le signe d'affectation est = .

En PYTHON, on a pas besoin de déclarer une variable avant de s'en servir. Python reconnaît automatiquement le type de chaque variable. Python est à typage dynamique
Le signe d'affectation est = .
La fonction type() permet de déterminer le type d'une variable ou d'une constante.

Les opérateurs numériques sont +, -, * , /, (division entière) , ./ division, ** puissance. Une chaîne de caractères est toujours mise entre ' ' ou " " L'opérateur + pour concaténer deux chaînes. (en pseudo code c'est &).

Exemple

```
>>>  x=5
>>>  type(x)
<type 'int'>
>>>  y=5.0
>>>  type(y)
<type 'float'>
>>> a='Bonjour'
>>>  type(a)
<type 'str'>
```

Autres types en Python En python, on trouve les types : liste, dictionnaire et tulpe.

```
L=['a',5,5.]  # liste définie grâce aux [ ]
D={"Janvier":1, "Février":2, "Mars":3}  # dictionnaire défi
T=('a',6,'g')   #  tulpe défini grâce aux ( )
```