

# Introduction à l'algorithmique

## Actions de base

Mehdi ROUAN-SERIK

Institut de maintenance et de sécurité industrielle, IMSI

Université d'Oran 2 Mohamed Ben Ahmed

`mehdi.rouan@gmail+L1HSI.com`

21 octobre 2022

# Introduction

# Introduction

- On parle d'action ou d'instructions.
- On en distingue des instructions simples et d'autres composées.
- Action eg : mettre des ingrédients dans une casserole.
- Instruction eg : Résoudre un problème à l'aide d'une machine.

Voici les quatre instructions simples :

1. Déclaration.
2. Affectation.
3. Lecture.
4. Écriture.

# Déclaration

# Déclaration

- Donner un **identificateur** (ou un nom) à une variable.
- Fournir un **type** (ou une forme) pour cette variable.
- Un identificateur (unique) pour chaque variable du problème.

Identificateur 

10
----

# Identificateur

- Il doit commencer par une lettre de l'alphabet obligatoirement.
- Il ne doit contenir aucun séparateur (l'espace, la tabulation, + , - , . , %,, ...) sauf le caractère souligné (\_).
- Il ne doit pas contenir de lettres accentuées ni de ligatures : (é, è, ç, ô, œ, à,..).
- Il ne doit pas être un mot clé : (**var**, **entier**, **réel**, **Algorithme**, ...).
- Il doit refléter le rôle joué par par cette variable dans le problème.

# Exemples

## Identificateurs non valides

- ~~A~~B
- Température
- ~~note~~info
- ~~note~~<sub>1</sub>

## Identificateurs valides

- A
- Temperature
- note\_info
- note1

# Syntaxe d'une déclaration

- Syntaxe : Façon générale d'écrire.
- Chaque instruction à sa propre syntaxe.
- Exemple : schéma type d'un algorithme.

- Syntaxe :

```
var identificateur : type ;  
liste_identificateurs : type ;
```

- Exemple :

```
var Temperature : réel ;  
i,j,k : entier ;
```

# Affectation

# Affectation

- Donner une valeur à une variable.
- Copier en mémoire la valeur dans la variable.
- Une affectation **écrase** le contenu de la variable.
- L'ordre d'exécution des instructions dans un algorithme est très important.

- Syntaxe :

identificateur  $\leftarrow$  valeur ;  
identificateur  $\leftarrow$  expression ;

- Exemple :

```
Temperature  $\leftarrow$  37.5 ;  
Taux  $\leftarrow$  (Prix*19)/100 ;  
i  $\leftarrow$  i+1;  
Symbole  $\leftarrow$  '%' ;  
Prenom  $\leftarrow$  "Mehdi";  
Etat  $\leftarrow$  Vrai;
```

# Exemple

**Problème 1.** Déroulement :

1. Donnez la valeur de chacune des trois variables après chaque instruction.
2. Que fait cet algorithme ?

**Algorithme :** Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

Z ← 0;

**Fin.**

# Solution

**Algorithme** : Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

**Fin.**

X	Y	Z
---	---	---

# Solution

**Algorithme** : Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

**Fin.**

	X	Y	Z
1	17		

# Solution

**Algorithme** : Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

**Fin.**

	X	Y	Z
1	17		
2	17	12	

# Solution

**Algorithme** : Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

**Fin.**

	X	Y	Z
1	17		
2	17	12	
3	17	12	17

# Solution

**Algorithme** : Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

**Fin.**

	X	Y	Z
1	17		
2	17	12	
3	17	12	17
4	12	12	17

# Solution

**Algorithme** : Execution ;

**var** X,Y,Z : entier ;

**Début**

X ← 17;

Y ← 12;

Z ← X;

X ← Y;

Y ← Z;

**Fin.**

	X	Y	Z
1	17		
2	17	12	
3	17	12	17
4	12	12	17
5	12	17	17

# Lecture

# Lecture

- Introduire une valeur à une variable à partir d'une unité d'entrée.

- Syntaxe :

```
lire(identificateur);  
lire(liste_identificateurs);
```

- Exemple :

```
lire(Temperature);  
lire(i,j,k);
```

# Écriture

# Écriture

- **Afficher** la valeur d'une variable sur un écran.

- Syntaxe :

Affichage d'un message :

```
ecrire("message");
```

Affichage du contenu d'une variable :

```
ecrire(identificateur);
```

- Exemple :

```
A ← 5;
```

```
ecrire("La valeur de A =");
```

```
ecrire(A);
```

# Expressions & Actions composés

# Expressions

- Expression : ensemble de valeurs reliées par des opérateurs où le résultat final est une valeur.
  - Expressions algébriques pour les types : **entier** et **réel**.
  - Expressions logiques pour le type **booléen**.
- `25 div 3 mod 2`
  - `-25+(3-6)*2+5`
  - `a*5+3*2/b`
  - `A Et B Ou Non(C Et D)`
  - `"Bon" || "jour" || C`

# Hiérarchie des opérateurs

- Faux résultats, si on ne connaît pas les priorités entre les opérations!
- $25*3+6 = 225$  si on exécute  $+$  en premier, au lieu de 81.
- Hiérarchie pour opérateurs algébriques :

Opérateurs	Hiérarchie
( )	1
^	2
- (signe moins)	3
*, /, div, mod	4
+, -	5

- Opérateurs logiques :

Opérateurs	Hiérarchie
( )	1
Non	2
Et, Ou	3

# Actions composées

- Suite d'actions simples.
- Un algorithme : action.s composée.s.

- Exemple :

**Algorithme** : Exemple

**var** a,b : réel ;

**Début**

```
    écrire("Donnez la valeur de a");  
    lire(a);  
    écrire("Donnez la valeur de b");  
    lire(b);  
    a ← a*b+a/2;  
    écrire("a = ",a);
```

**Fin.**