

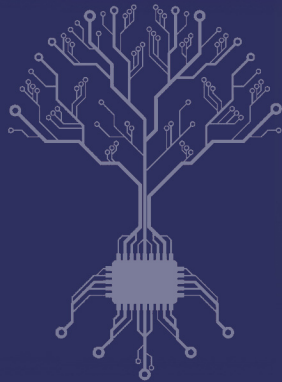


# ALGORITHMIQUE ET PROGRAMMATION

- 1 Notion d'algorithme
  - Données
  - Instructions de base

- 2 Notion de programme

# NOTION D'ALGORITHME



## Définition

Un algorithme est une suite d'instructions exécutées dans un ordre pour réaliser une tâche.

## Exemple : Calcul de la moyenne

Pour calculer la moyenne on peut suivre les instructions suivantes :

- Demander les notes (les données ) des matières
- Additionner les notes
- Diviser la somme par le nombre des matières
- Retourner le résultat

Alors on vient de concevoir un algorithme, exécutable par un être humain, pour calculer la moyenne des notes.

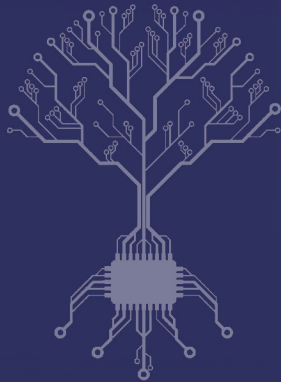
Et pour un ordinateur ? ! comment peut-on écrire cet algorithme ?

Dans le cas d'un ordinateur, après la déclaration des données, on utilise des instructions exécutables par ce dernier.

Les trois instructions de base sont :

*L'écriture, la lecture et l'affectation.*

# NOTION D'ALGORITHME : DONNÉES



## TYPES DE DONNÉES

Avant d'écrire l'algorithme, on déclare les données nécessaires à ce dernier.  
On trouve quatre types de données de base :

Type	Entier	Réel	Chaîne de caractères	Booléen
Exemples	...	...		
	-2	-7,7	'أحمد'	
	-1	-1/3		
	0	0,5	'Something159'	True ou 1
	1	1	'Ma classe'	False ou 0
	2	$\pi$	...	
	...	...		

**Table** – Types de données



# DÉCLARATION DES DONNÉES

Une donnée est déclarée sous la forme d'une variable.

## Définition

Une variable est un objet stocker dans la mémoire avec un nom unique, un type et une valeur, cette valeur peut être modifiée pendant l'exécution de l'algorithme.

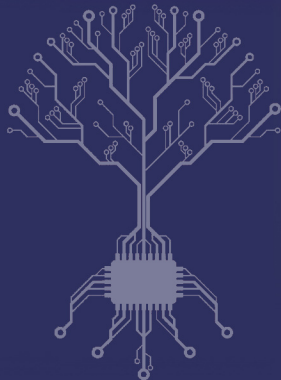
## Syntaxe

variable nom : type

## Exemples

- variable number\_students : entier
- variable GDP : réel
- variable name : chaîne de caractères

# NOTION D'ALGORITHME : INSTRUCTIONS DE BASE



# L'ÉCRITURE

Lorsqu'on a besoin d'écrire une chaîne de caractères sur l'écran on utilise l'instruction *écrire*.

## Syntaxe

On trouve deux syntaxes différentes :

→ Pour afficher une chaîne de caractères telle qu'elle est :

*écrire('chaîne de caractères')*

→ Pour afficher la valeur d'une variable :

*écrire(variable)*

## Exemple 1

*écrire('Hello world')* affiche *Hello world* sur l'écran.

## Exemple 2

Supposons maintenant qu'on a une variable *note* = 17, alors *écrire(note)* affiche 17 sur l'écran.

## Exemple 3

On peut combiner les deux cas en les séparant par une virgule :  
*écrire('La note est :', note)*, d'où *La note est : 17* s'affiche sur l'écran.

## LA LECTURE

Lorsqu'un algorithme a besoin d'une donnée (valeur d'une variable) on utilise l'instruction *lire*.

## Syntaxe

*lire(variable)*

*lire(variable1, variable2)*

## Exemple

En utilisant *lire(note)* l'ordinateur va demander d'entrer la valeur de la variable *note* à partir du clavier.

# L'AFFECTATION

Cette instruction est utilisée lorsqu'on veut affecter une valeur ou le résultat d'une expression à une variable.

## Syntaxe

`variable`  $\leftarrow$  *valeur*

`variable`  $\leftarrow$  *expression*

## Exemple 1

```
name  $\leftarrow$  'Sarah'  
note  $\leftarrow$  17
```

## Exemple 2

```
 $n1 \leftarrow 1 + 5$   
 $n2 \leftarrow n1 - 1$ 
```

Ici on affecte le résultat de l'expression  $1 + 5$  à la variable  $n1$ , et le résultat de  $n1 - 1$  à la variable  $n2$ . Alors  $n1 = 6$  et  $n2 = 6 - 1 = 5$ .

## Exemple de la moyenne

Finalement notre algorithme de la moyenne sera de la forme suivante :

variable note1 : réel

variable note2 : réel

...

variable nombre\_notes : entier

variable moyenne : réel

écrire('Entrer les notes :')

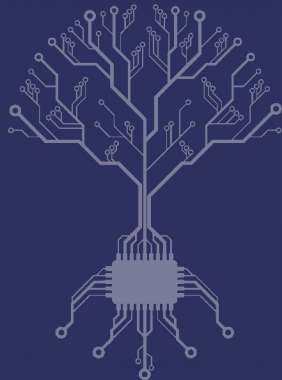
lire(note1, note2, ...)

$\text{moyenne} \leftarrow (\text{note1} + \text{note2} + \dots) / \text{nombre\_notes}$

écrire('La moyenne est :', moyenne)



# NOTION DE PROGRAMME



## Définitions

- Un programme est un ensemble d'instructions exécutables par l'ordinateur. Il n'est qu'un algorithme écrit non en langage naturel mais en langage de programmation.
- Un langage de programmation est un langage compréhensible par l'ordinateur. Plusieurs langages de programmation sont créés et chacun d'entre eux a ses propres règles d'écriture (syntaxes), alors pour qu'un programme fonctionne il faut, premièrement, respecter les règles d'écriture du langage choisi.

## Exemples de langages de programmation

*C, C++, Python, Java, ...*

Après le choix du langage de programmation on a besoin d'un *IDE(Integrated Development Environment)* qui est un environnement pour l'écriture et l'exécution des programmes. Dans le cas de *Python*, on en trouve plusieurs mais on utilisera *Jupyter Notebook* et *Spyder*.