

FORMATION PARTIE 1 : ALGORITHMES ET STRUCTURES DE DONNEES EN LANGAGE PYTHON.

Exemple : **14 juillet 1789 et 15 août 1960**

Données : Nombres et non numériques (caractères, images, sons, logiques, chaînes de caractères)

- Opérations arithmétiques : +, -, *, /
- Opérations logiques : ET, OU, NON
- Opérations de comparaison : =, <, >, <=, >=, <>
- Opérations de mouvement d'information (données)

Problème simple :

- 1) Dossier Algo 1
- 2) Dossier Algo 2

Dossier Algo 1 : Il s'agit ici d'analyser l'énoncé implicite du problème dans quatre (04) paragraphes.

- 1) Enoncé du problème
- 2) Résultats recherchés
- 3) Environnement (données, moyens)
- 4) Principe et méthode de résolution

Dossier Algo 2

C'est l'obtention de l'algorithme. Il ne s'obtient pas d'un seul coup. Il peut y avoir des transformations successives et la dernière transformation est l'algorithme recherché. Il faut enfin concevoir un jeu d'essai qui est un ensemble de données concrètes pour une exécution de l'algorithme.

Notion de variables :

Nom_variable ← valeur_variable

← (signe d'affectation)

NB : notation de variables (Camel case, Snake case)

Exemples :

premierNombre ← 0

premier_nombre ← 0

premiere_valeur ← 2

Ecrire() = Afficher()

Ecrire(premierNombre) ----- 0

Afficher(premierNombre) ----- 0

Types de variables :

- 1) Char (caractère)
- 2) String (chaîne de caractères)
- 3) Integer (entier)
- 4) Float (réel)
- 5) Boolean (Booléen) --- (True, False) ou (vrai ou faux) ou encore (1 ou 0) --- (oui ou non)

Ecrire('c')

Ecrire('Bonjour')

ma_valeur ← 12 en python (← c'est =)

Ecrire(ma_valeur) en python (Ecrire() c'est print())

monReel ← 7.2

Afficher(monReel)

mon_choix ← faux

Afficher(mon_choix)

message ← 'E'

Ecrire(message)

dire_bonjour ← 'Bonjour, le monde !'

Ecrire(dire_bonjour)

Lire() = Saisir() en python c'est input()

Ecrire('Tape une valeur : ')

Lire(x)

Exercice

Ecrire un algorithme qui fait la somme de deux entiers

Algorithme = pseudo code

Méthode 1 :

Solution

I) Dossier Algo 1 :

- 1) Enoncé du problème
Ecrire un algorithme qui fait la somme de deux entiers
- 2) Résultats recherchés
La somme de deux entiers \rightarrow s (entier)
- 3) Environnement (données, moyens)
a (entier)
b (entier)
s (entier)
- 4) Principe et méthode de résolution
Lire(a)
Lire(b)
 $s \leftarrow a + b$

Ecrire(s)

II) Dossier Algo 2

Lexique des objets	ORD (ordonnancement)	Définitions explicites
a (entier) b (entier) s (entier)	1 2 3 4 5 6	somme Ecrire('Tapez la valeur de a : ') Ecrire('Tapez la valeur de b : ') Lire(a) Lire(b) $s \leftarrow a + b$ Ecrire(s)

Méthode 2

Début

```
// Déclaration des variables  
Entier a  
Entier b  
Entier s
```

```
Ecrire('Tapez la valeur de a : ')  
Ecrire('Tapez la valeur de b : ')  
Lire(a)  
Lire(b)  
s  $\leftarrow$  a + b  
Ecrire(s)
```

Fin

Coder = programmer

Codage = programmation

Le code = Algorithme

Structure conditionnelle :

Si condition Alors

 Action

Finsi

(condition = opérations de comparaison) : =, <, >, <=, >=, <>

Si 5 > 2 Alors

 Ecrire('Vraie ')

Finsi

Si en python c'est if

Si condition Alors

```
    Action1
  Sinon
    Action2
Finsi
```

```
Si 5 > 2 Alors
  Ecrire('Vraie')
Sinon
  Ecrire('Faux')
Finsi
```

Sinon en python c'est else

Structure conditionnelle imbriquée

```
Si condition1 Alors
  Action1
  Sinon si condition2 Alors
    Action2
  Finsi
  Sinon si condition(n) Alors
    Action(n)
  Finsi
  Sinon
    Action
Finsi
```

```
Si 5 > 2 Alors
  Ecrire('Vraie')
```

Sinon si $2 > 5$ Alors

Ecrire('Faux')

Finsi

Sinon

Ecrire('Condition impossible')

Finsi

Sinon si en python c'est elif

Structures de contrôles (boucles)

- Boucle Pour :

Pour compteur \leftarrow valeur_initiale jusqu'à valeur_finale

Faire

Suite d'opérations

incrémentation

Finfaire

NB : incrémentation : $i \leftarrow i + 1$ et compteur : i

Pour $i \leftarrow 1$ jusqu'à 10

Faire

Ecrire('Bonjour')

$i \leftarrow i + 1$

Finfaire

Jeu d'essai :

Pour $i = 1$ bonjour

Bonjour

$i = 1 + 1 = 2$

$i = 2$

$i = 2 + 1 = 3$

$i = 10 + 1 = 11$

'Pour' en python c'est for

- Boucle : Tant que

Tant que condion

Faire

Suite d'instructions

Incrémentation

Finfaire

$i \leftarrow 5$

Tant que $i < 5$

Faire

Ecrire('Bonjour, les amis de la data !')

$i \leftarrow i + 1$

Finfaire

Jeu d'essai :

$l = 1$

$1 < 5$

$l = 1 + 1 = 2$

$2 < 5$

$l = 2 + 1 = 3$

$3 < 5$

$l = 3 + 1 = 4$

$4 < 5$

$l = 4 + 1 = 5$

$5 < 5$

'Tant que' en python c'est while

Boucle : Répéter

Répéter

Suite d'instructions

Incrémentation

Jusqu'à condition

$i \leftarrow 0$

Répéter

Ecrire('Salut, les amis de la data !')

$i \leftarrow i + 1$

Jusqu'à $i < 5$

Notion des tableaux

T

0	1	2	3	4	5	6	7	8	9
5	0	2	3	1	4	6	0	1	5

Taille = 10

Pas = taille - 1

= 10 - 1

Pas = 9

T[0]

Ecrire(T[7])

Indice = position = index

T (tableau (1..n) d'entiers)

X	0	1	2	3
	2	0	1	4

Taille = 4

Pas = taille - 1 = 4 - 1 = 3

$X[0] \leftarrow 2$

$X[1] \leftarrow 0$

$X[2] \leftarrow 1$

$X[3] \leftarrow 4$

Notion de fonctions :

- Procédure

Procédure nom_procédure(paramètres)

 Début

 Instructions

 Fin

Procédure Somme(x1, x2)

 Début

$s \leftarrow x1 + x2$

 Ecrire(s)

 Fin

Début

 Ecrire('Tape la valeur de a : ')

 Ecrire('Tape la valeur de b : ')

 Lire(a)

 Lire(b)

 Somme(a, b)

Fin

Procédure fonction :

Fonction nom_fonction(paramètres) : type_fonction

 Début

 Retourne résultat

 Fin

Programme Carre

Entier nombre \leftarrow 5

Fonction Carre(nombre) : entier

Début

Retourne nombre * nombre

Fin

Début

Entier resultat

resultat \leftarrow Carre(nombre)

Ecrire(resultat)

Fin

Retourne en python c'est return

NB : pour définir une fonction procédure ou une procédure en utilise le mot clé **def**

Modulo = mod en python %

= en python c'est ==

ET en python c'est and ou &

OU en python c'est or ou |

< > En python c'est !=