

**Séance 1 : Rappel structure Simple**

Groupe BacSc

**I. LES STRUCTURES SIMPLES**

Une structure simple peut être soit :

**1. Une instruction de lecture (entrée, saisie) de données**

L'instruction qui permet à l'utilisateur d'entrer (lire ou saisir) des valeurs au clavier (entrée Standard) (ou autre source d'entrée) s'appelle opération de lecture (ou d'entrée).

Syntaxe en algorithme	Syntaxe en python
Ecrire ('donner un variable'), <b>lire</b> (variable)	Variable = <b>input</b> ('donner un variable')

**Remarques :** Dès que le compilateur rencontre une instruction de lecture, l'exécution s'arrête en attendant que l'utilisateur saisisse une valeur pour la variable puis il valide la saisie par la touche Entrée. Après la validation, le compilateur passe à l'instruction suivante.

**2. Une instruction d'écriture (sortie, affichage) de résultat**

L'instruction qui permet d'afficher (d'écrire ou de sortir) des résultats (texte ou des valeurs) à

l'écran (sortie standard) (ou autre périphérique de sortie) s'appelle opération de sortie.

Syntaxe en algorithme	Syntaxe en python
Ecrire ('texte')	<b>print</b> ('texte')
Ecrire (variable)	<b>print</b> (variable)
Ecrire ('texte', variable)	<b>print</b> ('texte', variable)

**Remarques :** On peut afficher le résultat d'une expression par : Ecrire (expression). Exemples :

Ecrire (x/2-1), Ecrire (X>Y) ...

### 3. Une affectation (modification du contenu d'une variable)

L'affectation est l'instruction qui permet d'attribuer (affecter) une valeur à une variable ou de modifier la valeur qu'elle a déjà. Sa syntaxe est :

Syntaxe en algorithme	Syntaxe en python
Variable $\leftarrow$ valeur	Variable = valeur

**Remarque :** Il doit y avoir une comptabilité entre le type de la variable et celui de l'expression.

## Les types numériques

### Le type entier :

Si une variable est déclarée sous un type entier cela signifie que ses valeurs possibles sont les entiers appartenant à un certain intervalle [Minint, Maxint]. L'ensemble des entiers appartient à l'intervalle [-32768, 32767]. Il existe des sous types du type entier : entier court, entier long,

...

Le type entier en python : **int ()**

Les opérateurs arithmétiques qu'on peut appliquer sur une variable de type entier sont : l'addition (+), la soustraction (-), la multiplication (\*), la division entière (div), le reste de la division entière (mod) et le changement de signe (-).

Les opérateurs relationnels sont : <, >, ≤, ≥, ≠, =

### Le Type réel :

Comme pour les entiers, l'ensemble des nombres réels informatiques n'est pas  $\mathbb{R}$ , c'est un ensemble fini. Les valeurs d'une variable de type réel sont donc délimitées par un intervalle de validité (correspondant cette fois à un espace de 8 octets).

Le type réel en python : **float ()**

**Remarque :** Les opérateurs arithmétiques qu'on peut appliquer sur une variable de type réel sont : l'addition (+), la soustraction (-), la multiplication (\*), la division (/) et le changement de signe (-).

Les opérateurs relationnels sont : <, >, ≤, ≥, ≠, =

Tableau de déclaration des opérateur arithmétique, relationnels et logique.

Les opérateur	Syntaxe en algorithme	Syntaxe en python
L'addition	(+)	+
La soustraction	(-)	-
La multiplication	(*)	*
La division	(/)	/
La division entière	Div	//
Le reste de la division entière	Mod	%
Strictement supérieur	>	>
Strictement inférieur	<	<
Supérieur ou égale	>=	>=
Inférieur ou égale	<=	<=
Affectation	←	=
Différent	≠	!=
La comparaison	=	==
Et	Et	And
Ou	Ou	Or

**Exercice d'application :** Saisir un entier composé de 2 chiffres et calculer la somme de ses chiffres.

**Correction algorithme :**

**Début chiffres**

Ecrire ('donner un entier de deux chiffres')

Lire(n) Stocker

$d \leftarrow n \text{ div } 10$

$u \leftarrow n \text{ mod } 10$

le quotient de division euclidienne de (n sur 10) dans une variable d.

10 Stocker le reste de division euclidienne

somme  $\leftarrow d + u$  de (n sur 10) dans une variable u.

Ecrire ('la somme des chiffres', d, 'et', u, 'est', somme)

**Fin chiffres**

**Exercice 1 :** calculer et afficher la somme de chiffres d'un nombre de trois chiffres donné :

En algorithme	En python
Algorithme somme Début Ecrire ('donner un entier') Lire (n) $C \leftarrow n \text{ div } 100$ $D \leftarrow (n \text{ mod } 100) \text{ div } 10$ $U \leftarrow n \text{ mod } 10$ $S \leftarrow C + D + U$	<pre> n=int(input('donner un entier de trois chiffres :')) c=n//100 d=(n%100)//10 u=n%10 s=c+d+u print('la somme de chiffres est',s)                     </pre>

Ecrire ('la somme de chiffres est',S) Fin	
----------------------------------------------	--

TDO

objet	Type
n	entier
C,D,U,S	entier

Les fonctions Arithmétique :

Fonction	En python	Rôle	Exemple
Tronc (x) & Ent()	Le module qui gère Tronc en python est le module math : from math import * print ( <b>trunc</b> (x))	Extraire la partie entière de x (supprime la partie décimale)	Tronc (3.14) = 3 Tronc(3 .6) = 3 Ent(5.99)=5
Arrondi (x)	round (x)	Arrondir une valeur réelle à l'entier le plus proche.	Arrondi (9.499) = 9 Arrondi (2.5) = 2
Abs (x)	Abs(x)	Donner la valeur absolue de x	abs (-5) = 5
Racine carrée (x)	from math import * print(sqrt(12.5))	Donner la racine carrée de x (si x est positif)	Racine carre (16) = 4.0
Carrée (x)	x ** 2	Donne le carré de x	Carre (5) = 25
Alea(1,99)	from random import * print(randint(1,99))	Donne un nombre aléatoirement entre 1 et 99.	Alea(1,99)

**Exercice 2 :**

Ecrire un programme en python qui permet de convertir un temps donné en secondes en heures, minutes et secondes.

Exemple : Si temps = 3674 alors le programme affichera : 3674 s = 1 h : 1 min : 14 s

Algorithme	Python
Algorithme Temps Début Ecrire ('donner un entier') Lire(N) $H \leftarrow N \text{ div } 3600$ $M \leftarrow (N \bmod 3600) \text{ div } 60$ $S \leftarrow N \bmod 60$ Ecrire(H, 'heure ', M, 'Minute', S, 'Sec') Fin	<pre> n=int(input('donner un entier')) h=n//3600 m=(n%3600)//60 s=n % 60 print(h, 'heure',m, 'minute',s, 'seconde')                     </pre>