

Variables et instructions élémentaires.

I Variables, instructions, et premières fonctions.

I-1 Les variables.

En programmation une *variable* est le nom donné à quelque chose (nombre en général) de mémorisé dans la machine.

Remarques.

1. La variable en programmation n'est pas celle des mathématiques.
2. Si la variable avec la calculatrice est un nombre (décimal) avec Python il y a différents types de variables (cf *infra*).

I-2 Affectation

L'instruction qui consiste a créer une variable, *i.e.* enregistrer quelque chose dans une variable, est appelée une *affectation* (ou *assignation*).

Par exemple pour enregistrer le nombre 3 dans la variable C :

Scratch 	Algorithme $C \leftarrow 3$	Python 3 $C=3$	Ti $3 \rightarrow C$
---	--------------------------------	-------------------	-------------------------

Remarques.

1. Avec Python le nom de la variable peut être un mot, par exemple *lavariablc*, mais avec la calculatrice la variable est une seule lettre, C .
2. Contrairement au " = " des mathématiques, la commande d'affectation de Python " = " n'est pas symétrique. Il ne faut pas écrire $3=C$.
3. Si la variable C était déjà utilisée, affecter $5 + C$ à C efface la précédente valeur enregistrée, 3, dans C et la remplace par $5 + 3$.

I-3 Afficher.

Pour voir ce qui a été enregistré dans une variable nous utilisons une instruction d'affichage.

Par exemple pour afficher le contenu de la variable C :

Scratch 	Algorithme Afficher C	Python 3 print (C)	Ti Disp C
---	----------------------------	----------------------------------	----------------

I-4 Séquence.

Dans les langages de programmation que nous utiliserons, chaque nouvelle instruction est écrite sur une nouvelle ligne.

La machine exécute les instructions les unes après les autres en allant de haut en bas.

Une suite d'instructions qui se suivent est appelée une *séquence*.

Par exemple

1	$C \leftarrow 3$
2	$C \leftarrow 5 + C$
3	Afficher C

est une séquence de trois instructions.

Exercice 1

1. Quels sont les contenus des différentes variables après exécution de l'algorithme ci-dessous quand n contient la valeur 2 et m la valeur 7.

1	$x \leftarrow 10n + m$
2	$y \leftarrow 100x$
3	$z \leftarrow 100y$
4	$s \leftarrow x + y + z$

2. Exécutez l'algorithme pour d'autres valeurs de m et de n , entières entre 1 et 9. Quelle est la forme du nombre contenu dans s après exécution?

_____ Correction exercice 1 _____

Variables	n	m	x	y	z	s
-----------	-----	-----	-----	-----	-----	-----

_____ Fin de la correction _____

Exercice 2

1. Écrivez l'algorithme (langage naturel) correspondant à ce script Python

$a=n-2$
$b=a*n$
$c=b+1$

2. Que contient chaque variable après l'exécution des instructions du script si $n = 5$? $n = 3$? $n = 1$? $n = 4$?
3. Émettez une conjecture puis démontrez-la.

I-5 Les différents types de variables.

Les variables ne servent pas à mémoriser que des nombres mais différents types d'objets :

Types	Exemple	Nom en Python	Explication
Entier	3	int	int pour integer
Flottant	5.4	float	nombre décimal avec virgule flottante (comme une écriture scientifique)
Chaîne de caractères	"LUT"	str	str pour string. Souvent des mots ou des phrases.
Booléen	$5 > 2 - 1$	bool	il s'agit d'une phrase mathématique à laquelle nous pouvons associer soit la valeur vraie (1) soit la valeur faux (0).

Remarques.

1. Si le typage des variables existe sur la calculatrice, il est moins explicite que dans Python. Nous n'utiliserons avec la calculatrice quasiment que des variables de type nombre.

II Blocs d'instructions.

Les blocs d'instructions sont des séquences d'instructions qui sont séparées du reste du programme soit parce qu'il faut les lire plusieurs fois, soit il ne faut les lire que sous certaines conditions.

II-1 Instruction conditionnelle.

Il est possible de n'effectuer un bloc d'instruction que si une certaine condition est vérifiée comme, par exemple le monstre qui surgit dans un jeu vidéo si l'avatar s'en est trop approché.

<p>Scratch</p> 	<p>Algorithme</p> <pre> 1 si distance ≤ 10 alors 2 afficher "Bouh!" 3 sinon 4 afficher "Tout va bien" 5 fin </pre>
<p>Python 3</p> <pre> if distance <= 10: print("Bouh!") else: print("Tout_va_bien") </pre>	<p>Ti</p> <pre> :If D≤10 :Then :Disp "BOUH" :Else :Disp "OK" :End </pre>

Exercice 3

Considérons l'algorithme ci-dessous

```

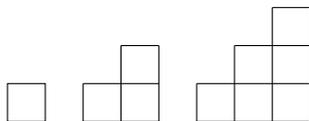
1 si a > b alors
2 | afficher b
3 sinon
4 | afficher a
5 fin

```

Que retourne cet algorithme si $a = 3$ et $b = 5$? $a = 2$ et $b = 0,5$? $a = 4$ et $b = 4$?

II-2 Boucle bornée (for).

Considérons un exemple. On construit un escalier de proche en proche.



Étape 1 Étape 2 Étape 3

Nous souhaiterions connaître le nombre de carrés qui forment l'escalier à la vingtième étape. Pour cela nous utilisons un programme dans lequel nbc désigne le nombre de carrés dessinés.

<p>Scratch</p> 	<p>Algorithme</p> <pre> 1 nbc ← 1 2 pour k allant de 2 à 20 faire 3 nbc ← nbc+k 4 fin </pre>
<p>Python 3</p> <pre> nbc=1 for k in range(2,20): nbc=nbc+k </pre>	<p>Ti</p> <pre> :1→N :For(K,2,20) :N+K→N :End </pre>

Exercice 4

On considère deux algorithmes suivants.

Algorithme 1

```

1 c ← 0
2 pour k allant de 1 à 6
  faire
3   | c ← k
4 fin

```

Algorithme 2

```

1 c ← 0
2 pour k allant de 1 à 6
  faire
3   | c ← c+k
4 fin

```

Indiquez le contenu de la variable c au fur et à mesure de l'exécution de chacun des algorithmes.

Exercice 5

Quelle sortie obtenez-vous en exécutant le deux scripts suivants?

Algorithme 1

```

a=5
for k in range(5):
    a=10*a-8
print(a)

```

Algorithme 2

```

a=5
for k in range(5):
    a=10*a-8
print(a)

```

II-3 Boucle non bornée (while)

Si, sur un jeu d'échecs, nous déposons un grain de riz sur la première case, puis deux sur la seconde, quatre sur la suivante et ainsi de suite en doublant à chaque

fois, sur la $case$ déposérons-nous plus d'un million de grain sur une case ?

Dans cet exemple nous ne savons pas combien de fois il faudra doubler le nombre. Nous utilisons alors une boucle non bornée.

Notons $case$ la variable comptant le nombre de cases remplies et riz celle comptant le nombre de grain riz déposés dans une case.

Scratch	Algorithme
	<pre> 1 $case \leftarrow 1$ 2 $riz \leftarrow 1$ 3 tant que $riz \leq 1000000$ faire 4 $riz \leftarrow 2 \times riz$ 5 $case \leftarrow case + 1$ 6 fin 7 Afficher $case$ </pre>
<p>Python 3</p> <pre> case=1 riz=1 while riz <=1000000: riz=2*riz case=case+1 print (case) </pre>	<p>Ti</p> <pre> :1→C :1→R :While R≤1000000 :2*R→R :C+1→C :End :Disp C </pre>

III Python.

Les opérateurs de calcul Python.

*	multiplication
+	addition
-	soustraction
/	division décimale
//	quotient de la division euclidienne
%	reste de la division euclidienne
**	exposant

Les opérateurs de comparaison Python.

Python	Mathématique	Exemple
<code>==</code>	$=$	<code>1 == 2</code> a la valeur faux
<code>!=</code>	\neq	<code>1 != 2</code> a la valeur vrai
<code><></code>	\neq	<code>1 <> 2</code> a la valeur vrai
<code>></code>	$>$	<code>1 > 2</code> a la valeur faux
<code><</code>	$<$	<code>1 < 2</code> a la valeur vrai
<code>>=</code>	\geq	<code>1 > 2</code> a la valeur faux
<code><</code>	\leq	<code>1 < 2</code> a la valeur vrai

Exercice 6

On affecte à n un entier naturel.

Reliez chaque instruction de la colonne de gauche à l'expression de la colonne de droite correspondante.

1.

`n*2`

2.

`n+1`

a. Le carré de n .

3.

`n%10`

b. L'entier qui précède n .

4.

`n**2`

c. Le double de n .

d. Le chiffre des unités de n .

5.

`n-1`

e. Le nombre de dizaines de n .

f. L'entier successeur de n .

6.

`n//10`

Exercice 7

Déterminez le type de la variable a dans chacun des cas suivants puis vérifiez votre réponse en utilisant la fonction `type()`. Par exemple tapez `type(2)` dans le shell pour le premier.

1.

`a=2`

2.

`a=4.5`

3.

`a=2+3.5`

4.

`a=51//6`

5.

`a=2.5+3.5`

6.

`a=14/2`

7.

`a="33"`

8.

`a=33`

9.

`a=2e10`

10.

`a=[4,6,12.3]`

IV Exercices.

Exercice 8

1. Qu'affiche le script Python suivant s'il est exécuté ?

```
a=6
b=3
a=a/b
b=b/a
c=a*b
print(c)
```

2. Si la valeur initiale de a est x et si celle de b est y , exprimez la valeur de c en fonction de x et y .

Correction exercice 8

1. Déterminons ce que retourne le script s'il est exécuté.

Variabes	a	b	c
1	6		
2	6	3	
3	2	3	
4	2	1,5	
5	2	1,5	3

Le programme affiche 3.

2. Exprimons c en fonction de x et y .

Reprenons le précédent tableau d'état des variables :

Variabes	a	b	c
1	x		
2	x	y	
3	$\frac{x}{y}$	y	
4	$\frac{x}{y}$	$\frac{y}{\frac{x}{y}} = \frac{y^2}{x}$	
5	$\frac{x}{y}$	$\frac{y^2}{x}$	$\frac{x}{y} \cdot \frac{y^2}{x} = y$

Si $a = x$ et $b = y$ alors $c = y$.

Fin de la correction

Exercice 9

Dans un magasin de photocopies les 20 premières photocopies sont facturées à 0,10 € et les suivantes à 0,08 €.

Complétez le programme suivant pour qu'il retourne le prix à payer p pour n photocopies. Puis programmez-le et donnez le prix à payer pour 350 photocopies.

```

1  $p \leftarrow 0$ 
2 si  $n \leq 20$  alors
3   | afficher ...
4 sinon
5   | afficher ...
6 fin

```

Correction exercice 9

```

1  p ← 0
2  si n ≤ 20 alors
3  |  afficher 0,10 × n
4  sinon
5  |  afficher
   |  0,10 × 20 + 0,08 × (p - 20)
6  fin

```

Pour 350 photocopies il fut payer 28,4 €.

_____ Fin de la correction _____

Exercice 10

Dites ce que renvoie le programme suivant.

```

1  x ← -3
2  si x < 0 alors
3  |  Afficher -x
4  sinon
5  |  Afficher x
6  fin

```

Que renvoie ce programme si maintenant $x = 2$? $x = -3,14$?

_____ Correction exercice 10 _____

Le nombre renvoyé par ce programme est appelé *valeur absolue de x*, que nous noterons $|x|$.

_____ Fin de la correction _____

Exercice 11

Expliquez le rôle du script Python suivant.

```

for k in range(11):
    print(8*k)

```

_____ Correction exercice 11 _____

Le script affiche la table de multiplication de 8.

_____ Fin de la correction _____

Exercice 12

Programmez l'algorithme suivant avec votre calculatrice et exécutez-le. Expliquez.

```

1  $n \leftarrow 10$ 
2 tant que  $n < 20$  faire
3   |  $n \leftarrow n - 1$ 
4 fin

```

Correction exercice 12

Le programme ne s'arrête jamais car n n'est jamais plus grand que 20. Il s'agit d'une boucle infinie. C'est ce qu'on appelle un bug.

Ce bug peut être utilisé sur votre calculatrice pour créer des programmes.

Fin de la correction

Exercice 13

Exécutez l'algorithme ci-dessous où a et b sont deux entiers et indique la valeur de c à la sortie quand $a = 235$ et $b = 13$, quand $a = 21$ et $b = 43$, quand $a = 84$ et $b = 7$.

```

1  $k \leftarrow 0$ 
2 tant que  $kb \leq a$  faire
3   |  $k \leftarrow k + 1$ 
4 fin
5  $c \leftarrow (k - 1)b$ 
6 Afficher  $c$ 

```

Que représente c pour a et b ?

Correction exercice 13

c est le plus grand multiple de b inférieur à a . Autrement dit la réponse à la question dans a combien de fois puis-je mettre b est $k - 1$.

Fin de la correction

Exercice 14

En partant de l'algorithme de l'exercice précédent rédigez un algorithme qui renvoie la division euclidienne de a par b .

Correction exercice 14

```

1  $k \leftarrow 0$ 
2 tant que  $kb \leq a$  faire
3   |  $k \leftarrow k + 1$ 
4 fin
5 Afficher "Quotient :",  $k - 1$ ,
   "Reste :",  $a - (k - 1)b$ 

```

Fin de la correction

Exercice 15

Rappelons que les opérandes `//` et `%` renvoient respectivement le quotient et le reste dans la division euclidienne.

1. Dans le programme suivant a et b sont des entiers naturels.

```
if a%b==0:
    print("oui")
else:
    print("non")
```

Si l'algorithme renvoie *oui* comment sont appelés b vis-à-vis de a et a vis-à-vis de b ?

2. Créez et programmez un algorithme en Python qui vérifie si un nombre n est premier.

Exercice 16

La formule de Tanner permet de prévoir la taille d'une personne à l'âge adulte connaissant la taille de ses parents. Pour un garçon, on ajoute à la demi-somme des tailles des deux parents 6,5. pour une fille on enlève 6,5 à la demi-somme des tailles des deux parents. Toutes les tailles sont exprimées en centimètres.

Concevez un algorithme qui, à partir de la taille des deux parents, calcule le résultat donné par la formule de Tanner. Puis programmez cet algorithme.

Exercice 17

Que calcule l'algorithme ci-dessous ? Programmez-le et testez-le en choisissant $n = 10$ puis $n = 100$.

```
n=4
f=1
for k in range(1, n+1):
    f=f*k
print(f)
```

Exercice 18

Nous voulons savoir à partir de quelle valeur de l'entier n la somme $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ dépasse 10.

1. Complétez l'algorithme suivant afin que la variable N contienne la valeur cherchée en fin d'algorithme.

```

1  N ← 1
2  S ← 0
3  tant que ... faire
4  |   S ← ...
5  |   N ← N + 1
6  fin

```

2. Programmez cet algorithme et déduisez-en la valeur recherchée.

Exercice 19

Réalisez un programme en Python qui vérifie si trois nombres forment un triplet pythagoricien.

Exercice 20

Exécutez le programme Python suivant pour différentes valeurs de a puis interprétez-le.

```

a=12
b=2
while a>1:
    while a%b==0:
        a=a/b
        print (b)
    b=b+1

```