



FICHE n°6 : PROGRAMMER DES BOUCLES

Syntaxe pour les boucles dans 4 langages de programmation :

Langage naturel	Python	Scilab	TI	CASIO
Tant que <i>Condition est vraie</i> Faire <i>Instructions</i>	while <i>Condition: Instructions</i>	while <i>Condition: Instructions</i> end	:While <i>Condition</i> <i>:Instructions</i> :End	While <i>Condition</i> ↓ <i>Instructions</i> ↓ WhileEnd ↓

Langage naturel	Python (*)	Scilab	TI	CASIO
Pour <i>i</i> allant de 3 à 7 Faire <i>Instructions</i>	for <i>i in range(3,8):</i> <i>Instructions</i>	for <i>i = 3:7</i> <i>Instructions</i> end	:For (<i>i,3,7</i>) <i>:Instructions</i> :End	For 3→ <i>i</i> To 7↓ <i>Instructions</i> ↓ Next

(*) En Python, **range(3,8)** désigne la séquence des entiers n vérifiant $3 \leq n < 8$.
range(10) désigne la séquence des entiers 0, 1, ..., 9.

Syntaxe pour sortir d'une boucle :

Langage naturel	Python	Scilab	TI	CASIO
Sortir de la boucle	<code>break</code>			Stop

Syntaxe pour générer un nombre aléatoire :

Langage naturel	Python (*)	Scilab	TI	CASIO
Générer un nombre entier aléatoire de l'intervalle [1,5]	<code>randint(1,5)</code>	<code>int(rand()*5)+1</code>	<code>randInt(1,5)</code>	<code>RanInt#(1,5)</code>

(*) En Python, la commande **randint** nécessite l'appel d'un module au début du programme. Pour ce faire, taper : **from random import***

Rappels :

Langage naturel	Python (*)	Scilab	TI	CASIO
Saisir A	<code>A=input('A=')</code>	<code>A=input("A=")</code>	Input A ou Prompt A	"A=" ?→A↓

Langage naturel	Python	Scilab	TI	CASIO
Si <i>Condition</i> Alors <i>Instructions1</i> Sinon <i>Instructions2</i>	if <i>condition:</i> <i>Instruction1</i> else: <i>Instruction2</i>	if <i>condition then</i> <i>Instruction1</i> else <i>Instruction2</i> end	:If <i>condition</i> :Then <i>:Instruction1</i> :Else <i>:Instruction2</i> :End	If <i>condition</i> ↓ Then <i>Instruction1</i> ↓ Else <i>Instruction2</i> ↓ IfEnd ↓

Exercice 1 :

Voici un algorithme écrit dans différents langages de programmation :

Python	Scilab	TI	CASIO
<code>for i in range(10):</code> <code>print(i)</code>	1 <code>for i=0:9</code> 2 <code>... afficher(i)</code> 3 <code>end</code>	<code>For(I,0,9)</code> <code>Disp I</code> <code>End</code>	<code>For 0→I To 9</code> ↓ <code>I</code> ↓ <code>Next</code> ↓

- 1) Tester cet algorithme à l'aide d'une calculatrice ou d'un logiciel. Qu'affiche-t-il en sortie ?
- 2) Rédiger et tester un programme qui affiche tous les entiers inférieurs à 16.
- 3) Rédiger et tester un programme qui affiche tous les entiers compris entre 18 et 45.

Exercice 2 :

Voici un algorithme écrit dans différents langages de programmation :

Python	TI	CASIO
<code>n=0</code> <code>while n<10:</code> <code>print(n)</code> <code>n=n+2</code>	<code>0→N</code> <code>While N<10</code> <code>Disp N</code> <code>N+2→N</code> <code>End</code>	<code>0→N</code> ↓ <code>While N<10</code> ↓ <code>Disp N</code> ↓ <code>N+2→N</code> ↓ <code>WhileEnd</code> ↓ (**)

- 1) Tester cet algorithme à l'aide d'une calculatrice ou d'un logiciel. Qu'affiche-t-il en sortie ?
- 2) Rédiger et tester un programme qui affiche tous les entiers pairs compris entre 18 et 45.
- 3) Rédiger et tester un programme qui affiche tous les entiers impairs compris entre 50 et 150.

(**) Sous Casio, l'algorithme affiche toujours par défaut la dernière variable stockée. Pour l'éviter, il suffit d'ajouter deux guillemets "" après la commande `WhileEnd`.

Exercice 3 :

Voici un algorithme écrit dans différents langages de programmation :

Python	Scilab	TI	CASIO
<code>S=0</code> <code>for i in range(101):</code> <code>S=S+i</code> <code>print(S)</code>	1 <code>S=0</code> 2 <code>for i=1:100</code> 3 <code>... S=S+i</code> 4 <code>end</code> 5 <code>afficher(S)</code>	<code>0→S</code> <code>For(I,1,100)</code> <code>S+I→S</code> <code>End</code> <code>Disp S</code>	<code>0→S</code> ↓ <code>For 1→I To 100</code> ↓ <code>S+I→S</code> ↓ <code>Next</code> ↓ <code>S</code> ↓

- 1) Recopier et compléter le tableau suivant par les premières valeurs prises par les variables S et i.

i		1	2	3									
S	0	1	3										

- 2) Quel problème permet de résoudre cet algorithme.
- 3) a) En s'inspirant des programmes précédents, écrire un programme (langage au choix) permettant de calculer la somme des entiers de 34 à 145.
Tester ce programme à l'aide d'une calculatrice ou d'un logiciel.
b) Même question pour la somme des entiers de 67 à 456.

Exercice 4 :

On place un capital de 500€ sur un compte rémunéré à 3% par an.
L'algorithme suivant, écrit en langage naturel, permet de calculer le nombre d'années au bout desquelles le capital sera doublé.

Langage naturel
Initialisation Affecter à S la valeur 500 Affecter à A la valeur 0
Traitement des données Tant que S<1000 Faire Affecter à S la valeur 1,03xS Affecter à A la valeur A+1
Sortie Afficher A

1) Chacun des programmes ci-dessous traduisant l'algorithme précédent comprend une erreur. Corriger un des programmes (au choix) et le tester à l'aide d'une calculatrice ou d'un logiciel.

Python	Scilab	TI	CASIO
<pre>S=500 A=0 while S<1000: S=1.03*S A=A+1 print(A)</pre>	<pre>1 S=500 2 A=0 3 while S<1000 4 S=1.03*S 5 A=A+1 6 end 7 afficher(A)</pre>	<pre>:500→S :0→A :While S<1000 :1.03*S→S :End :A+1→A :Disp A</pre>	<pre>500→S⇩ 0→A⇩ While S<1000⇩ 1.03×S→S⇩ WhileEnd⇩ A+1→A⇩ A.</pre>

2) Modifier le programme précédent de telle sorte que le capital et le taux de rémunération soient saisis en entrée.
L'algorithme affichera à nouveau en sortie le nombre d'années au bout desquelles le capital sera doublé.

Exercice 5 :

On dépose 25€ dans une tirelire.
L'algorithme suivant, écrit en langage naturel, permet de calculer le nombre de pièces de 1€ ou 2€ ajoutés de façon aléatoire dans la tirelire avant de dépasser 50€.

- 1) Pourquoi le programme affiche en sortie « D-1 » ?
- 2) Rédiger et tester un programme traduisant cet algorithme.
- Syntaxe pour un nombre aléatoire en page 1 de la fiche -

Langage naturel
Initialisation Affecter à S la valeur 25 Affecter à D la valeur 0
Traitement des données Tant que S<50 Faire Affecter à A la valeur aléatoire 1 ou 2 Affecter à S la valeur S+A Affecter à D la valeur D+1 Afficher A
Sortie Afficher D-1

Exercice 6 :

D'après "Document ressource pour la classe de seconde" – juin 2009

On demande à l'utilisateur de deviner en moins de six essais un nombre tiré au hasard entre 10 et 100.
On lui indique à chaque fois si le nombre proposé est supérieur ou inférieur au nombre cherché.

- 1) L'algorithme qui suit, écrit en langage naturel, permet d'effectuer le jeu.
 - a) Que représentent les variables E, S et N ?
 - b) Rédiger et tester un programme traduisant cet algorithme.
- Voir syntaxes en page 1 de la fiche -

2) Sans stratégie, il est difficile de gagner.
En effet, selon le choix des valeurs, il sera ou non possible de déterminer à coup sûr la solution.
La méthode consiste, en choisissant à chaque fois la valeur située au milieu de l'intervalle en cours, à réduire de moitié l'amplitude de l'intervalle dans lequel se trouve le nombre.
Tester cette stratégie pour gagner à tous les coups à ce jeu et détailler un coup gagnant.

Langage naturel
Initialisation Affecter à S la valeur d'un nombre aléatoire entier compris entre 10 et 100 Affecter à E la valeur 1
Traitement des données et entrée Tant que E<7 Saisir N Si N>S Alors afficher "C'est moins" Sinon Si N<S Alors afficher "C'est plus" Sinon Afficher "C'est gagné" Sortir de la boucle E=E+1
Sortie Si E=7 Alors afficher "C'est perdu"

Exercice 7 :

Rédiger et tester un programme permettant de calculer la somme des entiers naturels pairs inférieurs ou égaux à 1000.

Exercice 8 :

1) Compléter l'algorithme ci-dessous qui permet de déterminer la moyenne d'une série de notes, où N est le nombre de notes.

2) Rédiger et tester un programme traduisant cet algorithme.



Hors du cadre de la classe, aucune reproduction, même partielle, autres que celles prévues à l'article L 122-5 du code de la propriété intellectuelle, ne peut être faite de ce site sans l'autorisation expresse de l'auteur.

www.maths-et-tiques.fr/index.php/mentions-legales