ACTIVITÉ 4: RÉPÉTER DES INSTRUCTIONS EN PYTHON

Problématique: Comment répéter automatiquement une série d'instructions un nombre déterminé de fois ?

Objectifs: *À la fin de l'activité, vous serez capable de :*

- *comprendre le fonctionnement d'une boucle for ;*
- *utiliser range() pour déterminer les valeurs parcourues ;*
- *traduire une boucle « POUR » en langage Python ;*
- *utiliser une boucle pour automatiser des calculs répétitifs.*

Dans un programme, il est fréquent de devoir r**épéter plusieurs fois les mêmes instructions.** Nous avons déjà rencontré la **boucle while, qui permet de répéter des instructions tant qu'une condition est vérifiée**. Lorsque le **nombre de répétitions est connu à l'avance, on peut utiliser une boucle for**.

PARTIE 1 : DÉCOUVRIR LA BOUCLE « POUR »

On souhaite afficher les dix premiers multiples de 8. On considère l'algorithme suivant :

```
POUR i allant de 1 à 10
    afficher(8 × i)
```

1. Compléter le tableau suivant pour les premières étapes de l'algorithme :

Passage dans la boucle	Valeur de i	Valeur affichée
1	1	8
2	……	……
3	……	……
4	……	……
…		
10	……	……

2. Quelles sont les différentes valeurs prises par la variable i ? ……
3. Combien de fois l'instruction afficher(8 × i) est-elle exécutée ? ……
4. Quel est le rôle de la variable i dans cet algorithme ? ……

A RETENIR:

Une boucle POUR permet de répéter une ou plusieurs instructions lorsque le nombre de répétitions est connu à l'avance. La variable *i*, appelée compteur, prend successivement différentes valeurs.

PARTIE 2 : LA BOUCLE « FOR » EN PYTHON

En langage Python, l'algorithme de la partie 1 s'écrit :

```
for i in range(1,11):
    print(8*i)
```

A. Comprendre l'instruction RANGE()

1. *Partie sur ordinateur*
   a) Saisir le programme précédent dans Trinket et l'exécuter.
   b) Comparer les valeurs prises par i avec les nombres présents dans l'instruction `range(1,11)`
   Que remarquez-vous concernant la valeur 11 ?

   ……

   c) Remplacer successivement la première ligne par :

```
for i in range(1,5):
```

puis

```
for i in range(3,8):
```

Pour chacun des programmes, relever les différentes valeurs prises par `i`.

……

2. Compléter la phrase ci-dessous :

"Dans `range(a,b)`, le compteur prend successivement les valeurs entières de …… incluse à …… exclue."

A RETENIR:

La syntaxe d'une boucle `for` est :

```
for i in range(a,b):
    instruction
```

Le compteur i prend successivement toutes les valeurs entières de a incluse à b exclue.
Comme pour les conditions et les boucles `while`, les instructions à répéter sont indentées.

B. Traduire une boucle

On considère l'algorithme suivant et sa traduction en langage Python:

POUR i allant de 1 à 20 afficher(i^2)	`for i in range(……,……) :` `print(……)`

1. Combien de fois l'instruction afficher(i^2) est-elle exécutée ? ……
2. Quelle est la première valeur affichée ? La dernière ? ……
3. Compléter sa traduction en Python.
4. *Partie sur ordinateur*
   a) Saisir et compléter le programme dans Trinket.
   b) L'exécuter et vérifier les résultats obtenus.
   c) Modifier le programme afin d'afficher les cubes des entiers de 1 à 20.

APPEL → **Appeler le professeur pour vérification**

PARTIE 3 : EFFECTUER DES CALCULS RÉPÉTITIFS

A. Calculer une somme

On souhaite calculer la somme suivante : 1 + 2 + 3 + 4 + 5 + … + 100
On considère l'algorithme ci-dessous et sa traduction en langage Python:

somme ← 0
POUR n allant de 1 à 100
  somme ← somme + n
afficher(somme)
	`somme = …`
	`for ……. in …….. (……,……):`
	`  somme = ……..`
	`print(somme)`

1. Quelle est la valeur initiale de la variable somme ?
2. Compléter le tableau de variables :

Passage	**n**	somme **avant l'instruction**	somme **après** somme ← somme + n
1	1	0	1
2	2	1	……
3	3	……	……
4	4	……	……
5	5	……	……

3. Expliquer avec une phrase le rôle de l'instruction : `somme ← somme + n`

   ……

4. Compléter la traduction de l'algorithme.
5. *Partie sur ordinateur*
   a) Saisir et compléter le programme dans Trinket.
   b) L'exécuter et relever le résultat obtenu.
   c) Modifier le programme afin de calculer : 1 + 2+ 3 + 4 + 5 + … + 1000

APPEL → **Appeler le professeur pour vérification**

B. A vous de programmer

On souhaite maintenant calculer : $5^2 + 6^2 + 7^2 + \ldots + 25^2$

1. Compléter l'algorithme et traduire en langage Python

somme ← ……
POUR n allant de …… À ……
  somme ← ……
afficher(……)
	`somme = …`
	`for ……:`
	`  somme = ……`
	`print(……)`

2. *Partie sur ordinateur*
   a. Saisir votre programme dans Trinket et exécuter-le
   b. Vérifier que le programme fournit bien une seule valeur correspondant à la somme recherchée.

PARTIE 4 : RÉINVESTIR LES INSTRUCTIONS MATHÉMATIQUES (VOIR ACTIVITÉ 3)

On souhaite calculer : $\sqrt{1} + \sqrt{2} + \sqrt{3} + \ldots + \sqrt{50}$

1. Écrire un algorithme permettant de calculer cette somme et traduire le en langage Python:

2. *Partie sur ordinateur*

a) Saisir votre programme dans Trinket et exécuter-le.

b) Afficher le résultat arrondi à deux chiffres après la virgule.

APPEL → **Appeler le professeur pour vérification**

BILAN DE L'ACTIVITÉ 1.4 : LES BOUCLES FOR

Une boucle permet de répéter plusieurs fois une ou plusieurs instructions. Lorsque le nombre de répétitions est connu à l'avance, on peut utiliser une boucle FOR. L'instruction RANGE() permet de déterminer les valeurs successivement prises par le compteur de la boucle.

Dans:

```
for i in range(a,b):
    instruction
```

le compteur i prend les valeurs entières de a incluse à b **exclue**.

A RETENIR: DIFFÉRENCES ENTRE BOUCLE WHILE ET BOUCLE FOR (ACTIVITE 2)

Type de boucle	Quand l'utiliser ?	Python
TANT QUE	Le nombre de répétitions n'est pas connu à l'avance et dépend d'une condition.	`while condition:`
POUR	Pour parcourir successivement les éléments d'une séquence de nombres entiers connus à l'avance.	`for i in range(...):`