

Coder les séquences d'instructions suivantes et compléter les cases
« résultats affichés » et « observations ».

Séquence d'instructions 1	Séquence d'instructions 2	Séquence d'instructions 3
for n in range (5) : print (n) print (« FIN »)	for n in range (1,5) : print (n) print ("FIN")	for n in range (1,5,2) : print (n) print (« FIN »)
Résultats affichés :	Résultats affichés :	Résultats affichés :
Observation :	Observation :	Observation :

Affiche: 1 2 3 4 FIN

Cela affiche: 1 2 3 4 FIN

Cela affiche: 1 FIN

Observation: cela affiche la valeur de n Valeur de n dans l'intervalle 1; 5 Valeur de n dans l'intervalle 1;5 steps 2

Complétez la ligne d'instructions ci-dessous pour que n varie de 5 à 20 en exécutant des instructions toutes les 5 valeurs.

for n in range(..... , ,) :

↗
↑
↖

for n in range (5,20,5):

Séquence d'instructions 4	Séquence d'instructions 5	Séquence d'instructions 6
for n in range (1,5) : Res = Res+n print (Res) print ("FIN")	Res=0 for n in range (1,3) : Res = Res+n print (Res) print ("FIN")	for letter in "bien" : print(letter)
Résultats affichés :	Résultats affichés :	Résultats affichés :
Comparez ces 2 séquences d'instructions. Quelle conclusion en tirez-vous ?		Observation :

Affiche : **Erreur**

1 3 FIN

b i e n

Observation : Dans la première séquence la variable Res n'est pas définie. Valeur lettre dans "bien" affiche les lettres dans de"bien" dans l'ordre.

ACTIVITÉS SUR LES BOUCLES BORNÉES

Activité 1

Réaliser un programme, avec une boucle, permettant de répéter 100 fois la phrase « ceci est un test ».

Activité 1:

```
print("-----Ex n°2-----")
for loop in range(101):
    print("ceci est un test")
```

Activité 2

En utilisant une boucle, écrivez un programme qui affiche les 10 premiers multiples de 7.

Activité 3

Ecrivez un programme qui affiche les 16 premières puissances de 2 :

2^0
 2^1
 2^2
....

Activité 4

En utilisant une boucle, écrivez un programme pour afficher les 20 premiers multiples pairs de 7

Activité 5

Ecrivez un programme qui demande à l'utilisateur de taper un texte court et compte le nombre de fois où la lettre « e » apparaît.

Activité 6

Ecrivez un programme qui demande à l'utilisateur de taper un mot puis l'écrit dans une variable résultat en remplaçant chaque lettre « a » par un « x ». Affichez cette variable résultat.

Activité 7 : L'échiquier de Sissa

En Inde, un roi s'ennuie et demande qu'on lui invente un jeu pour le distraire. Le sage Sissa invente alors un jeu d'échecs. Pour le remercier, le roi lui demande de choisir sa récompense. Sissa demande alors au roi de prendre le plateau du jeu et de poser 1 grain de riz sur la première case, puis 2 grains sur la deuxième, 4 sur la troisième et ainsi de suite en doublant à chaque fois le nombre de grains de riz que l'on met sur chaque nouvelle case

Réalisez un programme comptant le nombre de grains de riz qu'il y aura sur un échiquier comportant 64 cases

Activité 2:

```
print("-----Ex n°3-----")
total = 0
nb = 0
```

```
while total < 10:
    nb += 1
    if nb%7 == 0:
        print(nb)
        total += 1
```

Activité 3:

```
print("-----Ex n°4-----")
total = 0
nb = 0
```

```
while total < 16:
    nb += 1
    if nb%2 == 0:
        print(2**total)
        total += 1
```

Activité 4:

```
print("-----Ex n°5-----")
total = 0
nb = 0
```

```
while total < 20:
    nb += 1
    if nb%7 == 0:
        if nb%2 == 0:
            print(nb)
            total += 1
```

Activité 5:

```
print("-----Ex n°6-----")
text = input("Ecrivez une phrase :")
total = 0
```

```
for n in text:
    if n == "e":
        total += 1
print("il y a", total, "e dans votre phrase")
```

Activité 6:

```
print("-----Ex n°7-----")
text = input("Ecrivez un mot :")
mot = ""
```

```
for n in text:
    if n == "a":
        mot += "x"
    else:
        mot += n
print(mot)
```

Activité 7:

```
print("-----Ex n°8-----")
total = 0
nb = 1
total1 = 0
```

```
for i in range(1,64):
    total1 = total
    nb *= 2
    total += nb
print(total)
```

Activité 8 : Culture informatique : La loi de Moore.



1) Effectuez une recherche mettant en évidence les informations suivantes :

- Qui est Gordon Moore ?
- Qu'appelle-t-on la loi de Moore ?
- Quand a-t-elle été formulée ?

2) On veut comparer les performances moyennes d'un processeur de 1970 et de 2018.

En notant P_1 la performance d'un ordinateur de 1970, proposez un programme utilisant une boucle qui donnera un résultat sous la forme :

« Un ordinateur de 2018 est ... fois plus puissant qu'un ordinateur de 1970 »

NB : en 1970 on considère que $P_1 = 1$

Activité 8:

1) Chimiste de formation, Gordon Earl Moore est devenu entrepreneur à l'âge de 39 ans lorsqu'il a fondé, en 1968, la société Intel, avec Andrew Grove et Robert Noyce. Les lois de Moore sont des lois empiriques qui ont trait à l'évolution de la puissance de calcul des ordinateurs et de la complexité du matériel informatique. En 1965, il énonce la très célèbre loi sur l'évolution des microprocesseurs, aujourd'hui connue sous le nom de Loi de Moore.

2) `print("-----Ex n°9-----")`

`p1 = 1000`

`p2 = 1000`

`total = 0`

`for n in range(1970,2018):`

`p2 *= 2000`

`total = p2/p1`

`print("Un ordinateur de 2018 est", total, "fois plus puissant qu'un ordinateur de 1970")`

Quelles sont les particularités des boucles NON bornées ?

Coder les séquences d'instructions suivantes et compléter les cases « résultats affichés » et « observations ».

Séquence d'instructions 1	Séquence d'instructions 2
<pre>n=0 while n<4: print(n) n+=1 print("FIN")</pre>	<pre>while n<4: print(n) n+=1 print("FIN")</pre>
Résultats affichés :	Résultats affichés :
Observation :	Observation :

Affiche : 1 2 3 FIN

Erreur

Observation : cela affiche la valeur de n

La variable n n'est pas définie

Sans coder les instructions, complétez le tableau ci-dessous en indiquant ce qui va se passer

Séquence d'instructions 3	Séquence d'instructions 4
<pre>n=1 while n>0: print(n) n+=1 print("FIN")</pre>	<pre>n=1 while n>2: print(n) n+=1 print("FIN")</pre>
Que va-t-il se passer ?	Que va-t-il se passer ?

Affiche : FIN

1 FIN

La boucle ne s'effectue pas car la condition n'est pas vérifiée il affiche la valeur de n jusqu'à ce que la condition ne soit plus vérifiée.

ACTIVITÉS SUR LES BOUCLES NON BORNÉES

Activité 1

En utilisant une boucle, écrivez un programme qui affiche les multiples de 7 inférieurs à 180.

Activité 2

On veut écrire un programme qui demande à l'utilisateur de choisir un nombre jusqu'à ce qu'il propose le nombre 14. On vous fournit l'algorithme. Complétez d'abord le programme dans le cadre prévu, puis codez le pour le tester.

Algorithme	Codage en Python
Saisir la valeur de a	a = int (input ("valeur choisie :"))
Tant que a et différent de 14 alors
Dire « Perdu »
Demander une nouvelle valeur pour a
Fin tant que
Afficher « Gagné »

Activité 1 :

```
print("-----Ex n°11-----")
n=1
```

```
while n<180:
    if n%7 == 0:
        print(n)
    n+=1
```

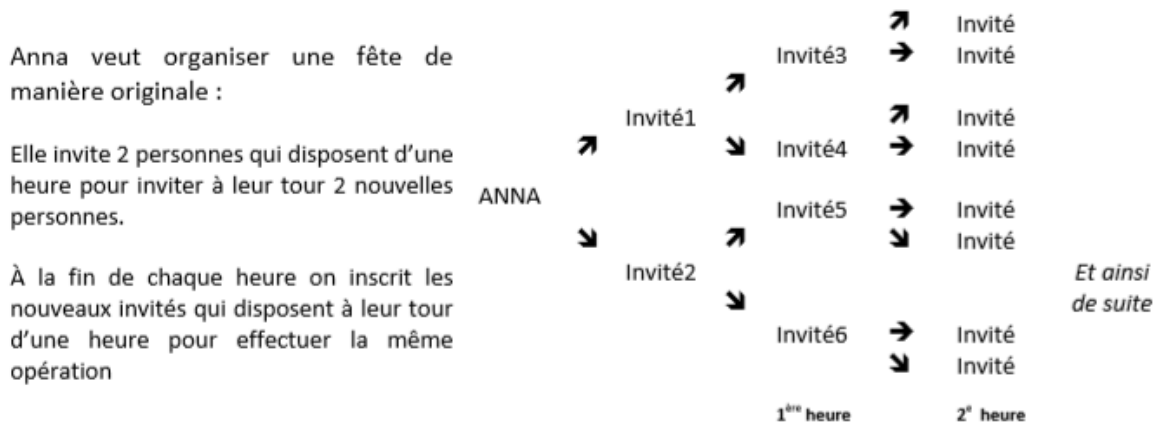
Activité 2:

```
print("-----Ex n°12-----")
a = int(input("valeur choisie :"))
```

```
while a != 14:
    print("Perdu")
    a = int(input("valeur choisie :"))
print("Gagné")
```

Activité 3 Invitations

NB : Vous enregistrerez les programmes des questions 1 et 2 dans 2 fichiers différents.



Questions :

- 1) Combien y aura-t-il de personnes invitées au bout de 8 heures.
- 2) Combien de temps faut-il pour remplir une salle de 1000 places ?

Activité 3:

- 1) Au bout de 8 heures il y a 1022 invités
- 2) Pour remplir toute la salle il faudra 8 heures.

LES CAS SPÉCIAUX : LES INSTRUCTIONS « BREAK » ET « CONTINUE »

Les instructions « break » et « continue » peuvent être insérées dans les boucles pour modifier leur exécution. Pour comprendre leur rôle codez les séquences suivantes et notez le résultat obtenu

Séquence 1	Séquence 2	Séquence 3
for letter in "bonjour" : print(letter)	for letter in "bonjour" : if letter=="j": break print(letter)	for letter in "bonjour" : if letter=="j": continue print(letter)
Résultat affiché :	Résultat affiché :	Résultat affiché :

Quel est le rôle de chaque instruction ?

BREAK :

CONTINUE :

Attention : ces deux instructions sont employées avec une instruction conditionnelle (IF)

Affiche : b o n j o u r	b o n	b o n o u r
Observation : cela affiche la valeur de n quand n	cela affiche la valeur de n jusqu'à ce que n soit égal à j	cela affiche la valeur de n à pars est égal à j

BREAK : permet de forcer l'arrêt de la boucle

CONTINUE : permet de sauter toutes les actions et reprendre au début de la boucle

LES CAS SPÉCIAUX : LES INSTRUCTIONS « WHILE » ET « DO... WHILE »

Codez les séquences suivantes et comparez les résultats

Séquence 1 While		Séquence 2 Do ... While....	
Codage	Résultat affiché :	Codage	Résultat affiché :
<pre>i=0 while i<6: print (i) i=i+1 print ("fin")</pre>		<pre>i=0 while True: if i<6 : print (i) i=i+1 else : break print ("fin")</pre>	

Selon les langages de programmation une boucle non bornée peut être codée différemment. Il faut alors introduire un test dans la boucle et répéter tant que la condition est respectée (exprimée sous forme de valeur booléenne) ou introduire une interruption de la boucle. Avec Python cette syntaxe est possible mais elle ne se justifie pas.

Activité 4 :

Ecrivez un programme qui demande à l'utilisateur de taper un mot puis l'écrit dans une variable résultat en supprimant les lettres « a » et en arrêtant le traitement si la lettre 'g' apparaît. Affichez la variable résultat.

Séquence :

Affiche : 0 1 2 3 4 5 fin

0 1 2 3 4 5 fin

Activité 4:

```
print("-----Ex n°16-----")
text = input("ecrivez un mot")
resultat = ""
```

```
for n in text:
    if n == "a":
        continue
    if n == "g":
        break
    resultat += n
print(resultat)
```