

# INF1 : Algorithmique et Programmation

## Cours 3 : Structures itératives

Domitile Lourdeaux

Université de technologie de Compiègne

Printemps 2024



- ① Motivations
- ② Répétitions
- ③ Boucles à Bornes définies
- ④ Boucles à Bornes indéfinies
- ⑤ Exercices

## 1 Motivations

## 2 Répétitions

## 3 Boucles à Bornes définies

## 4 Boucles à Bornes indéfinies

## 5 Exercices

## Exemple

« Boucle »

Tant que je comprends INF1 faire

je joue à Minecraft

je retravaille INF1

① Motivations

② Répétitions

③ Boucles à Bornes définies

④ Boucles à Bornes indéfinies

⑤ Exercices

# Différents types de boucles

## Boucles à bornes définies

---

### Algorithm 1 Boucle Pour

---

```
pour  $i \leftarrow n$  to  $m$  faire  
|  $\langle instruction(s) \rangle$   
fin
```

---

---

### Algorithm 2 Boucle Pour chaque

---

```
pour chaque  $element \in ensemble$   
faire  
|  $\langle instruction(s) \rangle$   
fin
```

---

## Boucles à bornes indéfinies

---

### Algorithm 3 Boucle Tant que

---

```
tant que  $\langle condition(s) \rangle$  faire  
|  $\langle instruction(s) \rangle$   
fin
```

---

---

### Algorithm 4 Boucle Répéter...

---

jusqu'à

---

```
répéter  
|  $\langle instruction(s) \rangle$   
jusqu'à  $\langle condition(s) \rangle$ 
```

---



## ① Motivations

## ② Répétitions

③ Boucles à Bornes définies  
Boucle Pour  
Boucle Pour Chaque

## ④ Boucles à Bornes indéfinies

## ⑤ Exercices



## « Boucle Pour » : Exemple

```
Bonjour étudiant n° 1  
Bonjour étudiant n° 2  
Bonjour étudiant n° 3  
Bonjour étudiant n° 4  
Bonjour étudiant n° 5
```

```
# Programme Bonjour les étudiants  
print("-----Essai 1 -----")  
print("Bonjour Etudiant n°1")  
print("Bonjour Etudiant n°2")  
print("Bonjour Etudiant n°3")  
print("Bonjour Etudiant n°4")  
print("Bonjour Etudiant n°5")
```

# « Boucle Pour » : Structure générale (1)

## Algorithmes

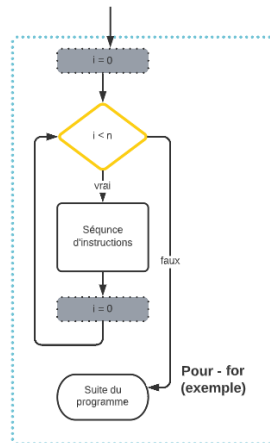
### Algorithm 5 Boucle Pour

```
pour  $i \leftarrow 0$  to  $n$  faire  
|  $\langle \text{instruction}(s) \rangle$   
fin
```

### Algorithm 6 Boucle Pour

```
pour  $i \leftarrow n$  to  $m$  faire  
|  $\langle \text{instruction}(s) \rangle$   
fin
```

## Organigramme



## « Boucle Pour » : Structure générale (2)

### Python

```
for i in range(n): # i prend alors les valeurs de 0 à n-1  
    < bloc d'instructions >  
  
for i in range(n, m): # i prend les valeurs de n à m-1  
    < bloc d'instructions >
```

### Remarques

- $i$  sert de compteur
- La boucle n'est exécutée que si  $n < m$

## « Boucle Pour » : Fonctionnement (1)

### Exemple 1 « Bonjour »- Algorithme

---

**Algorithm 7** Bonjour les étudiants 1 à 5

---

```
pour  $i \leftarrow 1$  to 5 faire  
| Afficher ("Bonjour étudiant n° ",  $i$ )  
fin
```

---

Voir en ligne...

# « Boucle Pour » : Fonctionnement (2)

## Exemple 1 « Bonjour »- Python et fonctionnement

```
# Programme Bonjour les étudiants
print("-----Essai 1 -----")
print("Bonjour Etudiant n°1")
print("Bonjour Etudiant n°2")
print("Bonjour Etudiant n°3")
print("Bonjour Etudiant n°4")
print("Bonjour Etudiant n°5")

print("-----Essai 2 -----")
for i in range(1,5):
    print("Bonjour étudiant n°", i)

print("-----Essai 3 -----")
for i in range(1,6):
    print("Bonjour étudiant n°", i)

print("-----Essai 4 -----")
for i in range(5):
    print("Bonjour étudiant n°", i)

print("-----Essai 5 -----")
for i in range(5):
    print("Bonjour étudiant n°", i+1)
```

```
-----Essai 1 -----
Bonjour Etudiant n°1
Bonjour Etudiant n°2
Bonjour Etudiant n°3
Bonjour Etudiant n°4
Bonjour Etudiant n°5

-----Essai 2 -----
Bonjour étudiant n° 1
Bonjour étudiant n° 2
Bonjour étudiant n° 3
Bonjour étudiant n° 4

-----Essai 3 -----
Bonjour étudiant n° 1
Bonjour étudiant n° 2
Bonjour étudiant n° 3
Bonjour étudiant n° 4
Bonjour étudiant n° 5

-----Essai 4 -----
Bonjour étudiant n° 0
Bonjour étudiant n° 1
Bonjour étudiant n° 2
Bonjour étudiant n° 3
Bonjour étudiant n° 4

-----Essai 5 -----
Bonjour étudiant n° 1
Bonjour étudiant n° 2
Bonjour étudiant n° 3
Bonjour étudiant n° 4
Bonjour étudiant n° 5
```

2. itération	i
--------------	---

1	1
2	2
3	3
4	4

3. itération	i
--------------	---

1	1
2	2
3	3
4	4
5	5

4. itération	i
--------------	---

1	0
2	1
3	2
4	3
5	4

5. it.	i	i + 1
--------	---	-------

1	0	1
2	1	2
3	2	3
4	3	4
5	4	5

## « Boucle Pour » : Fonctionnement (3)

### Exemple 2 « Somme de carrés »- Algorithme

---

**Algorithm 8** Somme des  $n$  premiers carrés

---

Afficher ("Nombre d'entiers")

Lire ( $nEntier$ )

$s \leftarrow 0$

**pour**  $i \leftarrow 1$  to  $nEntiers$  **faire**

$s \leftarrow s + i * i$

**fin**

Afficher ("Somme de ces premiers entiers au carré : ",  $s$ )

---

## « Boucle Pour » : Fonctionnement (4)

### Exemple 3 « Somme de carrés »- Python et fonctionnement

$s = 0$

```
# Somme des n premiers entiers au carré
n = int(input("Nombre d'entiers : "))
s = 0

for i in range(1,n+1):
    s = s + i**2

print("Somme de ces premiers entiers au carré :", s)
```

itération	i	s
1	1	1
2	2	5
3	3	14
4	4	30
5	5	55

Nombre d'entiers : 5

Somme de ces premiers entiers au carré : 55

## « Boucle Pour » : Fonctionnement (5)

### Exemple 3 « Somme de carrés »- Python et fonctionnement

#### Attention

#### A l'indentation

```
# Somme des n premiers entiers au carré
n = int(input("Nombre d'entiers : "))
s = 0

for i in range(1,n+1):
    s = s + i**2

print("Somme de ces premiers entiers au carré :", s)
```

```
Nombre d'entiers : 5
Somme de ces premiers entiers au carré : 55
```

```
for i in range(1,n+1):
    s = s + i**2
    print("Somme de ces premiers entiers au carré :", s)
```

```
Nombre d'entiers : 5
Somme de ces premiers entiers au carré : 1
Somme de ces premiers entiers au carré : 5
Somme de ces premiers entiers au carré : 14
Somme de ces premiers entiers au carré : 30
Somme de ces premiers entiers au carré : 55
```



## « Boucle Pour » : Fonctionnement (6)

### Exemple 4 « Inverse »- Algorithmme

---

#### Algorithm 9 Boucle des Carrés dans l'ordre inverse

---

```
pour  $i \leftarrow 5$  to 1 faire  
|   Afficher ("Le carré de ", i, " = ", i*i)  
fin
```

---

## « Boucle Pour » : Fonctionnement (7)

### Exemple 4 « Inverse »- Python et fonctionnement

```
# Boucle carrés dans l'ordre inverse
for i in range(5, 0, -1):
    print(f"Le carré de {i} est {i**2}")
```

```
Le carré de 5 est 25
Le carré de 4 est 16
Le carré de 3 est 9
Le carré de 2 est 4
Le carré de 1 est 1
```

itération	i	$i^2$
1	5	25
2	4	16
3	3	9
4	2	4
5	1	1

## « Boucle Pour » : Fonctionnement (8)

### Exemple 4 « Double boucle »- Algorithme

---

#### Algorithm 10 Boucle des Carrés dans l'ordre inverse

---

```
// Affichage de la matrice
1 pour i allant de 1 à 5 faire
    // Affichage d'une ligne
2    pour j allant de 1 à 5 faire
3        si i == j alors
4            // Affichage d'une case
5            Afficher ("1", end = " ")
6        sinon
7            Afficher ("0", end = " ")
8    Afficher ()
```

## « Boucle Pour » : Fonctionnement (9)

### Exemple 5 « Double boucle »- Python et fonctionnement

```
# Affichage d'une matrice
for i in range(1, 5):
    # Affichage d'une ligne
    for j in range(1, 5):
        # Affichage d'une case
        if (i == j):
            print("1", end=" ")
        else:
            print("0", end=" ")
    print()
```

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

itération	i	j
1	1	1
2		2
3		3
4		4
5	2	1
6		2
7		3
8		4
9	3	1
10		2
11		3
12		4
13	4	1
14		2
15		3
16		4

## ① Motivations

## ② Répétitions

## ③ Boucles à Bornes définies

Boucle Pour

Boucle Pour Chaque

## ④ Boucles à Bornes indéfinies

## ⑤ Exercices

## « Boucle Pour chaque » : Exemple

Liste des inscrits INF1 P23 = ["Alice", "Bob", "Anna", "Toto"]

```
-----Essai 5 -----  
Bonjour  Alice  
Bonjour  Bob  
Bonjour  Anna  
Bonjour  Toto
```

Liste des inscrits INF1 P23 = ["Alice", "Bob", "Anna", "Marcel",  
"Toto"]

# « Boucle Pour chaque » : Structure générale

## Algorithme

---

**Algorithm 11**    Boucle Pour  
chaque

---

*pour chaque*  $element \in ensemble$   
*faire*  
     $< instruction(s) >$   
*fin*

---

## Python

```
for element in ensemble:  
    < instruction(s) >
```

## « Boucle Pour chaque » : Fonctionnement (1)

### Exemple 1 « Envoyer des mails à un ensemble d'étudiants »

```
print("-----Essai 5 -----")
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    print("Bonjour ", nom)
```

```
-----Essai 5 -----
Bonjour Alice
Bonjour Bob
Bonjour Anna
Bonjour Toto
```



## « Boucle Pour chaque » : Fonctionnement (2)

Exemple 1 « Envoyer des mails à un ensemble d'étudiants n'ayant pas abandonné »

```
print("-----Essai 6 -----")
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
inf1_P23_abandon = ["Bob", "Anna"]
for nom in inf1_P23:
    if nom in inf1_P23_abandon:
        print("Ne pas envoyer mail à ", nom)
    else:
        print("Envoyer mail à ", nom)
```

```
-----Essai 6 -----
Envoyer mail à  Alice
Ne pas envoyer mail à  Bob
Envoyer mail à  Bob
Ne pas envoyer mail à  Anna
Envoyer mail à  Anna
Envoyer mail à  Toto
```

## « Boucle Pour chaque » : Fonctionnement (3)

Exemple 1 « Envoyer des mails à un ensemble d'étudiants n'ayant pas abandonné »

```
print("-----Essai 7 -----")
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
inf1_P23_abandon = ["Bob", "Anna"]
for nom in inf1_P23:
    if nom in inf1_P23_abandon:
        print("Ne pas envoyer mail à ", nom)
        continue
    print("Envoyer mail à ", nom)
```

```
-----Essai 7 -----
Envoyer mail à Alice
Ne pas envoyer mail à Bob
Ne pas envoyer mail à Anna
Envoyer mail à Toto
```

### Remarque

- **continue** permet de passer au tour de boucle suivant

## « Boucle Pour chaque » : Fonctionnement (3)

Exemple 1 « Envoyer des mails à un ensemble d'étudiants n'ayant pas abandonné »

```
print("-----Essai 7 -----")
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
inf1_P23_abandon = ["Bob", "Anna"]
for nom in inf1_P23:
    if nom in inf1_P23_abandon:
        print("Ne pas envoyer mail à ", nom)
        continue
    print("Envoyer mail à ", nom)
```

```
-----Essai 7 -----
Envoyer mail à Alice
Ne pas envoyer mail à Bob
Ne pas envoyer mail à Anna
Envoyer mail à Toto
```

### Remarque

- **continue** permet de passer au tour de boucle suivant
- On pourrait utiliser un **else** aussi

## « Boucle Pour chaque » : Fonctionnement (4)

### Exemple 2 « Envoyer des mails jusqu'à "Anna" »

- Qu'en pense ChatGPT ?



Ecris un programme en Python qui dit bonjour à tous les étudiants d'un ensemble



Voici un exemple de programme en Python qui dit bonjour à tous les étudiants d'un ensemble :



```
python Copy code

students = {'Alice', 'Bob', 'Charlie', 'Dave', 'Eve'} # ensemble d'étudiants

for student in students:
    print(f"Bonjour {student} !")
```

Dans cet exemple, nous créons un ensemble d'étudiants appelé `students` contenant les noms d'étudiants. Ensuite, nous utilisons une boucle `for` pour parcourir chaque élément de l'ensemble `students`. À chaque itération de la boucle, nous utilisons la fonction `print` pour afficher une chaîne de caractères personnalisée contenant le nom de l'étudiant.

Le résultat de l'exécution du programme sera une série de messages de salutation, un pour chaque étudiant de l'ensemble `students`.



modifies ce programme pour s'arrêter à Dave

# « Boucle Pour chaque » : Fonctionnement (5)

## Exemple 2 « Envoyer des mails jusqu'à "Anna" »

- Qu'en pense ChatGPT ?

 Voici un exemple de programme en Python qui dit bonjour à tous les étudiants d'un ensemble :

```
python Copy code
students = {'Alice', 'Bob', 'Charlie', 'Dave', 'Eve'} # ensemble d'étudiants

for student in students:
    print(f"Bonjour {student} !")
```

Dans cet exemple, nous créons un ensemble d'étudiants appelé `students` contenant les noms d'étudiants. Ensuite, nous utilisons une boucle `for` pour parcourir chaque élément de l'ensemble `students`. À chaque itération de la boucle, nous utilisons la fonction `print` pour afficher une chaîne de caractères personnalisée contenant le nom de l'étudiant.

Le résultat de l'exécution du programme sera une série de messages de salutation, un pour chaque étudiant de l'ensemble `students`.

## « Boucle Pour chaque » : Fonctionnement (6)

### Exemple 2 « Envoyer des mails jusqu'à "Anna" »

```
print("-----Essai 8 -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom != "Anna":
        print("Envoyer mail à ", nom)
print("-----Essai 8b -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
b = True
for nom in inf1_P23:
    if nom == "Anna":
        b = False
    elif nom != "Anna" and b == True:
        print("Envoyer mail à ", nom)
```

```
print("-----Essai 9 -----")
# Envoyer un mail jusqu'à Anna (nom compris)
# inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom == "Anna":
        break
    print("Envoyer mail à ", nom)
```

#### Remarques

- Essai 8 ne fonctionne pas

## « Boucle Pour chaque » : Fonctionnement (6)

### Exemple 2 « Envoyer des mails jusqu'à "Anna" »

```
print("-----Essai 8 -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom != "Anna":
        print("Envoyer mail à ", nom)
print("-----Essai 8b -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
b = True
for nom in inf1_P23:
    if nom == "Anna":
        b = False
    elif nom != "Anna" and b == True:
        print("Envoyer mail à ", nom)
```

```
print("-----Essai 9 -----")
# Envoyer un mail jusqu'à Anna (nom compris)
# inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom == "Anna":
        break
    print("Envoyer mail à ", nom)
```

#### Remarques

- Essai 8 ne fonctionne pas
- 8b est non-optimal (imaginez 10 000 étudiants, Anna est la 1<sup>re</sup> de la liste)

## « Boucle Pour chaque » : Fonctionnement (6)

### Exemple 2 « Envoyer des mails jusqu'à "Anna" »

```
print("-----Essai 8 -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom != "Anna":
        print("Envoyer mail à ", nom)
print("-----Essai 8b -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
b = True
for nom in inf1_P23:
    if nom == "Anna":
        b = False
    elif nom != "Anna" and b == True:
        print("Envoyer mail à ", nom)
```

```
print("-----Essai 9 -----")
# Envoyer un mail jusqu'à Anna (nom compris)
# inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom == "Anna":
        break
    print("Envoyer mail à ", nom)
```

#### Remarques

- Essai 8 ne fonctionne pas
- 8b est non-optimal (imaginez 10 000 étudiants, Anna est la 1<sup>re</sup> de la liste)
- **break** permet de casser la boucle



## « Boucle Pour chaque » : Fonctionnement (6)

### Exemple 2 « Envoyer des mails jusqu'à "Anna" »

```
print("-----Essai 8 -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom != "Anna":
        print("Envoyer mail à ", nom)
print("-----Essai 8b -----")
# Envoyer un mail jusqu'à Anna
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
b = True
for nom in inf1_P23:
    if nom == "Anna":
        b = False
    elif nom != "Anna" and b == True:
        print("Envoyer mail à ", nom)
```

```
print("-----Essai 9 -----")
# Envoyer un mail jusqu'à Anna (nom compris)
# inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
for nom in inf1_P23:
    if nom == "Anna":
        break
    print("Envoyer mail à ", nom)
```

#### Remarques

- Essai 8 ne fonctionne pas
- 8b est non-optimal (imaginez 10 000 étudiants, Anna est la 1<sup>re</sup> de la liste)
- **break** permet de casser la boucle
- Essai 8b et 9 sont à éviter / proscrire car elles sont peu intelligibles

## « Boucle Pour chaque » : Fonctionnement (7)

### Exemple 2 « Envoyer des mails jusqu'à "Anna" »

```
print("-----Essai 10 -----")
# Envoyer un mail à Anna (nom compris)
inf1_P23 = ["Alice", "Bob", "Anna", "Toto"]
index = 0
while index < len(inf1_P23) and inf1_P23[index] != "Anna":
    print("Envoyer mail à ", inf1_P23[index])
    index += 1
```

## ① Motivations

## ② Répétitions

## ③ Boucles à Bornes définies

## ④ Boucles à Bornes indéfinies

Boucle Tant que  
Boucle Répéter

## ⑤ Exercices

1 Motivations

2 Répétitions

3 Boucles à Bornes définies

4 Boucles à Bornes indéfinies

Boucle Tant que  
Boucle Répéter

5 Exercices

# « Boucle Tant que » : Structure générale

## Algorithme

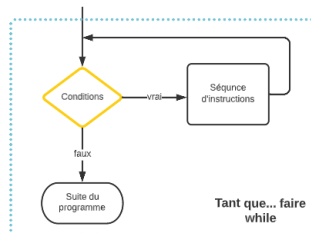
### Algorithm 12 Boucle Tant que

```
tant que < condition(s) > faire  
| < instruction(s) >  
fin
```

### Quand l'utiliser ?

- Si on n'est pas certain d'entrer dans la boucle
- La condition doit être initialisée au départ

## Organigramme



## Python

```
while <conditions> :  
    <bloc d'instructions>
```

## ① Motivations

## ② Répétitions

## ③ Boucles à Bornes définies

## ④ Boucles à Bornes indéfinies

Boucle Tant que  
Boucle Répéter

## ⑤ Exercices

# « Boucle Répéter » : Structure générale (1)

## Algorithme

**Algorithm 13** Boucle Répéter...  
jusqu'à

répéter

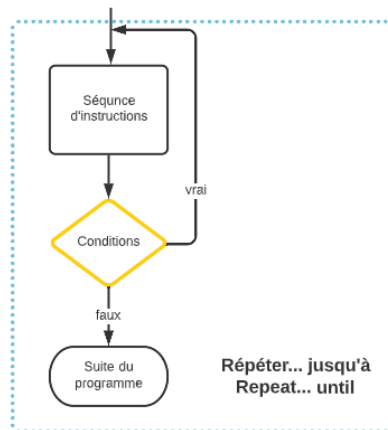
|  $\langle \text{instruction}(s) \rangle$

jusqu'à  $\langle \text{condition}(s) \rangle$

### Quand l'utiliser ?

- On est certain d'entrer dans la boucle
- La condition peut être initialisée dans la boucle

## Organigramme



## « Boucle Répéter » : Structure générale (2)

### Remarques

- La boucle Repeat n'existe pas en Python
- On peut la simuler en utilisant le while et en inversant la condition



## « Boucle Tant que » : Fonctionnement (1)

### Exemple 1 « Saisir un nombre positif »- Algorithme

---

#### Algorithm 14 Saisie d'un nombre positif - « Boucle Répéter »

---

répéter

    Afficher ("Saisir un nombre positif : ")

    Lire ( $n$ )

jusqu'à  $n \geq 0$

---

---

#### Algorithm 15 Saisie d'un nombre positif - « Boucle Tant que »

---

Afficher ("Saisir un nombre positif : ")

Lire ( $n$ )

tant que  $n < 0$  faire

    Afficher ("Ce nombre est négatif, saisir un nombre positif : ")

    Lire ( $n$ )

fin

---

## « Boucle Tant que » : Fonctionnement (2)

### Exemple 1 « Saisir un nombre positif »- Python et fonctionnement

```
n = int(input('Saisir un nombre positif : '))  
while n < 0:  
    n = int(input('Ce nombre est négatif, merci de saisir un nombre positif : '))
```

```
Saisir un nombre positif : -10  
Ce nombre est négatif, merci de saisir un nombre positif : -5  
Ce nombre est négatif, merci de saisir un nombre positif : -3  
Ce nombre est négatif, merci de saisir un nombre positif : 10
```

## « Boucle Tant que » : Fonctionnement (3)

### Exemple 2 « Augmentation de salaire »

- Le salaire augmente de 200 €, tous les ans jusqu'à 2000 €
- Le salaire augmente de 100 €, tous les ans jusqu'à 3000 €
- Puis le salaire augmente de 50 €, sans limite (enfin jusqu'à 62, 64, 80 ans?)
- Quel salaire obtient-on au bout de 20 ans?

Voir en ligne...

## « Boucle Tant que » : Fonctionnement (4)

### Exemple 2 « Augmentation de salaire »

```
# Salaire qui augmente de 200 € jusqu'à 2000 €  
# - Version boucle infinie  
salaire_courant = int(input("Quel est le salaire de début : "))  
annee = 0  
while salaire_courant < 2000:~  
    annee += 1  
    print(f'Au bout de {annee} an(s), votre salaire est de : {salaire_courant}')
```

#### Remarque

- Quel est le problème ?

## « Boucle Tant que » : Fonctionnement (5)

### Exemple 2 « Augmentation de salaire »

```
# Salaire qui augmente de 200 € jusqu'à 2000 €  
# - Version boucle infinie  
salaire_courant = int(input("Quel est le salaire de début : "))  
annee = 0  
while salaire_courant < 2000 :  
    annee += 1  
    print(f'Au bout de {annee} an(s), votre salaire est de : {salaire_courant}')
```

```
Au bout de 100040 an(s), votre salaire est de : 1000  
Au bout de 100041 an(s), votre salaire est de : 1000  
Au bout de 100042 an(s), votre salaire est de : 1000  
Au bout de 100043 an(s), votre salaire est de : 1000  
Au bout de 100044 an(s), votre salaire est de : 1000  
Au bout de 100045 an(s), votre salaire est de : 1000  
Au bout de 100046 an(s), votre salaire est de : 1000  
Au bout de 100047 an(s), votre salaire est de : 1000  
Au bout de 100048 an(s), votre salaire est de : 1000  
Au bout de 100049 an(s), votre salaire est de : 1000  
Au bout de 100050 an(s), votre salaire est de : 1000  
Au bout de 100051 an(s), votre salaire est de : 1000  
Au bout de 100052 an(s), votre salaire est de : 1000
```

### Remarque

- Bien penser la condition d'arrêt pour éviter une boucle infinie

# « Boucle Tant que » : Fonctionnement (6)

## Exemple 2 « Augmentation de salaire »

```
# Salaire qui augmente de 200 € jusqu'à 2000 €
salaire_courant = int(input("Quel est le salaire de début : "))
age = int(input("Quel est l'âge d'embauche : "))
annee = 0

while salaire_courant < 2000:
    salaire_courant += 200
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 100 € jusqu'à 3000 €
while salaire_courant < 3000:
    salaire_courant += 100
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 50 € jusqu'à la retraite au delà de 3000 €
while age < 62:
    salaire_courant += 50
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')
```

```
Au bout de 1 an(s) à 21 ans, votre salaire est de : 1200
Au bout de 2 an(s) à 22 ans, votre salaire est de : 1400
Au bout de 3 an(s) à 23 ans, votre salaire est de : 1600
Au bout de 4 an(s) à 24 ans, votre salaire est de : 1800
Au bout de 5 an(s) à 25 ans, votre salaire est de : 2000
Au bout de 6 an(s) à 26 ans, votre salaire est de : 2100
Au bout de 7 an(s) à 27 ans, votre salaire est de : 2200
Au bout de 8 an(s) à 28 ans, votre salaire est de : 2300
Au bout de 9 an(s) à 29 ans, votre salaire est de : 2400
Au bout de 10 an(s) à 30 ans, votre salaire est de : 2500
Au bout de 11 an(s) à 31 ans, votre salaire est de : 2600
Au bout de 12 an(s) à 32 ans, votre salaire est de : 2700
Au bout de 13 an(s) à 33 ans, votre salaire est de : 2800
Au bout de 14 an(s) à 34 ans, votre salaire est de : 2900
Au bout de 15 an(s) à 35 ans, votre salaire est de : 3000
Au bout de 16 an(s) à 36 ans, votre salaire est de : 3100
Au bout de 17 an(s) à 37 ans, votre salaire est de : 3200
Au bout de 18 an(s) à 38 ans, votre salaire est de : 3300
Au bout de 19 an(s) à 39 ans, votre salaire est de : 3400
Au bout de 20 an(s) à 40 ans, votre salaire est de : 3500
Au bout de 21 an(s) à 41 ans, votre salaire est de : 3550
Au bout de 22 an(s) à 42 ans, votre salaire est de : 3600
Au bout de 23 an(s) à 43 ans, votre salaire est de : 3650
Au bout de 24 an(s) à 44 ans, votre salaire est de : 3700
Au bout de 25 an(s) à 45 ans, votre salaire est de : 3750
Au bout de 26 an(s) à 46 ans, votre salaire est de : 3800
Au bout de 27 an(s) à 47 ans, votre salaire est de : 3850
Au bout de 28 an(s) à 48 ans, votre salaire est de : 3900
Au bout de 29 an(s) à 49 ans, votre salaire est de : 3950
Au bout de 30 an(s) à 50 ans, votre salaire est de : 4000
Au bout de 31 an(s) à 51 ans, votre salaire est de : 4050
Au bout de 32 an(s) à 52 ans, votre salaire est de : 4100
Au bout de 33 an(s) à 53 ans, votre salaire est de : 4150
Au bout de 34 an(s) à 54 ans, votre salaire est de : 4200
Au bout de 35 an(s) à 55 ans, votre salaire est de : 4250
Au bout de 36 an(s) à 56 ans, votre salaire est de : 4300
Au bout de 37 an(s) à 57 ans, votre salaire est de : 4350
Au bout de 38 an(s) à 58 ans, votre salaire est de : 4400
Au bout de 39 an(s) à 59 ans, votre salaire est de : 4450
Au bout de 40 an(s) à 60 ans, votre salaire est de : 4500
Au bout de 41 an(s) à 61 ans, votre salaire est de : 4550
Au bout de 42 an(s) à 62 ans, votre salaire est de : 4600
```

## Remarques

- Quel est le problème ?

# « Boucle Tant que » : Fonctionnement (6)

## Exemple 2 « Augmentation de salaire »

```
# Salaire qui augmente de 200 € jusqu'à 2000 €
salaire_courant = int(input("Quel est le salaire de début : "))
age = int(input("Quel est l'âge d'embauche : "))
annee = 0
while salaire_courant < 2000:
    salaire_courant += 200
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 100 € jusqu'à 3000 €
while salaire_courant < 3000:
    salaire_courant += 100
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 50 € jusqu'à la retraite au delà de 3000 €
while age < 62:
    salaire_courant += 50
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')
```

```
Au bout de 1 an(s) à 21 ans, votre salaire est de : 1200
Au bout de 2 an(s) à 22 ans, votre salaire est de : 1400
Au bout de 3 an(s) à 23 ans, votre salaire est de : 1600
Au bout de 4 an(s) à 24 ans, votre salaire est de : 1800
Au bout de 5 an(s) à 25 ans, votre salaire est de : 2000
Au bout de 6 an(s) à 26 ans, votre salaire est de : 2100
Au bout de 7 an(s) à 27 ans, votre salaire est de : 2200
Au bout de 8 an(s) à 28 ans, votre salaire est de : 2300
Au bout de 9 an(s) à 29 ans, votre salaire est de : 2400
Au bout de 10 an(s) à 30 ans, votre salaire est de : 2500
Au bout de 11 an(s) à 31 ans, votre salaire est de : 2600
Au bout de 12 an(s) à 32 ans, votre salaire est de : 2700
Au bout de 13 an(s) à 33 ans, votre salaire est de : 2800
Au bout de 14 an(s) à 34 ans, votre salaire est de : 2900
Au bout de 15 an(s) à 35 ans, votre salaire est de : 3000
Au bout de 16 an(s) à 36 ans, votre salaire est de : 3100
Au bout de 17 an(s) à 37 ans, votre salaire est de : 3200
Au bout de 18 an(s) à 38 ans, votre salaire est de : 3300
Au bout de 19 an(s) à 39 ans, votre salaire est de : 3400
Au bout de 20 an(s) à 40 ans, votre salaire est de : 3500
Au bout de 21 an(s) à 41 ans, votre salaire est de : 3550
Au bout de 22 an(s) à 42 ans, votre salaire est de : 3600
Au bout de 23 an(s) à 43 ans, votre salaire est de : 3650
Au bout de 24 an(s) à 44 ans, votre salaire est de : 3700
Au bout de 25 an(s) à 45 ans, votre salaire est de : 3750
Au bout de 26 an(s) à 46 ans, votre salaire est de : 3800
Au bout de 27 an(s) à 47 ans, votre salaire est de : 3850
Au bout de 28 an(s) à 48 ans, votre salaire est de : 3900
Au bout de 29 an(s) à 49 ans, votre salaire est de : 3950
Au bout de 30 an(s) à 50 ans, votre salaire est de : 4000
Au bout de 31 an(s) à 51 ans, votre salaire est de : 4050
Au bout de 32 an(s) à 52 ans, votre salaire est de : 4100
Au bout de 33 an(s) à 53 ans, votre salaire est de : 4150
Au bout de 34 an(s) à 54 ans, votre salaire est de : 4200
Au bout de 35 an(s) à 55 ans, votre salaire est de : 4250
Au bout de 36 an(s) à 56 ans, votre salaire est de : 4300
Au bout de 37 an(s) à 57 ans, votre salaire est de : 4350
Au bout de 38 an(s) à 58 ans, votre salaire est de : 4400
Au bout de 39 an(s) à 59 ans, votre salaire est de : 4450
Au bout de 40 an(s) à 60 ans, votre salaire est de : 4500
Au bout de 41 an(s) à 61 ans, votre salaire est de : 4550
Au bout de 42 an(s) à 62 ans, votre salaire est de : 4600
```

## Remarques

- Quel est le problème ?
- Penser à tous les cas

# « Boucle Tant que » : Fonctionnement (7)

## Exemple 2 « Augmentation de salaire »

```
# Salaire qui augmente de 200 € jusqu'à 2000 €
salaire_courant = int(input("Quel est le salaire de début : "))
age = int(input("Quel est l'âge d'embauche : "))
annee = 0
while salaire_courant < 2000:
    salaire_courant += 200
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 100 € jusqu'à 3000 €
while salaire_courant < 3000:
    salaire_courant += 100
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 50 € jusqu'à la retraire au dela de 3000 €
while age < 62_:
    salaire_courant += 50
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')
```

```
Quel est le salaire de début : 2000
Quel est l'âge d'embauche : 40
Au bout de 1 an(s) à 61 ans, votre salaire est de : 2100
Au bout de 2 an(s) à 62 ans, votre salaire est de : 2200
Au bout de 3 an(s) à 63 ans, votre salaire est de : 2300
Au bout de 4 an(s) à 64 ans, votre salaire est de : 2400
Au bout de 5 an(s) à 65 ans, votre salaire est de : 2500
Au bout de 6 an(s) à 66 ans, votre salaire est de : 2600
Au bout de 7 an(s) à 67 ans, votre salaire est de : 2700
Au bout de 8 an(s) à 68 ans, votre salaire est de : 2800
Au bout de 9 an(s) à 69 ans, votre salaire est de : 2900
Au bout de 10 an(s) à 70 ans, votre salaire est de : 3000
```

## Remarque

PB = Si démarre plus tard



# « Boucle Tant que » : Fonctionnement (8)

## Exemple 2 « Augmentation de salaire »

```
# Salaire qui augmente de 200 € jusqu'à 2000 € et si age retraite non atteint
AGE_RETRAITE = 62
salaire_courant = int(input("Quel est le salaire de début : "))
age = int(input("Quel est l'âge d'embauche : "))
annee = 0
while salaire_courant < 2000 and age < AGE_RETRAITE_:
    salaire_courant += 200
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 100 € jusqu'à 3000 € et si age retraite non atteint
while salaire_courant < 3000 and age < AGE_RETRAITE_:
    salaire_courant += 100
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')

# Salaire qui augmente de 50 € jusqu'à la retraite au delà de 3000 € et si age retraite non atteint
while age < AGE_RETRAITE_:
    salaire_courant += 50
    annee += 1
    age += 1
    print(f'Au bout de {annee} an(s) à {age} ans, votre salaire est de : {salaire_courant}')
```

```
Quel est le salaire de début : 2000
Quel est l'âge d'embauche : 60
Au bout de 1 an(s) à 61 ans, votre salaire est de : 2100
Au bout de 2 an(s) à 62 ans, votre salaire est de : 2200
```

## « Boucle Tant que » : A éviter

### Boucles infinies et break

```
def copieFichier(source, destination):  
    "copie intégrale d'un fichier"  
    fs = open(source, 'r')  
    fd = open(destination, 'w')  
    while 1:  
        txt = fs.read(50)  
        if txt == "":  
            break  
        fd.write(txt)  
    fs.close()  
    fd.close()  
    return
```

```
while <condition 1> :  
    --- instructions diverses ---  
    if <condition 2> :  
        break  
    --- instructions diverses ---  
    if <condition 3>:  
        break  
etc.
```

① Motivations

② Répétitions

③ Boucles à Bornes définies

④ Boucles à Bornes indéfinies

⑤ Exercices

# A vous de jouer (1)

## Cherchez les erreurs

How to create programming exercises with automated feedback?



**Correctness:** What could be assessed with functional tests?

Here are some aspects you might validate:

- **Output**
  - Highlighting of discrepancies? Which line(s) has/have errors?
  - Correct print texts? Upper- / Lowercase? Typos?
- **Conditions**
  - Dividable by 3
  - Dividable by 5
  - Dividable by 3 and 5
  - Neither dividable by 3 nor 5
- **Loop**
  - Was a loop used at all?
  - End condition: 100 vs 101
  - Do you want to encourage a specific loop?

## Voir Annexe

Exercice créé par : <https://hpi.de/meinel/lehrstuhl/team/current-phd-students/sebastian-serth.html>

et <https://hpi.de/meinel/lehrstuhl/team/current-phd-students/christiane-hagedorn.html>

**Style:** What could be assessed with static program analysis?

Here are some aspects you might validate:

- **Code Formatting**
  - Spaces, newlines, indentation
  - Syntax errors?
- **Readability**
  - Comments / documentation
  - Naming of variables and functions
  - Reusability / modularity / Don't Repeat Yourself (DRY)
  - Typos? (e.g., in comments or output)
- **Architecture**
  - Performance / efficiency
  - Conventions / best practices
    - Condition: `if` vs `match` vs short-hand evaluation
    - Loop: `for` vs `while` vs `do...while`

# A vous de jouer (2)

## Cherchez les erreurs

Rule	Feedback Message

Exercice créé par : <https://hpi.de/meinel/lehrstuhl/team/current-phd-students/sebastian-serth.html>

et <https://hpi.de/meinel/lehrstuhl/team/current-phd-students/christiane-hagedorn.html>

# A vous de jouer (3)

## Cherchez les erreurs

### Sample Exercise: FizzBuzz



#### Task

- Write a program that prints the numbers from 1 to 100.
- If the number is dividable by 3 print *Fizz*,  
if the number is dividable by 5 print *Buzz* instead of the number.
- If the number is dividable by 3 and 5 print *FizzBuzz*.



```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
```

Sebastian Serth &  
Christiane Hagedorn

JTESS 2022

**How to create  
programming  
exercises with  
automated feedback**

9

Exercice créé par : <https://hpi.de/meinel/lehstuhl/team/current-phd-students/sebastian-serth.html>

et <https://hpi.de/meinel/lehstuhl/team/current-phd-students/christiane-hagedorn.html>

## A vous de jouer (4)

### Nombre Mystère

Faire un programme qui :

- Demande à l'utilisateur de trouver un nombre entre 1 et 100, que le programme aura choisi
- Indique à l'utilisateur s'il a trouvé ou sinon si c'est trop petit ou trop grand

# A vous de jouer (4)

## Nombre Mystère

```
# ===== C3 =====  
# Faire deviner un nombre compris entre 1 et 100  
# =====  
import random  
  
b_gagne = False  
print("Je pense à un nombre compris entre 1 et 100, essayez de le deviner")  
nb_mystere = random.randint(1, 100)  
nb_coup = 0  
  
while not b_gagne:  
    print('-----')  
    nb = int(input("A quel nombre pensez-vous : "))  
    nb_coup += 1  
    if nb < nb_mystere:  
        print('Trop petit')  
    elif nb > nb_mystere:  
        print('Trop grand')  
    else:  
        b_gagne = True  
        print('-----')  
        print(f'Bravo vous avez trouvé en {nb_coup} coups')  
        print('-----')
```



# Questions...