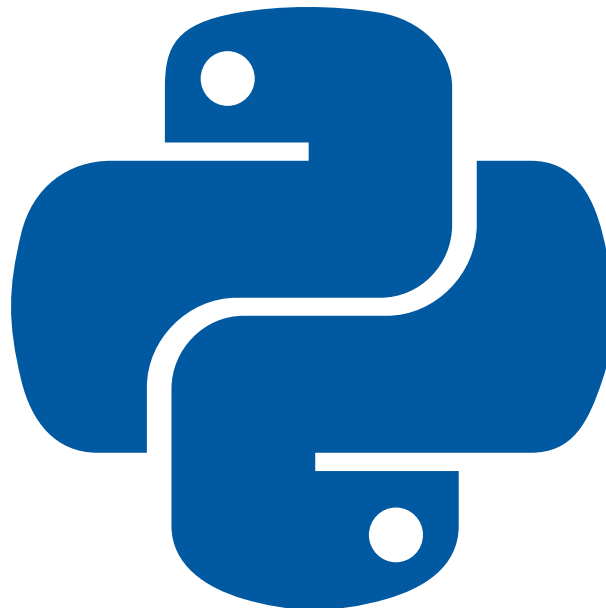




Structures Répétitives

Les bases de l'algorithmiques

Réalisation : Omar OUGHZAL





Introduction à les structures itératives

Ecrire un programme qui calcule la somme de 3 nombre saisis au clavier

Algorithme Somme

Var N, S : Réel

Début

S \leftarrow 0

Ecrire("Enter le nombre N° 1 :")

Lire(N)

S \leftarrow S + N

Ecrire("Enter le nombre N° 2 :")

Lire(N)

S \leftarrow S + N

Ecrire("Enter le nombre N° 3 :")

Lire(N)

S \leftarrow S + N

Ecrire("La Somme est :", S)

Fin

1

2

3

Mêmes instructions
répétées 3 fois

Et si on veut calculer
la somme de 1000
nombre 🤖





Introduction à les structures itératives

Ecrire un programme qui demande un nombre entre 0 et 20 jusqu'à ce que la réponse convienne

```
Ecrire("Enter le nombre entre 0 et 20: ")  
Lire(N)
```

Si $N < 0$ ou $N > 20$ alors

```
    Ecrire("Enter le nombre entre 0 et 20: ")  
    Lire(N)
```

finSi

1

Si $N < 0$ ou $N > 20$ alors

```
    Ecrire("Enter le nombre entre 0 et 20: ")  
    Lire(N)
```

finSi

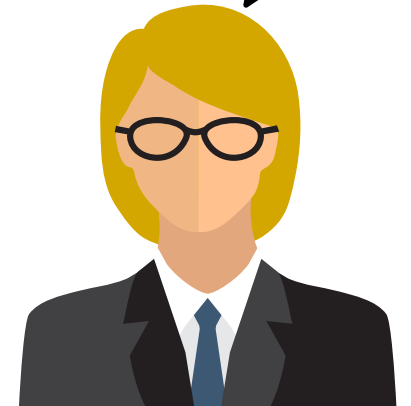
2



A quelle tentative
l'utilisateur va
entrer un nombre
entre 0 et 20



utilise la
structure
itérative





La structure répétitive

Une structure répétitive, encore appelée **boucle**, est utilisée quand une instruction ou une liste d'instruction, doit être répétée plusieurs fois. La répétition est soumise à une condition.

Il existe trois types de boucles en algorithmiques :

- **La boucle Pour.... Faire**
- **La boucle Tant Que..... Faire**
- **La boucle Répéter.... Jusqu'à**





La boucle Pour.... Faire

La boucle pour... faire permet de répéter une liste d'instructions un nombre connu de fois.

Syntaxe :

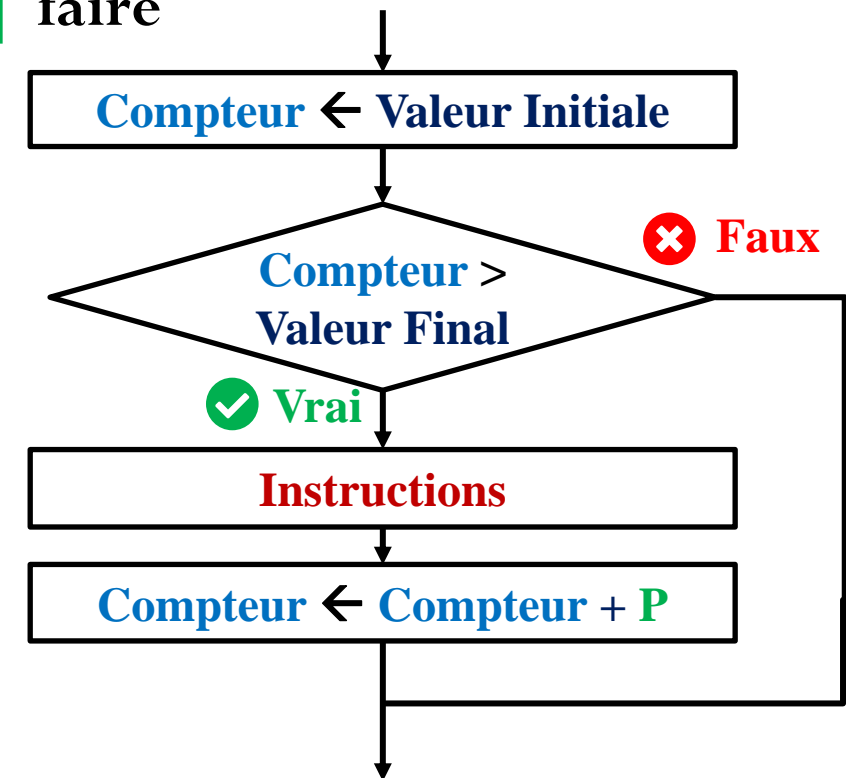
Pour **compteur** de **valeur initiale** à valeur finale [**Pas P**] faire

Instruction(s)

FinPour

La variable **compteur** est de type entier. Il prend **la valeur initiale** et à chaque tour la boucle, il es incrémenté de la valeur **du pas**.

Lorsque la variable compteur vaut **la valeur finale**, le traitement est exécuté une dernière fois puis le programme sort de la boucle





Exemple

Ecrire un algorithme permettant de calculer la somme des dix premiers nombres entiers

Algorithme Somme

Var S, i : entier

Début

$S \leftarrow 0$

Pour i de 1 à 10 **faire**

$S \leftarrow S + i$

Fin Pour

Ecrire (“la somme des dix premiers nombres entiers est : “, S)

Fin

Remarque :

- Par défaut la variable compteur est incrémenté de 1 à chaque tour de la boucle
- Pour modifier la valeur de l’incrément, il suffit de rajouter le mot **Pas**

Pour compteur \leftarrow valeur initiale **jusqu’à** valeur finale **pas** valeur de **Pas** **Faire**



Exemple

- Ecrire un algorithme permettant de calculer la somme des dix premiers nombres entiers pairs

Algorithme Somme

Var S, i : entier

Début

$S \leftarrow 0$

Pour i **de** 1 **à** 22 **pas** 2 **faire**

$S \leftarrow S + i$

Fin Pour

Ecrire (“la somme des dix premiers nombre entiers pairs est : “, S)

Fin



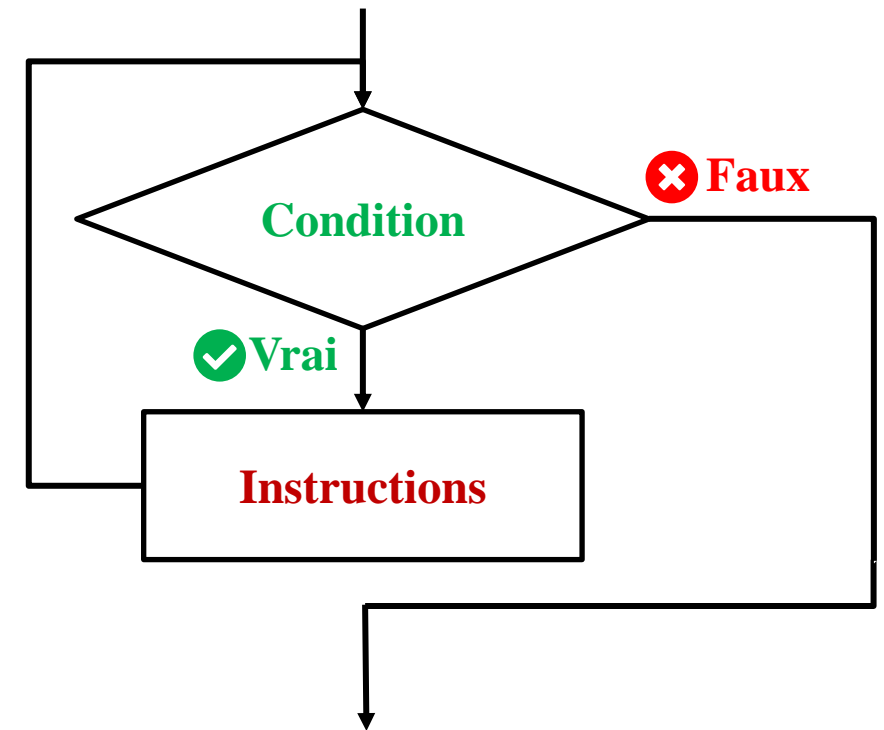
La boucle Tant Que..... Faire

La boucle Tant que permet de répéter un traitement tant que la condition est vraie.

Syntaxe :

Tant que **Condition** **Faire**
 Instruction(s)
Fin Tant Que

L'exécution de la boucle dépend de la valeur de la valeur de la condition. Si elle est vraie, le programme exécute les instructions qui suivent, jusqu'à ce qu'il rencontre la ligne Fin Tant Que. Il retourne ensuite sur la ligne du Tant Que, procède au même examen





Exemple

Ecrire un programme qui demande un caractère de valeur 'O' ou 'N', le programme doit redemander si la réponse ne correspond pas

Algorithme Contrôle_de_saisie

Var

Rep : caractère

Début

Ecrire ("Voulez-vous une copie de ce cours ? (O/N)")

Lire (Rep)

Tant Que Rep \neq 'O' et Rep \neq 'N' **faire**

Ecrire ("erreur de saisie")

Ecrire ("Voulez-vous une copie de ce cours ? (O/N)") **Lire** (Rep)

Fin Tant Que

Fin



La boucle Répéter.... Jusqu'à

Cette boucle sert à répéter une ou plusieurs instruction(s) jusqu'à ce qu'une condition soit vraie.

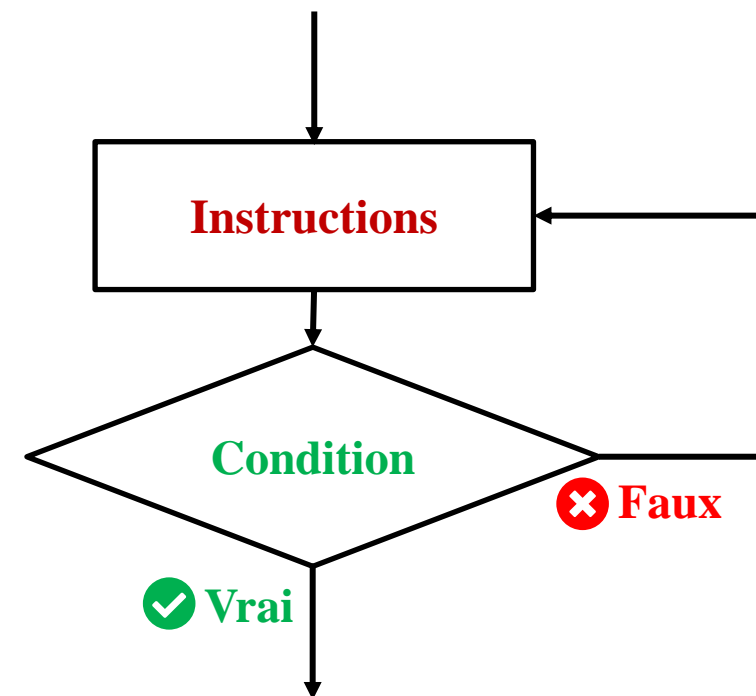
Syntaxe :

Répéter

Instruction (s)

Jusqu'à **Condition**

La liste d'instruction est exécutée, puis la condition est évaluée, si elle est fausse, le corps de la boucle est exécuté à nouveau puis la condition est réévaluée et si elle a la valeur vraie, le programme sort de la boucle et exécute l'instruction qui suit Jusqu'à.





Exemple

En utilisant la boucle répéter...jusqu'à, écrire un algorithme qui calcule la somme de N premiers nombres entiers. On suppose que N est strictement positif

Algorithme Somme

Var

S, i, N : entier

Début

Ecrire ("entrer une valeur strictement positive : ")

Lire (N)

$S \leftarrow 0$

$i \leftarrow 1$

Répéter

$S \leftarrow S + i$

$i \leftarrow i + 1$

Jusqu'à $i > N$

Ecrire ("la somme des ", N, "premiers entiers : ", S)

Fin



Traduction de la boucle Pour...FinPour

Pour **i** de **VI** à **VF** pas **P**

Faire

Instructions

FinFaire

for **i** in range(**VI**, **VF+1**, **P**):

Instructions

- La fonction range(VI,VF) génère une plage de nombre entier entre VI et VF-1 avec un pas de 1 par défaut
- Range(0,VF) → range(VF)



Traduction de l'instruction TantQue

```
Tantque Condition Faire
    Instructions
FinTantQue
```

```
while Condition :
    Instructions
```



Traduire l'instruction Répéter...Jusqu'à

Répéter

Instructions

jusqu'à condition



- L'instruction répéter jusqu'à n'existe pas en python, utiliser l'instruction tantQue à la place



Remarques

- On utilise la boucle “Pour” quand on connaît le nombre d’itérations à l’avance.
- Les boucle “Répéter” et “Tant Que” sont utilisées lorsqu’on ne sait pas au départ combien de fois il faudra exécuter ces boucles.
- A la différence de la boucle “Tant Que”, la boucle “Répéter” est exécuter au moins une fois.
- La condition d’arrêt de la boucle “Répéter” est la négation de la condition de poursuite de la boucle “Tant Que”

