



01

DÉFINITION

Qu'est ce qu'une structure conditionnelle?

02

FLOWCHART

Organigramme de la prise de décision.

03

STRUCTURE IF

Comparaison simple.

04

STRUCTURE IF ... ELIF ... ELSE

Comparaison avec deux ou plusieurs possibilités.

05


STRUCTURE IF IMBRIQUÉS

Comparaison avec plusieurs conditions imbriquées.

06

STRUCTURE UNIQUE

Structure conditionnelle en une ligne.

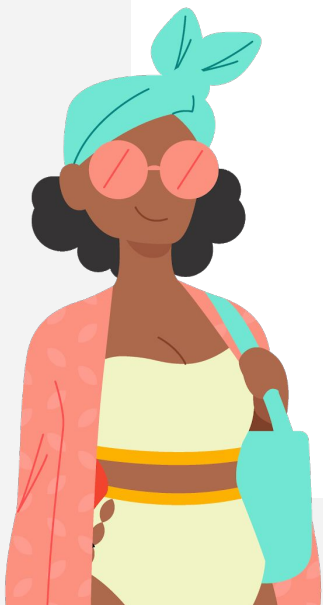




DÉFINITION

Souvent les problèmes nécessitent l'étude de plusieurs situations qui ne peuvent pas être traitées par une suite d'actions simples.

Avant l'exécution, nous ne connaissons pas quels cas de figure nous aurons à exécuter, il faut donc prévoir tous les cas possibles. Pour cela, nous devons utiliser des **structures conditionnelles** qui se basent sur des **conditions** auxquels nous pouvons répondre par **VRAI** ou **FAUX** afin d'exécuter différents blocs de code, adaptés à la situation rencontrée.



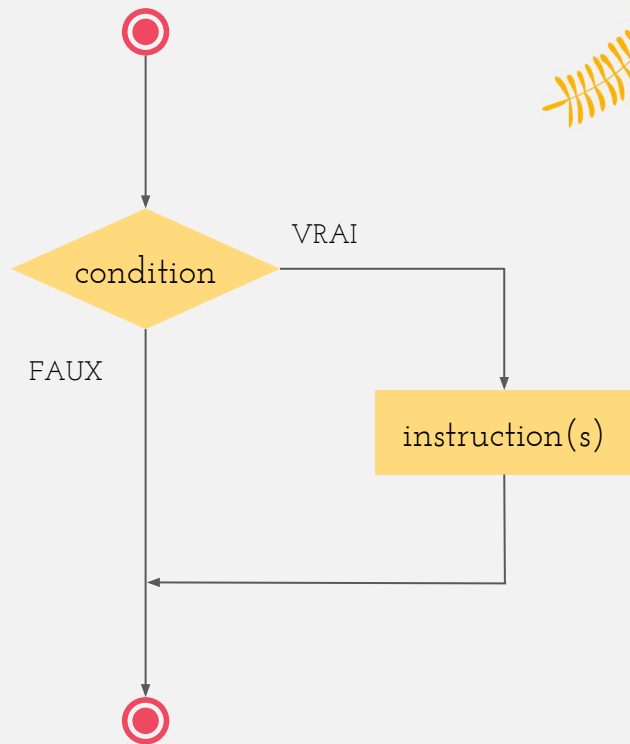
FLOWCHART

La **comparaison** est très utilisée pour la prise de décision. Python suppose que toute valeur non nulle est **VRAIE**, et **FAUSSE** si elles sont nulles ou vides.

Exemple : *Si il n'y a plus de banane, je pars en acheter.*

Si la condition est **VRAIE** une action est engagée, pour cela nous utilisons le mot "**SI**". Par contre rien n'est dit concernant le cas où il y a des bananes. Pour compléter la formulation nous devons spécifier une autre action. Ce qui donne :

Exemple : *Si il n'y a plus de banane je pars en acheter, sinon j'en mange une.*





STRUCTURE IF

La structure **IF** contient une **expression logique** à l'aide de laquelle les données sont **comparées** et une **décision** est prise sur la base du **résultat** de la **comparaison**.

Si l'expression **booléenne** est évaluée à **VRAI**, alors le bloc d'instruction(s) à l'intérieur de l'instruction **IF** est exécuté. Si l'expression **booléenne** est évaluée à **FAUX**, alors le premier ensemble de code après la fin de l'instruction **IF** est exécuté.

```
# Si il n'y a plus de banane, je pars en acheter.
```

```
bananas = 10  
if not bananas:  
    print("1. I'm going to buy bananas.")  
print("Good bye!")
```

```
bananas = 0  
if not bananas:  
    print("2. I'm going to buy bananas.")  
print("Good bye!")
```

Nous obtenons le résultat suivant :

```
Good bye!  
2. I'm going to buy bananas.  
Good bye!
```

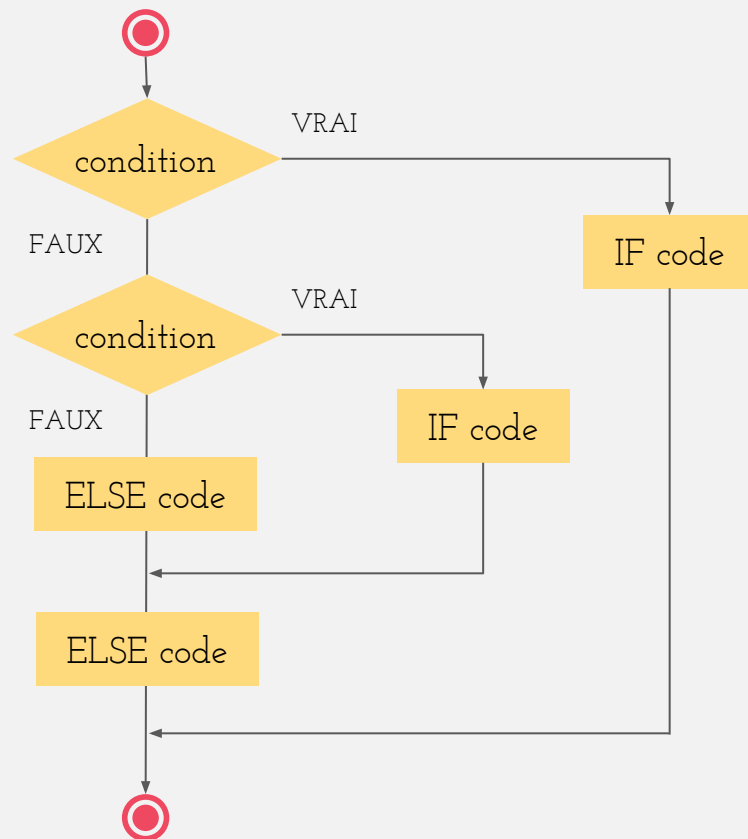


STRUCTURE IF ... ELIF ... ELSE

Une structure **IF** peut être suivie d'une instruction optionnelle **ELSE**, qui s'exécute lorsque la condition est **FAUSSE**. Il ne peut y avoir au maximum qu'une seule instruction **ELSE** à la suite de **IF**.

L'instruction **ELIF** permet de vérifier si plusieurs expressions sont **VRAIES** et d'exécuter un bloc de code dès qu'une des conditions est **VRAIE**.

Comme l'instruction **ELSE**, **ELIF** est facultative. Cependant, il peut y avoir autant d'instructions **ELIF** que vous souhaitez à la suite d'un **IF**.





STRUCTURE IF ... ELIF ... ELSE

Dans le premier cas, la condition en **IF** est invalidée donc la structure **ELSE** est exécutée car la variable **pineapples** est égale à 0.

Dans le second cas, la condition en **ELIF** est exécutée car la variable **cherries** est égale à 120, ce qui invalide la première condition ($\text{cherries} < 50$) et confirme la seconde ($\text{cherries} > 100$).

```
pineapples = 0
if pineapples:
    print "1. We have some pineapples."
else:
    print "1. We're out of pineapples."

cherries = 120
if cherries < 50:
    print "2. There are less than 50 cherries."
elif cherries > 100:
    print "2. There are more than 100 cherries."
else:
    print "2. There are between 51 and 100 cherries."
print "Good bye!"
```

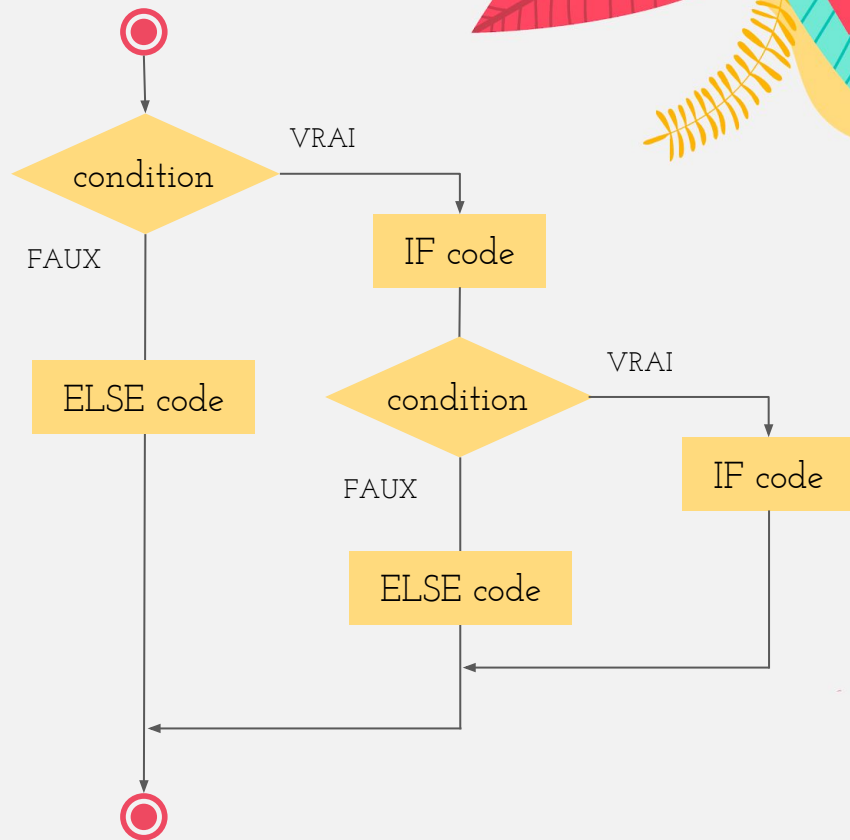
Nous obtenons le résultat suivant :

```
1. We're out of pineapples.
2. There are more than 100 cherries.
Good bye!
```

STRUCTURE IF IMBRIQUÉS

Dans certains cas, il peut arriver que vous souhaitiez vérifier une autre condition après qu'une condition se soit révélée vraie. Dans cette situation, vous pouvez utiliser la structure "IF imbriqués".

Dans une structure IF imbriqués, vous pouvez également avoir une structure IF...ELIF...ELSE à l'intérieur et ainsi de suite.





STRUCTURE IF IMBRIQUÉS

Nous avons 100 cerises, donc la première condition (`cherries < 200`) est validé.

Ensuite, nous retrouvons une structure en **IF...ELIF...ELSE**. La première condition de cette série est invalidée (`cherries == 150`), la condition suivante en **ELIF** est quand à elle validée (`cherries == 100`).

```
cherries = 100
if cherries < 200:
    print("There are less than 200 cherries.")
    if cherries == 150:
        print("There are 150 cherries.")
    elif cherries == 100:
        print("There are 100 cherries.")
    elif cherries < 100:
        print("There are less than 100 cherries.")
else:
    print("There are more than 200 cherries.")
print "Good bye!"
```

Nous obtenons le résultat suivant :

```
There are 100 cherries.
Good Bye
```

STRUCTURE UNIQUE

Il est possible de réduire une **structure conditionnelle à une seule ligne** mais ce n'est pas une pratique que je cautionne car elle diminue la lisibilité du code.

A titre informatif, si vous souhaitez enchaîner les déclarations, vous pouvez utiliser ";" pour les séparer.

Dans le cas d'une seule déclaration cette structure reste acceptable.

```
fruits = ["apple", "banana", "cherry"]  
if "apple" in fruits: print "yes!"
```

```
for fruit in fruits: print(fruit); print(id(fruit))
```

Nous obtenons le résultat suivant :

```
yes!  
apple  
5645755725  
banana  
5787957696  
cherry  
7886563246
```

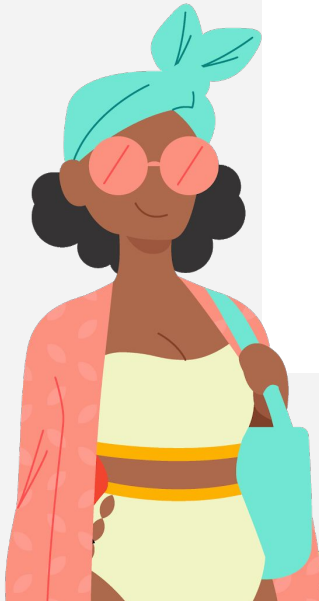


The background of the slide is a vibrant, colorful pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, with some featuring detailed vein patterns. The pattern is dense and fills the entire frame, creating a lively and summery atmosphere.

EXERCICES

EXERCICES

1. Créez une variable `a`, et assignez lui le résultat de `789 modulo 3`.
2. Testez si la valeur de `a` est supérieure à 0, si oui afficher le message 'a supérieur à 0', sinon afficher le message 'a est inférieur ou égal à 0'.
3. Complétez le code précédent en testant si `a` est inférieur à 0. Si `a` est inférieur à 0, afficher le message 'a est inférieur à 0' et adapter le message du cas par défaut.
4. Créez le dictionnaire `vegetables` contenant la clés `seeds` ayant pour valeur le tuple comportant les éléments suivants : eggplant, cucumber, bean, lentil, corn, pumpkin, tomato, pea.
Ajoutez la clés `roots` associée à la liste dont les éléments sont : garlic, beetroot, celeriac, fennel, onion, potato, radish.
Idem pour la clés `leaves` avec le tuple suivant : celery, chicory, cabbage, spinach, lettuce, leek, salsifi.



EXERCICES

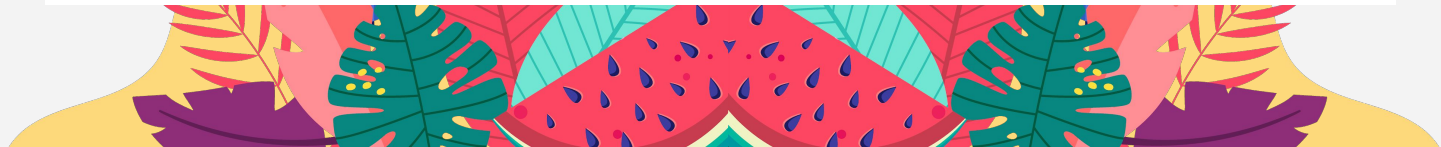
5. Testez si le dictionnaire *vegetables* est vide. Affichez le dictionnaire s'il existe, sinon afficher un message pour avertir que le dictionnaire est vide.

6. Complétez le code précédent pour vérifier que la clés *roots* est contenue dans le dictionnaire *vegetables*. Si oui, afficher la valeur de cette clés, sinon afficher les clés du dictionnaire.

7. Compléter le code précédent pour vérifier que “*carrot*” fait partie des valeurs des légumes de type racines. Si la valeur “*carrot*” n'est pas présente dans la liste, afficher le message, “*carrot is missing in roots vegetables list.*”

8. Créez une variable *priority_level* et lui assigner la valeur 3.

9. Si *priority_level* est égal à 0, alors afficher le message low.
Si *priority_level* est égal à 1, alors afficher le message normal.
Si *priority_level* est égal à 2, alors afficher le message high.
Si *priority_level* est égal à 3, afficher le message urgent.
Sinon afficher le message not defined.



The background of the slide is a vibrant, colorful pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, creating a dense and lively border around the central text. Some leaves are large and detailed, while others are smaller and more stylized. The overall effect is a tropical and energetic aesthetic.

SOLUTIONS

CORRECTION

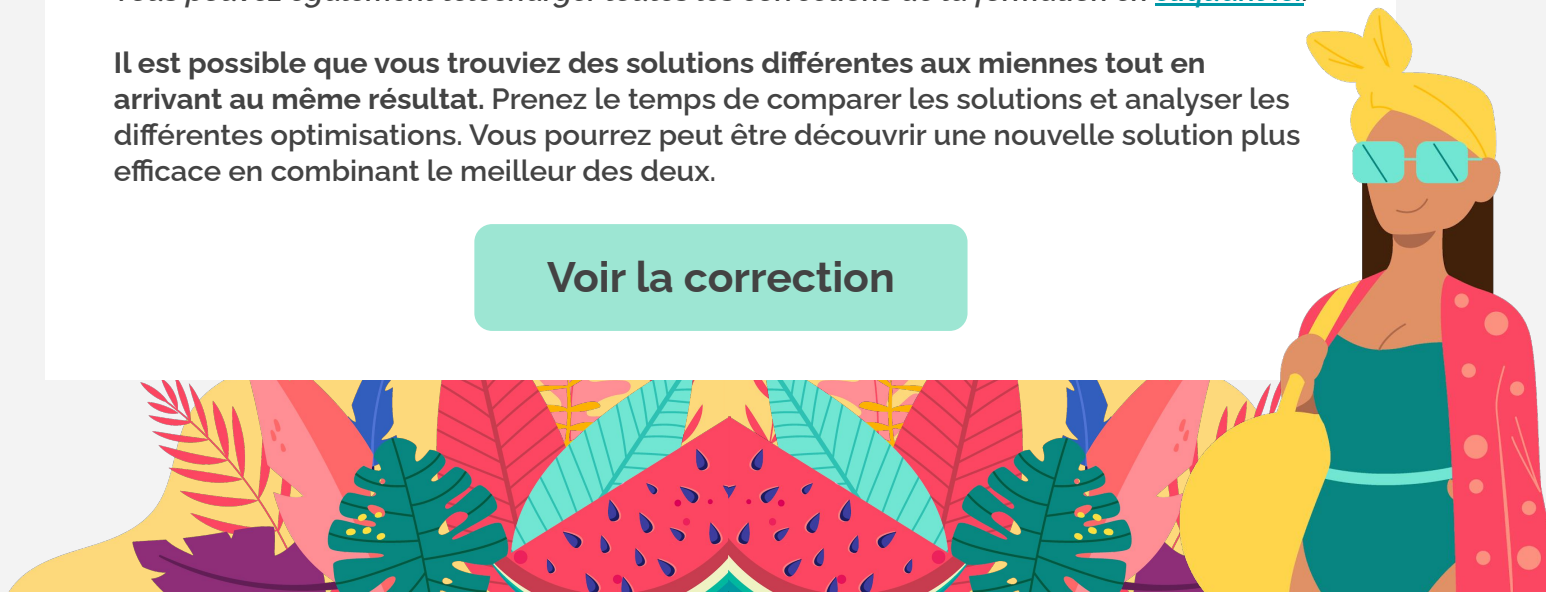
Pour visualiser la correction du chapitre cliquer sur le bouton ci-dessous.

*Le fichier de la correction s'ouvrira dans un nouvel onglet de votre navigateur préféré.
Pour cela vous devez avoir accès à une connexion Internet.*

Vous pouvez également télécharger toutes les corrections de la formation en [cliquant ici](#).

Il est possible que vous trouviez des solutions différentes aux miennes tout en arrivant au même résultat. Prenez le temps de comparer les solutions et analyser les différentes optimisations. Vous pourrez peut être découvrir une nouvelle solution plus efficace en combinant le meilleur des deux.

Voir la correction





**Félicitation vous avez terminé
le chapitre sur les structures
conditionnelles!**

A decorative border of various tropical leaves in vibrant colors like red, orange, yellow, green, and purple surrounds the central white area.

CRÉDITS

- Modèle de la présentation par [Slidesgo](#)
- Icônes par [Flaticon](#)
- Images et infographies par [Freepik](#)