

## TP5 : des variables aux boucles...

Initiation à l'algorithmique et la programmation

L3 TAL — Semestre 5 (2025)

### Questions conceptuelles

#### Problème 1: Différence entre = et ==

Quelle est la différence entre = et == en Python ? Donnez un exemple de chacun.

#### Problème 2: Erreur de type

Pourquoi ce code cause-t-il une erreur ?

```
1 age = "25"  
2 next_year = age + 1
```

#### Problème 3: Différence entre while et if

Quelle est la différence entre while et if ? Quand utiliseriez-vous chacun ?

#### Problème 4: Opérateurs de division

Quelle est la différence entre les opérateurs / et // ? Donnez des exemples où chacun est utile.

### Révisions des bases

#### Variables et conversion de types

#### Problème 5: Conversion simple

Demandez à l'utilisateur d'entrer un nombre (qui sera stocké comme une chaîne de caractères). Convertissez-le en entier, multipliez-le par 2, puis affichez le résultat.

#### Problème 6: Concaténation vs addition

Quelle est la différence entre ces deux lignes ?

```
1 print("5" + "3")  
2 print(5 + 3)
```

## Structures conditionnelles (if-elif-else)

### Problème 7: Maximum de deux nombres

Demandez deux nombres à l'utilisateur et affichez le plus grand des deux.

### Problème 8: Catégorie d'âge

Demandez l'âge et affichez la catégorie :

- Enfant : 0-12 ans
- Adolescent : 13-17 ans
- Adulte : 18-64 ans
- Senior : 65 ans et plus

## Chaînes de caractères et indexation

### Problème 9: Premier et dernier caractère

Demandez un mot à l'utilisateur. Affichez le premier et le dernier caractère du mot.

### Problème 10: Longueur de chaîne

Demandez une phrase à l'utilisateur et affichez combien de caractères elle contient (espaces inclus).

### Problème 11: Extraction de sous-chaîne

Avec le mot "Fontaines", utilisez l'indexation pour extraire et afficher :

- Les trois premiers caractères
- Les trois derniers caractères
- Tous les caractères sauf le premier et le dernier

### Problème 12: Majuscules et minuscules

Demandez un mot à l'utilisateur. Affichez-le en majuscules, puis en minuscules.

### Problème 13: Vérification de caractère

Demandez une chaîne et un caractère à l'utilisateur. Vérifiez si le caractère est présent dans la chaîne (utilisez `in`).

## Opérateurs mathématiques

### Problème 14: Division entière et modulo

Demandez deux nombres entiers. Affichez :

- Le résultat de la division normale
- Le résultat de la division entière
- Le reste de la division

### Problème 15: Pair ou impair

Demandez un nombre entier. Utilisez l'opérateur modulo pour déterminer s'il est pair ou impair.

### Problème 16: Puissance

Demandez deux nombres :  $a$  et  $b$ . Calculez et affichez la  $a^b$ .

## Boucles while simples

### Problème 17: Compte à rebours simple

Utilisez une boucle `while` pour afficher les nombres de 5 à 1, puis affichez "Décollage !".

### Problème 18: Table de multiplication

Demandez un nombre à l'utilisateur. Affichez sa table de multiplication de 1 à 5 en utilisant une boucle `while`.

### Problème 19: Somme des N premiers nombres

Demandez un nombre  $N$ . Calculez la somme de tous les nombres de 1 à  $N$  en utilisant une boucle `while`.

### Problème 20: Répétition de chaîne

Demandez un mot et un nombre à l'utilisateur. Affichez le mot ce nombre de fois (une fois par ligne) en utilisant une boucle `while`.

### Problème 21: Compte jusqu'à condition

Créez une boucle qui demande des nombres à l'utilisateur et qui s'arrête quand l'utilisateur entre -1. Affichez "Programme terminé" à la fin.

## Compléter le code

### Problème 22: Complétez le compteur

Complétez ce code pour compter de 1 à 10 :

```
1 number = 1
2 while number ___ 10:
3     print(number)
4     number = number ___ 1
```

### Problème 23: Complétez le vérificateur de mot de passe

Complétez ce code (le mot de passe doit avoir 8+ caractères et contenir "@" ) :

```
1 password = input("Entrez le mot de passe : ")
2 if ___ >= 8 and _____:
3     print("Mot de passe valide")
4 else:
5     print("Mot de passe invalide")
```

### Problème 24: Complétez la calculatrice

Complétez ce code de calculatrice :

```
1 num1 = int(input("Premier nombre : "))
2 num2 = int(input("Deuxieme nombre : "))
3 operation = input("Operation (+, -, *, /) : ")
4
5 if ___ == ___:
6     result = num1 + num2
7 elif ___:
8     result = num1 - num2
9 # Continuez...
```

## Questions de prédiction

### Problème 25: Quelle est la sortie ? (1)

Prédisez la sortie avant d'exécuter le code :

```
1 x = 5
2 y = 2
3 result = x // y + x % y
4 print(result)
```

### Problème 26: Quelle est la sortie ? (2)

Que va afficher chaque instruction print ?

```
1 text = "Python"
2 print(text[1:4])
3 print(text[-1])
4 print(text[:3] + text[3:])
```

### Problème 27: Quelle est la sortie ? (3)

Tracez la boucle et prédisez toute la sortie :

```
1 count = 1
2 while count <= 3:
3     print(count * 2)
4     count = count + 1
```

### Problème 28: Quelle est la sortie ? (4)

Qu'est-ce qui s'affiche et pourquoi ?

```
1 name = "alice"
2 if name == "Alice":
3     print("Bonjour Alice !")
4 else:
5     print("Qui etes-vous ?")
```

## Exercices de débogage

### Problème 29: Corrigez le convertisseur de température

Le code suivant contient des erreurs. Identifiez-les et corrigez-les :

```
1 celcius = input("Entrez la temperature en Celsius : ")
2 fahrenheit = celcius * 9/5 + 32
3 print("Temperature en Fahrenheit : " + fahrenheit)
```

### Problème 30: Corrigez le compteur

Ce code a une erreur. Quelle est-elle ?

```
1 count = 0
2 while count < 5:
3     print("Le compte est : " + count)
4     count = count + 1
```

### Problème 31: Corrigez le calculateur de notes

Ce code fonctionne mais a un problème de style. Que devrait-on changer ?

```
1 score = int(input("Entrez votre score : "))
2 if score >= 90:
3     print("Note : A")
4 elif score >= 80:
5     print("Note : B")
6 elif score >= 70:
7     print("Note : C")
8 elif score >= 60:
9     print("Note : D")
10 if score < 60:
11     print("Note : F")
```

### Problème 32: Corrigez le validateur de mot de passe

La logique est inversée. Corrigez-la pour que les mots de passe forts aient 10+ caractères, moyens 6-9, et faibles moins de 6.

```
1 password = input("Entrez le mot de passe : ")
2 if len(password) > 8:
3     print("Mot de passe fort")
4 elif len(password) > 6:
5     print("Mot de passe moyen")
6 else:
7     print("Mot de passe faible")
```

### Problème 33: Identifiez le problème potentiel

Ce code fonctionne actuellement. Mais identifiez ce qui le rendrait une boucle infinie si on changeait une seule chose :

```
1 number = 10
2 while number > 0:
3     print(number)
4     number = number - 1
5 print("Termine !")
```

## Défis logiques avancés

### Problème 34: Calculateur d'année bissextile

Écrivez un programme qui détermine si une année est bissextile. Règles :

- Divisible par 4 : année bissextile
- SAUF si divisible par 100 : pas une année bissextile
- SAUF si divisible par 400 : année bissextile quand même

Testez avec : 2000, 1900, 2024, 2023

### Problème 35: Détective de suites numériques

Demandez trois nombres à l'utilisateur. Déterminez s'ils forment :

- Une suite arithmétique (chaque nombre diffère du même montant : 2, 5, 8)
- Une suite géométrique (chaque nombre est multiplié par le même facteur : 2, 6, 18)
- Ni l'un ni l'autre

### Problème 36: Compression de chaînes

Étant donné une chaîne comme "aaabbbcc", compressez-la en "a3b3c2". Si la compression n'économise pas d'espace, retournez la chaîne originale.

### Problème 37: Système de connexion avancé

Créez un système de connexion où :

- Le nom d'utilisateur doit avoir au moins 5 caractères
- Le mot de passe doit avoir au moins 8 caractères ET contenir au moins un chiffre
- L'utilisateur a 3 tentatives au total pour les échecs de nom d'utilisateur ET de mot de passe combinés
- Suivez combien de tentatives restent après chaque échec

### Problème 38: Vérificateur de nombre premier

Demandez un nombre et déterminez s'il est premier. Un nombre premier n'est divisible que par 1 et lui-même. Utilisez une boucle `while` pour tester les diviseurs de 2 jusqu'au nombre.

## Puzzles logiques avancés

### Problème 39: Racine numérique

Continuez à additionner les chiffres d'un nombre jusqu'à obtenir un seul chiffre. Par exemple :  $987 \rightarrow 9+8+7 = 24 \rightarrow 2+4 = 6$ . Utilisez une boucle `while` et des méthodes de chaînes.

### Problème 40: Tournoi Pierre-Papier-Ciseaux

Créez un jeu de pierre-papier-ciseaux où :

- Meilleur de 5 manches
- L'ordinateur choisit aléatoirement

- Suivez le score des deux joueurs
- Annoncez le gagnant après 5 manches

#### **Problème 41: Nombre d'Armstrong**

Vérifiez si un nombre est égal à la somme de ses chiffres élevés à la puissance du nombre de chiffres. Exemple :  $153 = 1^3 + 5^3 + 3^3$  (un nombre à 3 chiffres)

#### **Problème 42: Inverser les mots, garder l'ordre**

Entrée : "Bonjour le Monde"

Sortie : "ruojnoB el ednoM"

(Chaque mot est inversé mais reste à la même position)

#### **Problème 43: Conjecture de Collatz**

Commencez avec n'importe quel entier positif. Si pair, divisez par 2. Si impair, multipliez par 3 et ajoutez 1. Répétez jusqu'à atteindre 1. Comptez combien d'étapes cela prend. (Essayez avec 7, 10, 27)

## Mini-projets

### Problème 44: Jeu de devinettes 2.0

Version améliorée :

- Nombre aléatoire entre 1-100
- Maximum 7 tentatives
- Après chaque mauvaise réponse, indiquez combien de tentatives restent
- Indiquez s'ils sont à moins de 10 du résultat (« vous chauffez ! »)

### Problème 45: Système bancaire simple

Créez un distributeur automatique simple :

- Solde initial : 1000€
- Menu : Consulter le solde, Déposer, Retirer, Quitter
- Utilisez une boucle `while` pour le menu
- N'autorisez pas le retrait de plus que le solde
- N'autorisez pas les dépôts négatifs