

## TP5 : des variables aux boucles... [Correction succincte]

### 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.

*Réponse : = est un opérateur d'assignation (= permet d'assigner à une variable son contenu), tandis que == est un opérateur de comparaison (permet de vérifier si les membres gauche et droite sont égaux).*

*Exemples : `x = 5` (assignation) et `if x == 5:` (comparaison)*

### Problème 2: Erreur de type

Pourquoi ce code cause-t-il une erreur ?

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

*Réponse : l'opération `age + 1` est une opération invalide. En effet, on ne peut pas additionner une chaîne de caractère à un nombre entier. Il faut d'abord convertir : `next_year = int(age) + 1`*

### Problème 3: Différence entre while et if

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

*Réponse : `if` exécute le code une seule fois si la condition est vraie, tandis que `while` répète le code tant que la condition reste vraie. On utilise `if` pour une décision unique et `while` pour une répétition (itérations).*

### Problème 4: Opérateurs de division

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

*Réponse : / effectue une division normale (résultat décimal), // effectue une division entière (arrondi vers le bas). Exemples : `7/2` donne 3.5, `7//2` donne 3. La division entière est utile pour compter des groupes entiers, vérifier qu'un entier est un multiple d'un autre entier (ex : on utilise la division entière pour vérifier qu'un nombre est pair).*

# 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.

Réponse :

```
1 nombre = input("Entrez un nombre : ")
2 resultat = int(nombre) * 2
3 print(resultat)
```

### Problème 6: Concaténation vs addition

Quelle est la différence entre ces deux lignes ?

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

Réponse : La première ligne affiche "53" (concaténation de chaînes), la seconde affiche 8 (addition de nombres).

## Structures conditionnelles (if-elif-else)

### Problème 7: Maximum de deux nombres

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

Réponse :

```
1 num1 = int(input("Premier nombre : "))
2 num2 = int(input("Deuxieme nombre : "))
3 if num1 > num2:
4     print("Le plus grand est :", num1)
5 else:
6     print("Le plus grand est :", num2)
```

### 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

Réponse :

```
1 age = int(input("Entrez votre age : "))
2 if age <= 12:
3     print("Enfant")
4 elif age <= 17:
5     print("Adolescent")
6 elif age <= 64:
7     print("Adulte")
8 else:
9     print("Senior")
```

## 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.

Réponse :

```
1 mot = input("Entrez un mot : ")
2 print("Premier caractere :", mot[0])
3 print("Dernier caractere :", mot[-1])
```

### Problème 10: Longueur de chaîne

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

Réponse :

```
1 phrase = input("Entrez une phrase : ")
2 print("Nombre de caracteres :", len(phrase))
```

### 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

Réponse :

```
1 mot = "Fontaines"
2 print(mot[:3])      # "Fon"
3 print(mot[-3:])    # "nes"
4 print(mot[1:-1])   # "ontaine"
```

### Problème 12: Majuscules et minuscules

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

*Réponse :*

```
1 mot = input("Entrez un mot : ")
2 print("Majuscules :", mot.upper())
3 print("Minuscules :", mot.lower())
```

### 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).

*Réponse :*

```
1 chaine = input("Entrez une chaine : ")
2 caractere = input("Entrez un caractere : ")
3 if caractere in chaine:
4     print("Le caractere est present")
5 else:
6     print("Le caractere n'est pas present")
```

## 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

*Réponse :*

```
1 num1 = int(input("Premier nombre : "))
2 num2 = int(input("Deuxieme nombre : "))
3 print("Division normale :", num1 / num2)
4 print("Division entiere :", num1 // num2)
5 print("Reste :", num1 % num2)
```

### Problème 15: Pair ou impair

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

*Réponse :*

```
1 nombre = int(input("Entrez un nombre : "))
2 if nombre % 2 == 0:
```

```
3     print("Pair")
4 else:
5     print("Impair")
```

### Problème 16: Puissance

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

Réponse :

```
1 a = int(input("Entrez a : "))
2 b = int(input("Entrez b : "))
3 print("Resultat :", 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 !".

Réponse :

```
1 nombre = 5
2 while nombre >= 1:
3     print(nombre)
4     nombre -= 1
5 print("Decollage !")
```

### 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`.

Réponse :

```
1 nombre = int(input("Entrez un nombre : "))
2 i = 1
3 while i <= 5:
4     print(nombre, "x", i, "=", nombre * i)
5     i += 1
```

### 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`.

Réponse :

```
1 n = int(input("Entrez N : "))
2 somme = 0
3 i = 1
4 while i <= n:
```

```
5     somme = somme + i
6     i += 1
7 print("Somme :", somme)
```

### 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`.

*Réponse :*

```
1 mot = input("Entrez un mot : ")
2 nombre = int(input("Combien de fois : "))
3 i = 0
4 while i < nombre:
5     print(mot)
6     i += 1
```

### 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.

*Réponse :*

```
1 nombre = 0
2 while nombre != -1:
3     nombre = int(input("Entrez un nombre (-1 pour arreter) : "))
4 print("Programme termine")
```

## 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
```

*Réponse :*

```
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")
```

Réponse :

```
1 password = input("Entrez le mot de passe : ")
2 if len(password) >= 8 and "@" in password:
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...
```

Réponse :

```
1 num1 = int(input("Premier nombre : "))
2 num2 = int(input("Deuxieme nombre : "))
3 operation = input("Operation (+, -, *, /) : ")
4
5 if operation == "+":
6     result = num1 + num2
7 elif operation == "-":
8     result = num1 - num2
9 elif operation == "*":
10    result = num1 * num2
11 elif operation == "/":
12    result = num1 / num2
13 print("Resultat :", result)
```

## 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)
```

Réponse : 3. Car  $5 // 2 = 2$  et  $5 \% 2 = 1$ , donc  $2 + 1 = 3$

### 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:])
```

Réponse :

- `text[1:4]` affiche "yth"
- `text[-1]` affiche "n"
- `text[:3] + text[3:]` affiche "Python" (reconstitue le mot entier)

### 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
```

Réponse : Le programme affiche :

- 2 ( $1 * 2$  ; car `count = 1`)
- 4 ( $2 * 2$  ; car `count = 2`)
- 6 ( $3 * 2$  ; car `count = 3`)

### 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 ?")

```

Réponse : "Qui etes-vous ?" s'affiche car Python est sensible à la casse : "alice" n'est pas égal à "Alice" (minuscule vs majuscule).

## 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)

```

Réponse : Deux erreurs :

1. *celcius* doit être converti en nombre : `float(celcius)`
2. *fahrenheit* doit être converti en chaîne pour la concaténation : `str(fahrenheit)`

Code corrigé :

```

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

```

Réponse : On ne peut pas concaténer un entier avec une chaîne. Il faut convertir `count` en chaîne :

```

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

```

Ou utiliser une virgule : `print("Le compte est :", count)`

### 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")
```

Réponse : Le dernier `if` devrait être un `else` pour maintenir la cohérence de la structure conditionnelle :

```
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 else:
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")
```

Réponse : Les seuils sont incorrects. Code corrigé :

```
1 password = input("Entrez le mot de passe : ")
2 if len(password) >= 10:
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 !")
```

Réponse : Si on oublie la ligne `number = number - 1` ou si on change `- 1` en `+ 1`, la boucle deviendrait infinie car la condition `number > 0` resterait toujours vraie.