

Python - Base

Questions :

1. Directement dans la console, effectuez les calculs suivants :

```
| >>> 15+3
```

```
| >>> 2-17
```

```
| >>> 8+3*4
```

```
| >>> (8+3)*4
```

```
| >>> 20/3
```

```
| >>> 20//3
```

```
| >>> 20%3
```

2. Ces opérations ont-elles le comportement attendu?

Oui, ces opérations produisent le comportement attendu. Les résultats montrent les règles de priorité des opérations en Python, comme la multiplication avant l'addition et l'usage des parenthèses pour modifier l'ordre.

3. Toujours dans la console, saisissez les instructions suivantes :

```
| >>> a=3
```

```
| >>> a
```

```
| >>> a+10
```

```
| >>> a/2
```

```
| >>> a
```

```
| >>> a+=5
```

```
| >>> a
```

4. Ces opérations ont-elles le comportement attendu? Expliquez le résultat de la dernière instruction?

Oui, ces opérations se comportent comme prévu. La dernière instruction **a += 5** ajoute 5 à la valeur actuelle de **a**, ce qui met à jour sa valeur de 3 à 8.

Cette opération **d'incrémentation** modifie directement la variable _a.

5. Encore et toujours dans la console, saisissez les instructions suivantes :

```
| >>> a=2
```

```
| >>> type(a)
```

```
| >>> b=5.5
```

```
| >>> type(b)
```

```
| >>> c="Hello "
```

```
| >>> type(c)
```

```
| >>> d="vous !"
```

```
| >>> a+b
```

```
| >>> type(a+b)
```

```
| >>> c+d
```

```
| >>> type(c+d)
```

6. Comment s'est comporté le signe « + » ?

Le signe « + » s'est comporté différemment selon le type des variables :

- Pour `a + b`, il a effectué une **addition** entre un entier et un flottant, donnant un résultat de type flottant.
- Pour `c + d`, il a effectué une **concaténation** entre deux chaînes de caractères, créant une nouvelle chaîne `Hello vous !`.

Ainsi, le signe « + » adapte son comportement selon le type des opérandes : addition pour les nombres et concaténation pour les chaînes de caractères.

III. Un premier programme

Mo

Après ces nombreux tests, il est temps de garder des traces de votre travail...vous allez sauvegarder une suite d'instruction : c'est un programme!

Questions :

1. Dans la partie éditeur de l'IDE recopier le programme suivant puis exécutez-le (en cliquant sur la flèche verte) :

```
1 a=12
2 b=5
3 b=a*b
4 a=3=b
5 print("La valeur de a est :",a)
6 print("La valeur de b est :",b)
```

2. Expliquez le comportement de ce programme.

Le programme donné contient **une erreur** à la ligne 4 : **a=3=b**. En Python, on ne peut pas utiliser l'opérateur = de cette façon. La syntaxe correcte pour affecter une valeur est soit `a = 3` ou `b = 3`, mais **on ne peut pas combiner deux affectations** dans une seule expression.

En exécutant ce code, l'IDE renverra une **erreur de syntaxe** à la ligne 4.

IV. Aide à la conjecture

Mo

1. En vous inspirant de la question 10 traitée sur Capytale (le périmètre d'un cercle), écrivez un programme qui permet pour un entier naturel n saisi en entrée par l'utilisateur de renvoyer le résultat de l'expression $n^2 + (n + 1)^2 - (n + 2)^2 + (n + 3)^2$.
2. Testez votre programme avec plusieurs valeurs différentes.
3. Quelle conjecture pouvez-vous émettre?
4. Pouvez-vous le prouver algébriquement?

Voir mon site web.