

Opérateurs du jour 3

Booléen

Un type de données booléen représente l'une des deux valeurs: *True* ou *False*. L'utilisation de ces types de données sera claire une fois que nous commencerons à utiliser l'opérateur de comparaison. La première lettre T pour True et F pour False devrait être capitale contrairement à JavaScript. Exemple:
Valeurs booléennes

```
print (True)
print (False)
```

Opérateurs

Python Language prend en charge plusieurs types d'opérateurs. Dans cette section, nous nous concentrerons sur quelques-uns d'entre eux.

Opérateurs d'affectation

Les opérateurs d'affectation sont utilisés pour attribuer des valeurs aux variables. Prenons = comme exemple. Le signe égal en mathématiques montre que deux valeurs sont égales, mais en python, cela signifie que nous stockons une valeur dans une certaine variable et nous l'appelons affectation ou une valeur attribuée à une variable. Le tableau ci-dessous montre les différents types d'opérateurs d'attribution Python, tirés de W3School.

| Operator | Example | Same As |
|----------|---------|------------|
| = | x = 5 | x = 5 |
| += | x += 3 | x = x + 3 |
| -= | x -= 3 | x = x - 3 |
| *= | x *= 3 | x = x * 3 |
| /= | x /= 3 | x = x / 3 |
| %= | x %= 3 | x = x % 3 |
| //= | x //= 3 | x = x // 3 |
| **= | x **= 3 | x = x ** 3 |
| &= | x &= 3 | x = x & 3 |
| = | x = 3 | x = x 3 |
| ^= | x ^= 3 | x = x ^ 3 |
| >>= | x >>= 3 | x = x >> 3 |
| <<= | x <<= 3 | x = x << 3 |

Opérateurs arithmétiques:

• Addition (+): $a + b$ • soustraction (-): $a - b$ • multiplication (*): $a * b$ • division (/): a / b • modulus (%): $a \% b$ • division (//): $a // b$ • exponentiation (**): $a ** b$

| Operator | Name | Example |
|----------|----------------|----------|
| + | Addition | $x + y$ |
| - | Subtraction | $x - y$ |
| * | Multiplication | $x * y$ |
| / | Division | x / y |
| % | Modulus | $x \% y$ |
| ** | Exponentiation | $x ** y$ |
| // | Floor division | $x // y$ |

Exemple: entiers

Opérations arithmétiques à Python

Entiers

```
print('Addition: ', 1 + 2)      # 3
print('Subtraction: ', 2 - 1)   # 1
print('Multiplication: ', 2 * 3) # 6
print ('Division: ', 4 / 2)      # 2.0  Division in Python
gives floating number
print('Division: ', 6 / 2)       # 3.0
print('Division: ', 7 / 2)       # 3.5
print('Division without the remainder: ', 7 // 2)  # 3,
gives without the floating number or without the remaining
print ('Division without the remainder: ', 7 // 3)  # 2
print('Modulus: ', 3 % 2)        # 1, Gives the remainder
print('Exponentiation: ', 2 ** 3) # 9 it means 2 * 2 * 2
```

Exemple: flottants

Nombres flottants

```
print ('Floating Point Number, Pi', 3.14) Print ('Floating Point Number, Gravity', 9.81)
```

Exemple: nombres complexes

```
# Nombres complexes PRINT ('Numéro complexe:', 1 + 1J) Imprimer ('Multiplier des nombres complexes:', (1 + 1J) * (1 - 1J))
```

Déclarons une variable et attribuons un type de données numérique. Je vais utiliser une variable à caractère unique mais n'oubliez pas de ne pas développer l'habitude de déclarer de tels types de variables. Les noms de variables doivent être tout le temps mnémonique.

Exemple:

```
# Déclarant la variable en haut d'abord
```

```
A = 3 # A est un nom de variable et 3 est un type de données entier B = 2 # B est un nom de variable et 3 est un type de données entier
```

```
# Opérations arithmétiques et attribuant le résultat à une variable totale = a + b diff = a - b produit = a * b division = a / b reste = a % b plancher_division = a // b exponentielle = a ** b
```

```
# J'aurais dû utiliser la somme au lieu du total, mais la somme est une fonction intégrée - essayez d'éviter de remplacer les fonctions intégrées Imprimer (total) # Si vous n'étiquetez pas votre imprimé avec une chaîne, vous ne savez jamais d'où le résultat vient de l'imprimer ('A - V13} B =', total) print ('A - B =', diff) imprimer ('a * b =', un produit) imprimer (a / b) =', division) print ('a % b =', reste) print ('a // b =', plancher_division) print ('a ** b =', exponentiation)
```

Exemple:

```
print ('== ajout, soustraction, multiplication, division, module ==')

# Déclarant les valeurs et les organisant ensemble num_one = 3 num_two = 4

# Opérations arithmétiques total = num_one + num_two diff = num_two - num_one Product = num_one * num_two div = num_two / num_one reste = num_two % num_one

# Impression Valeurs avec étiquette Print ('Total:', Total) Print ('Difference:', Diff) Print ('Product:', Product) Print ('Division:', Div) Print ('Rester:', reste)
```

Commençons par commencer à connecter les points et à utiliser ce que nous savons déjà pour calculer (surface, volume, densité, poids, périmètre, distance, force).

Exemple:

```
# Calcul de la zone d'un rayon de cercle = 10 # Radius d'un cercle Area_Of_Circle = 3.14 * RADIUS ** 2 # Two * Sign signifie Exponent OU Power Print ('Zone d'un cercle:', Area_Of_Circle) # Calcul de la zone d'un rectangle print ('zone du rectangle:', area_of_rectangle) # calculer un poids d'une masse d'objet = 75 Gravity = 9.81 poids = masse * gravité print (poids, 'n') # unité d'ajout au poids # calculer la densité d'une masse liquide = 75 # en kg
```

```
Volume = 0,075 # en densité de mètre cube = Masse / Volume # 1000 kg / m³
```

Opérateurs de comparaison

Dans la programmation, nous comparons les valeurs, nous utilisons des opérateurs de comparaison pour comparer deux valeurs. Nous vérifions si une valeur est supérieure ou inférieure ou égale à une autre valeur. Le tableau suivant montre des opérateurs de comparaison Python qui ont été tirés de W3School.

| Operator | Name | Example |
|----------|--------------------------|---------|
| == | Equal | x == y |
| != | Not equal | x != y |
| > | Greater than | x > y |
| < | Less than | x < y |
| >= | Greater than or equal to | x >= y |
| <= | Less than or equal to | x <= y |

Exemple: opérateurs de comparaison

```
Imprimer (3 > 2) # true, car 3 est supérieur à 2
imprimer (3 >= 2) # true, car 3 est supérieur à 2
imprimer (3 < 2) # false, car 3 est supérieur à 2
imprimer (2 < 3) # true, car 2 est inférieur à 3
print (2 <= 3) # true, car 2 est moins que 3
print (3 == 2) # false, car 3 n'est pas égal à 2
print (len ('mango') == len ('avocado')) # false
print (len ('mango') != len ('avocado')) # true
print (len ('') < len ('avocado')) # true
print (len ('lait') != len ('viande')) # false
print (len ('lait') == len ('viande')) # true
print (len ('tomato') == len ('Potato')) # true
print (len ('python') > len ('dragon'))
```

Comparer quelque chose donne un vrai ou un faux

```
print ('true == true:', true == true)
print ('true == false:', true == false)
print ('false == false:', false == false)
```

En plus de l'opérateur de comparaison ci-dessus, Python utilise:

- **is**: Renvoie True si les deux variables sont le même objet (x est y)
- **is not**: Renvoie True si les deux variables ne sont pas le même objet (x n'est pas y)
- **in**: Renvoie True si la liste interrogée contient un certain élément (x en y)

•**not in**: Renvoie True si la liste interrogée n'a pas un certain élément (x en y)

```
print('1 is 1', 1 is 1) # True - because the
data values are the same
print('1 is not 2', 1 is not 2) # True - because 1
is not 2
print('A in Asabeneh', 'A' in 'Asabeneh') # True - A found in
the string
print('B in Asabeneh', 'B' in 'Asabeneh') # False - there is
no uppercase B
print('coding' in 'coding for all') # True - because coding
for all has the word coding
print('a in an:', 'a' in 'an') # True
print('4 is 2 ** 2:', 4 is 2 ** 2) # True
```

Opérateurs logiques

Contrairement aux autres langages de programmation, Python utilise des mots clés **and**, **or** et **not** pour les opérateurs logiques. Les opérateurs logiques sont utilisés pour combiner les déclarations conditionnelles:

| Operator | Description | Example |
|----------|---|-----------------------|
| and | Returns True if both statements are true | x < 5 and x < 10 |
| or | Returns True if one of the statements is true | x < 5 or x < 4 |
| not | Reverse the result, returns False if the result is true | not(x < 5 and x < 10) |

```
print (3 > 2 et 4 > 3) # true - car les deux instructions sont vraies
print (3 > 2 et 4 < 3) # false - parce que la deuxième instruction est fausse
print (3 < 2 et 4 < 3) # false - car les deux instructions sont fausses
imprimer ('vrai et vrai:', vrai et vrai)
print (3 > 2 ou 4 > 3) # true - car les deux instructions sont vraies
print (3 > 2 ou 4 < 3) # true - car l'une des instructions est vraie
print (3 < 2 ou 4 < 3) # false - car les deux instructions sont fausses
print ('true ou false:', true ou false) print (pas 3 > 2) # false - car 3 > 2 est vrai, alors pas vrai do
nne false print (pas true) # false - négation, l'opérateur ne fait pas vrai pour false imprimer (pas
false) # true)
```

imprimer (pas pas vrai) # vrai imprimer (pas pas faux) # false

Vous avez une énergie illimitée. Vous venez de terminer les défis du jour 3 et vous avez trois pas en avant pour la grandeur. Maintenant, faites des exercices pour votre cerveau et vos muscles.

Exercices - Jour 3

1. Déclarez votre âge comme variable entière
2. Déclarez votre taille en tant que variable flottante
3. Déclarer une variable qui stockent un numéro complexe
4. Écrivez un script qui invite l'utilisateur à entrer la base et la hauteur du triangle et calculer une zone de ce triangle ($\text{zone} = 0,5 \times b \times h$).

```
Entrez la base: 20
Entrez la hauteur: 10
La zone du triangle est de 100
```

5. Écrivez un script qui invite l'utilisateur à entrer le côté A, le côté B et le côté C du triangle. Calculez le périmètre du triangle ($\text{périmètre} = a + b + c$).

```
Entrez le côté A: 5
Entrez le côté b: 4
Entrez le côté C: 3
Le périmètre du triangle est 12
```

6. Obtenez la longueur et la largeur d'un rectangle à l'aide de l'invite. Calculez sa surface ($\text{surface} = \text{Longueur} \times \text{largeur}$) et périmètre ($\text{périmètre} = 2 \times (\text{longueur} + \text{Largeur})$)
7. Obtenez le rayon d'un cercle à l'aide d'invite. Calculez la zone ($\text{zone} = \pi \times r \times r$) et circonférence ($c = 2 \times \pi \times r$) où $\pi = 3.14$.
8. Calculez la pente, l'ordonnée X et l'ordonnée y de $y = 2x - 2$
9. La pente est ($m = \frac{y_2 - y_1}{x_2 - x_1}$). Trouvez la pente et la distance euclidienne entre le point (2, 2) et le point (6,10)
10. Comparez les pentes dans les tâches 8 et 9.
11. Calculez la valeur de y ($y = x^2 + 6x + 9$). Essayez d'utiliser différentes valeurs X et de déterminer quelle valeur X va être 0.
12. Trouvez la longueur de «Python» et «Dragon» et faites une instruction de comparaison Falsy.
13. Utilisez l'opérateur and pour vérifier si 'o' n' se trouve dans «python» et «dragon»
14. *I hope this course is not full of jargon*. Utilisez l'opérateur in pour vérifier si *jargon* est dans la phrase.
15. Il n'y a pas de 'on' dans Dragon et Python
16. Trouvez la longueur du texte *python* et convertissez la valeur pour flotter et convertir

à la chaîne

17. Même les nombres sont divisibles par 2 et le reste est zéro. Comment vérifiez-vous

k

Si un nombre utilise ou n'utilise pas Python?

18. Vérifiez si la division de plancher de 7 par 3 est égale à la valeur convertie Int de 2,7.
19. Vérifiez si le type de «10» est égal au type de 10
20. Vérifiez si int ('9.8') est égal à 10
21. Écrivez un script qui invite l'utilisateur à saisir les heures et à évaluer par heure. Calculer la rémunération de la personne?

```
Entrez les heures: 40 Entrez le taux par heure: 28
Votre gain hebdomadaire est de 1120
```

22. Écrivez un script qui invite l'utilisateur à saisir le nombre d'années. Calculez le nombre de secondes qu'une personne peut vivre. Supposons qu'une personne puisse vivre cent ans

```
Entrez le nombre d'années que vous avez vécu: 100
Vous vivez pendant 3153600000 secondes.
```

23. Écrivez un script Python qui affiche le tableau suivant

```
1 1 1 1 1 1 2 1 2 4
8 3 1 3 9 27 4 1 4 1
6 64 5 1 5 25 125
```

Félicitations!