

Opérations et instructions conditionnelles

Algorithmique et structuration de données

Léna Gaubert

lana.gaubert@proton.me

L3 TIM/TAL @ INALCO

Précédemment...

- premiers pas en Python,
- variables,
- types de données (numériques, textuelle...),
= opérateurs mathématiques,
- fonctions `print()`, `input()`, et `type()`.

Opérateurs

En Python, les opérateurs permettent de réaliser différents types d'opérations sur des variables. On distingue les types d'opérateurs suivants :

- opérateurs **arithmétiques**
- opérateurs d'**assignation**
- opérateurs d'**incrémentatation**
- opérateur de **comparaison**
- opérateurs **logiques**

Opérations arithmétiques

Opérateur	Dénomination	Effet
+ , -	Addition, soustraction	additionne, soustrait
*	Multiplication	multiplie
/	Division	divise
// , %	Quotient et reste de la division euclidienne	$11//2=5$ $11\%2=1$
**	Puissance	$2^3 = 2 * 2 * 2$

Exercice :

Écrire un programme qui prend en entrée deux nombres, a et b , et qui affiche le résultat des opérations suivantes :

- addition
- soustraction
- multiplication
- division

Opérateurs d'assignation et d'incrémentation

Opérateur	Effet	Exemple
<code>+=</code>	additionne deux valeurs et stocke le résultat	<code>x += 2</code>
<code>-=</code>	soustrait deux valeurs et stocke le résultat	<code>x -= 2</code>
<code>*=</code>	multiplie deux valeurs et stocke le résultat	<code>x *= 2</code>
<code>/=</code>	divise deux valeurs et stocke le résultat	<code>x /= 2</code>

Opérateurs d'assignation et d'incrémentation

Les opérateurs d'incrémentation permettent de simplifier l'écriture de votre code.

```
# méthode 1  
a = 4  
a = a + 2  
  
# méthode 2  
a = 4  
a += 2
```

Exemples

```
a = 4
a += 2
print(a)
a -= 0.3
print(a)
a *= 0.75
print(a)
a /= 2
print(a)
```

Opérateurs de comparaison

Une comparaison renvoie un booléen (`True` ou `False`).

Pour les exemples, $a = 2$.

Opérateur	Dénomination	Exemple
<code>==</code>	égalité	<code>a == 2</code>
<code><</code>	infériorité stricte	<code>a < 3</code>
<code><=</code>	infériorité	<code>a <= 4</code>
<code>></code>	supériorité stricte	<code>a > 1</code>
<code>>=</code>	supériorité	<code>a >= 0</code>
<code>!=</code>	différence	<code>a != 1</code>

Exercice : que renvoient les conditions ci-dessous ?

```
print("1 == 30 :", 1 == 30)
print("1 != 30 :", 1 != 30)
print("1 < 30 :", 1 < 30)
print("1 > 30 :", 1 > 30)
```

Opérateurs logiques

Opérateur	Dénomination	Effet
or	OU logique	vérifie q'une des conditions est vérifiée
and	ET logique	vérifie que toutes les conditions soient vérifiées
not	NON logique	inverse l'état d'une variable booléenne

Les opérateurs logiques permettent de vérifier **plusieurs conditions simultanément**.

On écrit : `cond1 or cond2`, `cond1 and cond2`, `not cond1`...

Exercice : essayer de prévoir le résultat et de l'expliquer.

```
is_sunny = True
```

```
is_warm = True
```

```
is_good_weather = is_sunny and is_warm
```

```
is_rainy_day = not is_sunny
```

```
print(is_good_weather)
```

```
print(is_warm)
```

Structures conditionnelles

- Une **structure conditionnelle** est un ensemble d'instructions qui permettent de tester *si une condition est vraie ou non*.
- Les structures conditionnelles permettent d'exécuter différentes parties de code en fonction de conditions spécifiques.

En Python, la structure conditionnelle s'écrit avec les instructions `if` , `else` , ou encore `elif` .

if-else

```
if condition1:  
    # instruction 1  
else:  
    # instruction 2
```

La condition 1 est *évaluée*.

- Si elle est **VRAIE**, alors l'instruction 1 s'exécute.
- **Sinon** (la condition est **FAUSSE**), alors l'instruction 2 s'exécute.

Exemple

```
age = int(input("How old are you?"))  
if a == 18:  
    print("Too cool for school!")  
else:  
    print("Go back to class!")
```

if-elif-else

```
if condition1:  
    # instruction 1  
elif condition2:  
    # instruction 2  
else:  
    # instruction 3
```

La condition 1 est évaluée.

- Si elle est **VRAIE**, alors l'instruction 1 s'exécute.
- Si elle est **FAUSSE**, la condition 2 est évaluée. Si elle est **VRAIE**, l'instruction 2 s'exécute.
- **Sinon** (la condition 2 est **fausse**), l'instruction 3 s'exécute.

exemple

```
t = int(input("temperature?"))
if t > 30:
    print("Drink water, it's hot!")
elif t > 10 and t <= 20:
    print("Nice day ahead!")
else:
    print("Brrrrrrrr")
```

time to practice! (n.o.o.) > .o