LSIN200A - Cours 2

Coline Gianfrotta

coline.gianfrotta@ens.uvsq.fr

12 février 2026

Cours et TD

Vous trouverez les polycopiés de cours et de TD à l'adresse suivante:

https://github.com/ColineUVSQ/l1-python-in200A

Les structure conditionnelles

Utilisées pour exécuter certaines instructions dans un cas spécifique, et d'autres instructions dans d'autres cas.

Exemple

Si le solde bancaire est insuffisant alors rejeter la transaction.
Sinon continuer.

Les instructions conditionnelles contrôlent quelle partie du code est exécutée en fonctions de certaines conditions.

Structures possibles

```
if
if...  else
if...elif...else
```

L'instruction if

- Jusqu'à présent, les différentes instructions saisies sont toutes exécutées d'une manière linéaire (consécutive)
- Avec les structures conditionnelles, on peut sauter des instructions dans le cas où certaines conditions ne seraient pas satisfaites.

Syntaxe

```
if <expression> :
    instruction1
    instruction2
```

Bloc d'instructions

Une série d'instructions qui s'exécute dans un cas précis.
Un décalage vers la droite permet d'identifier où se trouvent le début et la fin d'un bloc d'instructions: c'est ce qu'on appelle *l'indentation*

L'instruction if

Exemple

```
meteo = "pluie"
if meteo == "pluie":
    # si égalité, alors afficher le message "parapluie"
    print("parapluie")
    # si égalité, alors afficher le message "impermeable"
    print("impermeable")
print("Bonne journée!") # afficher le message "Bonne
journée"
```

- L'opérateur == permet de vérifier si la variable `meteo` est égale à `pluie`. Que renvoie-t-il ?
- Dans quel cas le message "Bonne journée!" s'affiche-t-il ?

L'instruction if-else

Syntaxe

```
if <expression> :
    instruction(s)
else :
    instruction(s)
```

- Le mot-clé `else` ne peut figurer sans instruction `if`
- Le mot-clé `else` doit être au même niveau d'indentation que l'instruction `if` qu'elle complète
- Le code du bloc `else` s'exécute si le code du bloc `if` ne s'exécute pas.
- `else` doit également être suivi de deux points (:)

L'instruction if - else

Exemple

```
meteo = "froid"
if meteo == "pluie":
    print("parapluie")
    print("impermeable")
else:
    print("t-shirt")
    print("shorts")
print("Bonne journée!")
```

Que va afficher le programme suivant ?

L'instruction elif

Syntaxe

```
if <expression> :
     instruction(s)
elif <expression> :
     instruction(s)
else :
     instruction(s)
```

- Le mot-clé `elif` doit être au même niveau d'indentation que l'instruction `if` qu'elle complète
- `elif` doit également être suivi d'une condition et de deux points (:)
- L'instruction `else` ne peut figurer qu'une seule fois en clotûre du bloc de la condition `if`

L'instruction elif

Exemple

```
meteo = "froid"
if meteo == "pluie":
     print("parapluie")
     print("impermeable")
elif meteo == "froid" :
     print("pull")
     print("echarpe")
else:
     print("t-shirt")
     print("shorts")
print("Bonne journée!")
```

- Il est possible de mettre autant de `elif` qu'on le souhaite après une condition `if`.
- Les instructions `elif` et `else` sont facultatives: lorsqu'une instruction `if` ou `elif` est définie, il n'est pas obligatoire de prévoir un `else` après

Chercher l'erreur

Chercher l'erreur dans le code suivant:

```
meteo = "froid"
if meteo == "froid":
     print("echarpe")
print("Bonne journée!")
elif meteo == "froid" or meteo == "venteux" :
     print("manteau")
     print("Bonne journée!")
else:
     print("t-shirt")
print("Bonne journée!")
```

Les structures conditionnelles et les opérateurs de comparaison

Les opérateurs de comparaison retournent `True` ou `False`.
Nous pouvons donc les utiliser dans des instructions conditionnelles.

Opérateur	Signification
<	strictement inférieur à
>	strictement supérieur à
<=	inférieur ou égal à
>=	supérieur ou égal à
==	égal à
!=	différent de

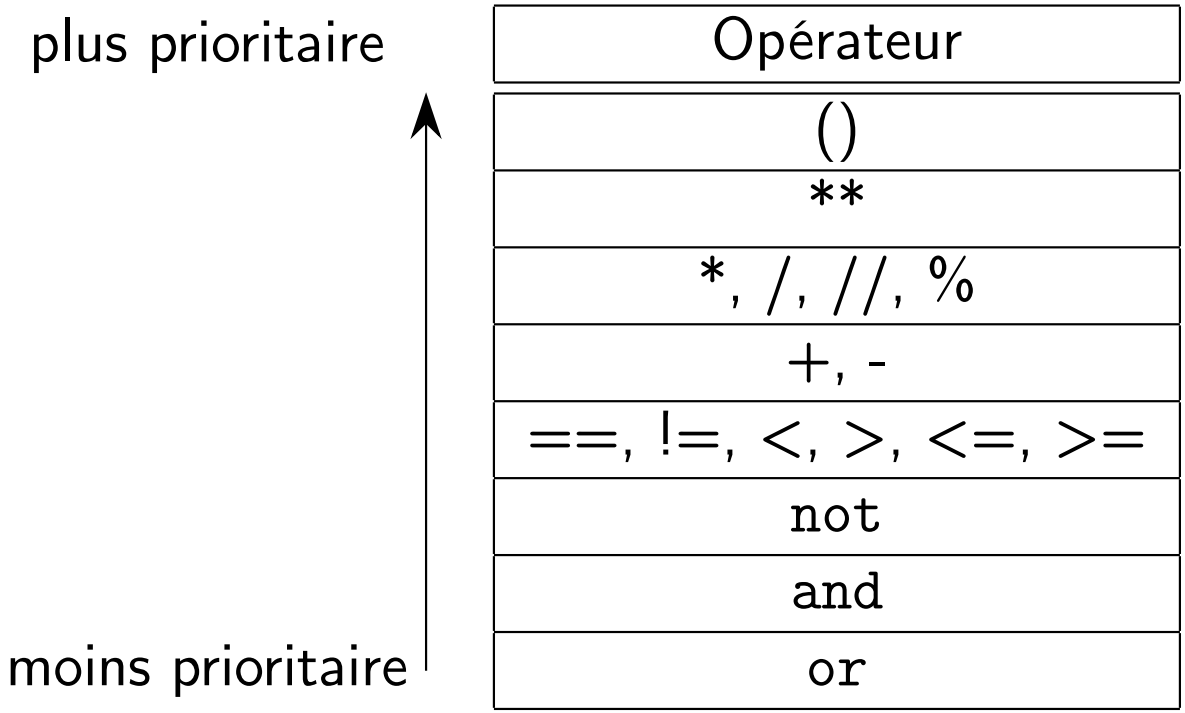
Exemple: `a >= b` renvoie `True` si a est supérieur ou égal à b, `False` sinon.

Exemple

```
solde = 20.0
prix_achat_hors_taxe = 19.0
taxe = 1.08
# comparer le solde bancaire du client au prix d'achat TTC
if solde >= prix_achat_hors_taxe * taxe:
    print("Transaction possible")
else:
    print("Transaction impossible")
print("Fin")
```

Remarque: la multiplication se fait avant la comparaison, il n'y a pas besoin d'ajouter de parenthèses

Priorité des opérateurs



Que valent les expressions suivantes ?

```
a,b = 3,2
not a > 0 and 5*b <= 3
not (a > 0 and 5*b <= 3)
```

Evaluation paresseuse des expressions avec and et or

Les expressions qui n'ont pas besoin d'être évaluées pour déterminer le résultat ne sont pas évaluées.

x and y

On évalue d'abord x.
Si x vaut `False`, sa valeur est renvoyée et y n'est pas évaluée.
Sinon, la variable y est évaluée et la valeur résultante est renvoyée.

x or y

On évalue d'abord x.
Si x vaut `True`, sa valeur est renvoyée et y n'est pas évaluée.
Sinon, la variable y est évaluée et la valeur résultante est renvoyée.

Exemples avec and

```
x = 0
if x != 0 and 2 // x == 2 :
    print("Test de l'évaluation paressseuse")
print("Fin")
```

```
x = 0
if 2 // x == 2 and x != 0 :
    print("Test de l'évaluation paressseuse")
print("Fin")
```

Exemples avec or

```
x = 0
if x == 0 or 2 // x == 2 :
     print("Test de l'évaluation paressseuse")
print("Fin")
```

```
x = 0
if 2 // x == 2 or x == 0 :
     print("Test de l'évaluation paressseuse")
print("Fin")
```

Les opérateurs and et or ne sont pas symétriques: l'ordre des expressions à tester a de l'importance.

La portée des variables

On appelle *portée d'une variable* (ou scope) l'ensemble des endroits du programme où elle existe.

Exemple

```
num1 = 1
num2 = 2
resultat = "resultat non connu"
if num1 < num2:
     resultat = "num2 est plus grand que num1"
print(resultat)
print("Fin")
```

- La portée de la variable `resultat` est de la ligne 3 jusqu'à ce que le programme se termine.
- La ligne 5 est dans la portée de la variable `resultat`.

La portée des variables

Exemple

```
num1 = 1
num2 = 2
if num1 < num2:
      max = num2
print(max)
print("Fin")
```

En Python, la portée d'une variable commence lorsqu'elle est créée et se termine lorsqu'une terminaison se produit (exemple: la fin du programme, l'utilisation de fonctions,...).

Instructions if imbriqués

On peut avoir des instructions `if` à l'intérieur d'un bloc d'instructions `if`.

Syntaxe

```
if <expression>:
    instruction(s)
    if <expression>:
        instruction(s)
```

Exemple

```
num1, num2 = 2, 5
max = num1
if num1 < num2:
    max = num2
    num3 = 3
    if num2 < num3:
        max = num3
print(max)
```

Quelle est la portée de la variable num3 ?
Qu'affiche-t-on dans ce programme ?

Exemple pour finir: qu'affiche ce programme ?

```
solde = 21.0
taxe = 1.08
nom_detenteur_carte = "Daniel Dupont"
vendeurs_de_confiance = ["Maria", "Yoann", "Emilie", "Tia"]
prix_achat_hors_taxes = 19.0
nom_client = "Daniel Dupont"
vendeur = "Fred"
decouvert_autorise = True
if (solde <= prix_achat_hors_taxes * taxe and not decouvert_autorise):
      print("Transaction refusée solde insuff.  ou pas de découvert
autorisé)
else:
      if not nom_detenteur_carte == nom_client:
            print("Transaction non approuvée:  client non valide")
      else:
            if not vendeur in vendeurs_de_confiance:
                  print("Transaction refusée:vendeur non autorisé")
            else:
                  print("Transaction approuvée")
```