Compléments sur les booléens, les conditions et les opérateurs logiques

Table des matières

1. Le type booléen (bool)....1
a. Définition....1
b. Exemples :....1
2. Les comparaisons (renvoient un booléen)....1
3. Structure conditionnelle if....2
a. Syntaxe :....2
b. Exemple :....2
4. Opérateurs logiques....2
a. Exemple :....2
5. Précision : Priorité des opérateurs logiques....2
6. Conseil pour maintenir un code lisible....2
7. Exemples pratiques....3
a. Vérifier une valeur comprise dans un intervalle :....3
b. Mot de passe correct ?....3
8. Résumé rapide....3
9. Exercices....3

1. Le type booléen (`bool`)

a. Définition

Un **booléen** est une valeur qui ne peut être que :

- `True` (vrai)
- `False` (faux)

b. Exemples :

```
vrai = True
faux = False

print(type(vrai))  # <class 'bool'>
```

2. Les comparaisons (renvoient un booléen)

Opérateur	Signification	Exemple	Résultat
`==`	Égal à	`5 == 5`	`True`
`!=`	Différent de	`5 != 3`	`True`
`<`	Strictement inférieur	`3 < 5`	`True`
`<=`	Inférieur ou égal	`3 <= 3`	`True`
`>`	Strictement supérieur	`6 > 4`	`True`
`>=`	Supérieur ou égal	`6 >= 6`	`True`

3. Structure conditionnelle `if`

a. Syntaxe :

```
if condition:
    instructions_si_vrai
elif autre_condition:
    instructions_si_autre_vrai
else:
    instructions_si_aucune
```

b. Exemple :

```
age = 18
if age < 18:
    print("Mineur")
elif age == 18:
    print("Tout juste majeur")
else:
    print("Majeur")
```

4. Opérateurs logiques

Opérateur	Nom	Utilité
`and`	ET logique	Vrai si les **deux** conditions sont vraies
`or`	OU logique	Vrai si **au moins une** des conditions est vraie
`not`	NON logique	Inverse la valeur de vérité

a. Exemple :

```
age = 20
carte = True

if age >= 18 and carte == True:
    print("Accès autorisé")
```

5. Précision : Priorité des opérateurs logiques

Priorité	Opérateur
1er	`not`
2e	`and`
3e	`or`

Tu peux utiliser des **parenthèses** pour forcer l'ordre :

```
if (age > 18 or carte) and not interdit:
```

6. Conseil pour maintenir un code lisible

Pour la clarté du code, les parenthèses sont conseillées ; de même que l'écriture sur plusieurs lignes pour les conditions composées complexes. Exemple :

```
    if      (a >25 and  b<15 and c>10 and d >0 ) \
         or (a >35 and  b<25 and c<10 and d >0 ) \
         or (a >55 and  b<15 and c>10 and d <0 ) \
```

7. Exemples pratiques

a. Vérifier une valeur comprise dans un intervalle :

```
temperature = 25
if 20 <= temperature <= 30:      # écriture pratique mais propre à python !
    print("Température agréable")
```

b. Mot de passe correct ?

```
mdp = input("Mot de passe : ")
if mdp == "secret123":
    print("Accès autorisé")
else:
    print("Mot de passe incorrect")
```

8. Résumé rapide

Élément	Exemples
Booléens	`True`, `False`
Comparaisons	`==`, `!=`, `<`, `<=`, `>`
Conditions	`if`, `elif`, `else`
Opérateurs logiques	`and`, `or`, `not`
Usage des parenthèses	`(cond1 and cond2) or not cond3`

9. Exercices

On souhaite autoriser l'accès à un site à trois personnes. Pour entrer chacune de ces personnes devra fournir un mot de passe personnalisé.

Nom de l'utilisateur/mot_de_passe : philippe/marmotte2022 nicolas/bubulle2018 david/grisouille20

Ecrire un programme qui demande le nom et le mot de passe de l'utilisateur et qui répond « accés autorisé » ou « accès refusé » en fonction des réponses obtenues.

Pour aller plus loin : ajouter deux variables heures et minutes au programme. Modifier le programme pour que l'accès soit autorisé pour philippe entre 8h15 et 18h30 ; pour nicolas tous le temps, pour david entre 10h et 20h .

Pour aller plus loin : comment faire si on souhaite que david ne soit autorisé à entrer que si nicolas est déjà entré ?