

■ Cours & Exercices — Les conditions en Python

■ Notions essentielles

Les **conditions** permettent d'exécuter du code seulement si une **expression booléenne** est vraie. On combine **comparaisons** (`==` `!=` `<` `<=` `>` `>=`) et **logique** (`and`, `or`, `not`) et on valide avec **print()**. Ici : aucune boucle ni fonction.

■ Booléens et comparaisons

Une comparaison renvoie True ou False.

```
print(5 > 2) # True print(3 == 4) # False
```

True False

■ Logique : and, or, not

Priorité : **not** > **and** > **or**. Utilise des parenthèses pour être explicite.

A	B	A and B	A or B	not A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

```
age = 17; etudiant = True print((age < 18) or etudiant) # True
```

True

■ Structures if / elif / else

Le premier test vrai est exécuté, les suivants sont ignorés.

```
note = 15 if note >= 16: print("Très bien") elif note >= 12:  
print("Bien") else: print("À renforcer")
```

Bien

■ Comparaisons enchaînées

Python autorise les intervalles lisibles : $0 \leq x < 10$ équivaut à $(x \geq 0) \text{ and } (x < 10)$.

```
x = 7 print(0 <= x < 10) # True
```

True

■ Expression conditionnelle (ternaire)

Syntaxe : `val_si_vrai if condition else val_si_faux` (idéal pour affecter une variable).

```
age = 20 message = "Mineur" if age < 18 else "Majeur" print(message)
```

Majeur

■ Chaînes & sens des comparaisons

Les comparaisons de chaînes sont *lexicographiques* (ordre alphabétique, sensible à la casse).

```
print("Z" > "a") # False (car "Z" < "a" en Unicode) print("abc" < "abd")  
# True
```

False True

■ Pièges classiques

- **Égalité vs identité** : `==` compare les valeurs, `is` compare les objets (éviter `is` pour les nombres/chaînes).
- **Floats** : éviter `==` direct après calcul (arrondir ou tester un écart).
- **Courte-circuit** : `and/or` n'évaluent pas la 2^e partie si inutile.

■ Bonnes pratiques

- **Nommer** des booléens parlants (ex : `a_droit_reduit = (age < 18) or etudiant`).
- **Parenthèses** pour clarifier. **`elif`** plutôt que **`if`** imbriqués. **`print()`** explicite.

■ À retenir

- Combine **comparaisons** et **and/or/not** pour des tests clairs.
- Préfère **`elif`** pour des cas exclusifs + **`else`** pour le défaut.
- Utilise **parenthèses** et **f-strings** avec `print()` pour des messages lisibles.

■ Exercices d'application (corrigés)

■ Exercice 1 — Éligibilité vote (input)

Demande à l'utilisateur son âge (entier). Affiche « Tu peux voter » si $\text{age} \geq 18$, sinon « Tu ne peux pas voter ». ■■ Utilise `int(input(...))` et un `if/else`.

■ Correction :

```
age = int(input("Âge : ")) if age >= 18: print("Tu peux voter") else:
print("Tu ne peux pas voter")
```

Exemple (entrée 19) → Tu peux voter

■ Exercice 2 — Livraison gratuite (input)

Saisis montant (float) et pays (str, ex : FR). Livraison gratuite si $\text{montant} \geq 100$ **et** $\text{pays} == \text{"FR"}$. Sinon frais: 4,90€. Affiche un message clair avec `print()`.

■ Correction :

```
montant = float(input("Montant € : ")) pays = input("Pays (ex: FR) : ")
if (montant >= 100) and (pays == "FR"): print("Livraison gratuite")
else: print("Frais de port: 4,90€")
```

Exemple (120, "FR") → Livraison gratuite

■ Exercice 3 — Catégorie IMC (sans input)

Avec `imc` (float) déjà défini, affiche : < 18.5 « Insuffisance pondérale », $18.5\text{--}24.9$ « Normal », $25\text{--}29.9$ « Surpoids », ≥ 30 « Obésité ». Utilise `elif` et des **comparaisons enchaînées**.

■ Correction :

```
imc = 26.1 if imc < 18.5: print("Insuffisance pondérale") elif 18.5 <=
imc < 25: print("Normal") elif 25 <= imc < 30: print("Surpoids") else:
print("Obésité")
```

Surpoids

■ Exercice 4 — Total TTC avec remise (input + ternaire)

Demande montant (float) et `code_promo` (str « oui/non »). Remise 10% si « oui », sinon 0%. Calcule TTC (TVA 20%) et affiche Total TTC: X € arrondi à 2 décimales. Utilise un **ternaire** pour la remise.

■ Correction :

```
montant = float(input("Montant HT : ")) code_promo = input("Code promo
(oui/non) : ").lower() remise = 0.10 if code_promo in ["oui", "o",
"yes"] else 0.0 ht_remise = montant * (1 - remise) ttc = ht_remise *
1.20 print("Total TTC:", round(ttc, 2), "€")
```

Exemple (100, oui) → Total TTC: 108.0 €

■ Exercice 5 — Accès offert (input)

Saisis age (int) et etudiant (str « oui/non »). Affiche « Accès gratuit » si age < 12 **ou** si etudiant vaut « oui ». Sinon « Accès payant ».

■ Correction :

```
age = int(input("Âge : ")) etu = input("Étudiant (oui/non) : ").lower()
if (age < 12) or (etu in ["oui", "o", "yes"]): print("Accès gratuit")
else: print("Accès payant")
```

Exemple (10, non) → Accès gratuit

■ Exercice 6 — Max de 3 nombres (sans max())

Avec a, b, c donnés, affiche le plus grand (sans boucles ni fonctions).

■ Correction :

```
a, b, c = 12, 7, 19 m = a if b > m: m = b if c > m: m = c print("Le plus
grand est:", m)
```

Le plus grand est: 19

■ Exercice 7 — Triangle valide (input)

Demande trois longueurs a, b, c (floats). Affiche « Triangle valide » si chaque côté est strictement inférieur à la somme des deux autres, sinon « Triangle invalide ».

■ Correction :

```
a = float(input("a : ")) b = float(input("b : ")) c = float(input("c :
")) if (a < b + c) and (b < a + c) and (c < a + b): print("Triangle
valide") else: print("Triangle invalide")
```

Exemple (3, 4, 5) → Triangle valide