

Semaine 2 : opérateurs et conditions

Initiation à l'algorithmique et à la programmation

Revekka Kyriakoglou

28 septembre 2026

Aujourd'hui

- 1 Révision et objectifs
- 2 Expressions et opérateurs
- 3 Décider avec if
- 4 Déboguer et s'entraîner

De la semaine 1 à la semaine 2

La semaine dernière : variables, types, calculs, `input()`, `print()` et erreurs.

```
mot = input("Entrez un mot : ")  
print("Vous avez saisi :", mot)
```

Aujourd'hui : faire prendre une décision au programme selon la valeur saisie.

À la fin du cours, vous saurez...

- calculer une expression avec les opérateurs usuels ;
- distinguer = et == ;
- lire une condition qui renvoie True ou False ;
- suivre les branches de if, elif, else ;
- écrire et déboguer un petit programme conditionnel.

Calculer avec Python

+	addition	-	soustraction
*	multiplication	/	division (float)
//	quotient entier	%	reste
**	puissance		

```
print(7 / 2) # 3.5  
print(7 // 2) # 3  
print(7 % 2) # 1
```

Avec des nombres négatifs, // arrondit vers le bas : `-7 // 2` vaut `-4`.

Affectation et mise à jour

```
compteur = 2 # affectation
compteur += 3 # comme compteur = compteur + 3
print(compteur) # 5
compteur -= 1 # 4
```

`+=` et `-=` modifient la valeur associée à la variable. En Python, `compteur++` n'existe pas : on écrit `compteur += 1`.

Dans quel ordre ?

Parenthèses, puissance, multiplication et division, puis addition et soustraction. À priorité égale, la plupart des opérations se lisent de gauche à droite.

```
print((2 + 3) * 2 + 3) # 13  
print(2 + 3 * 2) # 8
```

À prévoir : $10 - 6 / 2$; $(10 - 6) / 2$.

Comparer : le résultat est un booléen

==	égal à	!=	différent de
<	inférieur à	<=	inférieur ou égal à
>	supérieur à	>=	supérieur ou égal à

```
mot = "chat"  
print(mot == "chat") # True  
print(mot == "Chat") # False
```

= affecte une valeur ; == pose une question. La casse des chaînes compte.

Combiner des conditions

```
age = 20
inscrit = True
print(age >= 18 and inscrit) # True
print(age < 18 or inscrit) # True
print(not inscrit) # False
```

and : les deux conditions ; or : au moins une ; not : inversion.
Utilisez des parenthèses pour rendre les tests composés lisibles.

Anticiper avant d'exécuter

```
mot = "bonjour"  
longueur = len(mot)  
print(longueur >= 5 and mot == "bonjour")  
print(longueur < 5 or mot == "salut")
```

Question : quelles sont les deux lignes affichées ? `len(mot)` donne le nombre de caractères du mot.

Premier choix : if

```
mot = input("Entrez un mot : ")  
if mot == "bonjour":  
    print("Salutation reconnue")  
print("Fin du programme")
```

La ligne indentée s'exécute uniquement si la condition est vraie. La dernière ligne s'exécute toujours. Les deux-points et l'indentation sont nécessaires.

Deux issues : if / else

```
mot = input("Entrez un mot : ")  
if mot == "bonjour":  
    print("Salutation reconnue")  
else:  
    print("Autre mot")
```

Une seule des deux branches est exécutée. **À prévoir** : que se passe-t-il pour bonjour ? Et pour Bonsoir ?

Plusieurs cas : if / elif / else

```
mot = input("Entrez un mot : ")  
if mot == "bonjour":  
    print("Salutation du jour")  
elif mot == "bonsoir":  
    print("Salutation du soir")  
else:  
    print("Autre mot")
```

Les tests sont examinés dans l'ordre. Dès qu'une branche correspond, les suivantes sont ignorées. else est facultatif.

Indentation : quelles lignes appartiennent au if ?

```
mot = "chat"
if mot == "chat":
    print("Animal reconnu")
    print("Quatre lettres")
print("Termin")
```

Toutes les lignes d'un bloc ont le même retrait (en pratique : quatre espaces). **À prévoir** : quelles lignes apparaissent si `mot = "chien"` ?

Convertir input avant de comparer des nombres

```
longueur = int(input("Nombre de lettres : "))  
if longueur >= 6:  
    print("Mot long")  
else:  
    print("Mot court")
```

input() fournit une chaîne. int() permet ici une comparaison numérique. Une saisie non numérique provoque ValueError : nous apprendrons plus tard à la traiter.

Conditions combinées : intervalle

```
longueur = int(input("Nombre de lettres : "))
if longueur >= 4 and longueur <= 7:
    print("Entre 4 et 7 lettres")
else:
    print("En dehors de l'intervalle")
```

À prévoir : sorties pour 3, 4, 7 et 8. On peut aussi écrire `4 <= longueur <= 7`.

Deux if ou un if / elif ?

```
n = 8
if n > 0:
    print("positif")
if n % 2 == 0:
    print("pair")
```

Deux if indépendants peuvent tous les deux s'exécuter. Avec elif, une seule branche de la chaîne est retenue.

Conditions imbriquées

```
mot = "chat"
if len(mot) >= 4:
    if mot == "chat":
        print("Le mot est chat")
    else:
        print("Mot de quatre lettres ou plus")
else:
    print("Mot court")
```

L'indentation indique à quel if appartient chaque else. À prévoir : sorties pour chat, chien, rat.

Trouver trois erreurs

```
mot = input("Un mot : ")  
if mot = "banane"  
    printf("Le mot est banane")  
else:  
    print("Autre mot")
```

Méthode : identifiez l'erreur ; proposez une correction ; testez avec banane puis poire.

Avant le TP

Pour chaque programme : **prédire**, **exécuter**, **expliquer**. Si le résultat surprend, afficher les variables intermédiaires et relire l'indentation. Conservez votre code dans des fichiers `.py`.