Conditionals in Python

1. Core Practice Exercises

1.1 Predict the Output

For each, predict True or False (or the actual printed result) before running:

```
5 < 5 or 5 == 5

10 >= 2 and not 3 > 5

'cat' < 'catalog'

'Hi' and ''          # ??

bool('False')        # ??
```

1.2 Hot or Mild

Ask the user for an integer temperature and print "Hot" if ≥ 30, otherwise print "Mild".

1.3 Mixed Comparison

```
a = 0

print(10 > a != False)        # Explain why this prints True or False
```

---

2. Eercises

e 1 – Character Classifier

Ask for one character. Print whether it is an **Uppercase letter**, **Lowercase letter**, **Digit**, or **Other**.

e 2 – Password Strength Hints

Ask for a password string.

- If it contains a space → print "No spaces allowed"
- If its length < 8 → print "Too short"

- Otherwise → print "Length OK"

e 3 – Substring Finder

Ask for two strings text and pattern. Print whether the pattern occurs inside the text.

e 4 – Range Reporter

Ask the user for a number 0-100 and print which decade bucket (0-9, 10-19, …, 90-100) it falls in.

---

3. Challenges

Challenge 1 – Second-Degree Equation Solver

Solve ax2+bx+c=0$ax^2 + bx + c = 0$ for user-supplied a, b, c. Print the real roots (two, one, or none) or that it is not quadratic.

Challenge 2 – Triangle Classifier

Ask for three side lengths. Check validity and print whether the triangle is **Equilateral**, **Isosceles**, **Scalene**, or invalid.

Challenge 3 – Leap-Year Detector

Ask for a year and print whether it is leap.

Challenge 4– Tax-Bracket Calculator

Ask for income and print the applicable tax-rate bracket.

Challenge 5 – Coordinate Quadrant

Ask for x and y and print which quadrant the point lies in, or whether it is on an axis or at the origin.

Challenge 6 – Letter Gradebook

Ask for a numeric mark (0-100) and print the letter grade (A/B/C/D/F).

Challenge 7 – Simple Calculator

Ask for two numbers and an operator symbol (+ − * /). Perform calculation or print an error.

Challenge 8 – Even–Odd–Zero Classifier

Ask for an integer. Print "Zero" if 0, "Even" if divisible by 2, otherwise "Odd".

---

4. A real case – ISO Date Validator + Day-of-Year Calculator

Task

Write a program that:

1. Reads a date string in the exact format YYYY-MM-DD.
2. Validates:
   - Year is 1…9999.
   - Month is 1…12.
   - Day is valid for that month (leap-year aware).
   - The string contains exactly two dashes at positions 4 and 7 and all other characters are digits.
3. If invalid, print a clear message: Invalid format / Invalid year / Invalid month / Invalid day.
4. If valid, print:
   - Valid: YYYY-MM-DD
   - Leap year: Yes/No
   - Day of year: N

Constraints:

- No loops, no functions, no imports.
- Use only string slicing, membership checks, int(), arithmetic, if / elif / else.

Hints

- Use date_str[4] and date_str[7] to check separators.
- Use .isdigit() to check numeric parts.
- Use leap-year test: (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0).
- Compute day-of-year by cumulative month lengths plus an extra day after February if leap.

Example Valid Inputs

- 2024-02-29 → leap year; day 60.
- 2023-03-01 → non-leap; day 60.
- 1999-12-31 → valid; day 365.

Example Invalid Inputs

- 20240229 (no dashes)
- 2024/02/29 (wrong separators)
- 2021-02-29 (non-leap Feb 29)
- 2024-04-31 (April has 30)