✅ 1. Comprehensions in Python

Python comprehensions are concise ways to create **lists, dictionaries, sets**, or **generators**.

1.1 List Comprehension

Used to create lists in **one readable line**.

Syntax

```
[expression for item in iterable if condition]
```

Example

```
squares = [x*x for x in range(1, 6)]
# [1, 4, 9, 16, 25]
```

With conditions

```
even_nums = [x for x in range(10) if x % 2 == 0]
```

With if-else

```
results = ["even" if x % 2 == 0 else "odd" for x in range(5)]
```

1.2 Dictionary Comprehension

```
{key_expression: value_expression for item in iterable}
```

Example

```
num_map = {x: x*x for x in range(1, 6)}
# {1:1, 2:4, 3:9, ...}
```

With condition:

```
even_map = {x: x*x for x in range(10) if x % 2 == 0}
```

1.3 Set Comprehension

Same as list comprehension but with `{ }`.

```
unique_squares = {x*x for x in range(10)}
```

1.4 Generator Comprehension

Uses `( )` instead of `[ ]`. Does **not** create the full list in memory → uses **lazy evaluation**.

```
gen = (x*x for x in range(10))
```

Unpacking Tricks

5.1 Unpacking sequences

```
a, b, c = [1, 2, 3]
```

5.2 Extended unpacking

```
a, *middle, c = [1,2,3,4,5]
# a=1, middle=[2,3,4], c=5
```

✅ `for…else` and `while…else` in Python

These `else` blocks *do not work like if-else*.
Instead:

👉 The `else` block runs ONLY if the loop completes normally (no `break`).

Meaning:

- If the loop **breaks early**, the `else` block **does NOT run**.
- If the loop **finishes all iterations**, the `else` **WILL run**.

🔵 1. `for…else`

Best used for **searching**.

✔ Example: Find a number in list

```
nums = [1, 3, 5, 7, 9]

target = 6

for n in nums:
    if n == target:
        print("Found!")
        break
else:
    print("Not found")
```

Explanation

- If number is found → `break` → **else won't execute**
- If loop completes without break → **else runs** → prints "Not found"

🔥 Real-world use case: retry until success

Example: Try password 3 times

```
attempts = 0
correct_pass = "1234"

while attempts < 3:
    pwd = input("Enter password: ")
    if pwd == correct_pass:
        print("Login successful!")
        break
    attempts += 1
else:
    print("Too many attempts. Locked!")
```