**Ternary Operator in Python**

**1️⃣ What is a Ternary Operator?**

* It is a **one-line shortcut** for writing an if-else statement.
* Also called a **Conditional Expression**.

**2️⃣ Syntax**

python

CopyEdit

value\_if\_true if condition else value\_if\_false

**3️⃣ Example (Basic)**

Instead of:

python

CopyEdit

x = 10

if x > 5:

result = "Greater"

else:

result = "Smaller"

We can write:

python

CopyEdit

x = 10

result = "Greater" if x > 5 else "Smaller"

print(result) # Output: Greater

**4️⃣ Nested Ternary (Multiple Conditions)**

python

CopyEdit

x = 7

result = "Positive" if x > 0 else "Zero" if x == 0 else "Negative"

print(result) # Output: Positive

⚠️ **Note:** Nested ternary can be hard to read. Use only if it’s simple.

**5️⃣ Use Cases**

* Setting default values
* Quick conditional assignments
* Returning values in a single line

**6️⃣ Example with Function Return**

python

CopyEdit

def check\_even\_odd(n: int) -> str:

return "Even" if n % 2 == 0 else "Odd"

print(check\_even\_odd(5)) # Output: Odd

**7️⃣ Difference from if-else**

* **Ternary:** used when you need a quick single-line expression.
* **If-else:** better when you have multiple lines of logic or complex conditions.

**8️⃣ Quick Example in List Comprehension**

python

CopyEdit

nums = [1, 2, 3, 4, 5]

labels = ["Even" if n % 2 == 0 else "Odd" for n in nums]

print(labels) # Output: ['Odd', 'Even', 'Odd', 'Even', 'Odd']

**📝 Key Points**

1. Only use for **simple conditions**.
2. Improves readability for short checks.
3. Avoid complex nesting.