**1. What is the Walrus Operator?**

* Introduced in **Python 3.8**, the := is called the **Walrus Operator**.
* It allows you to **assign a value to a variable as part of an expression**.
* This avoids writing the same expression twice.

**2. Syntax**

python

CopyEdit

variable := expression

* The expression is evaluated, assigned to the variable, and returned at the same time.

**3. Basic Example (without Walrus)**

python

CopyEdit

value = input("Enter something: ")

while value != "quit":

print(f"You typed: {value}")

value = input("Enter something: ")

**With Walrus:**

python

CopyEdit

while (value := input("Enter something: ")) != "quit":

print(f"You typed: {value}")

👉 We don't have to call input() twice.  
The variable value is both **assigned** and **used** inside the condition.

**4. Why Use It?**

* **Reduces redundancy:** Avoids recalculating or re-calling functions.
* **Cleaner code:** Fewer lines and better readability (when used carefully).

**5. Examples**

**a) While Loops**

python

CopyEdit

# Without Walrus

line = input("Enter text: ")

while line:

print(line)

line = input("Enter text: ")

# With Walrus

while (line := input("Enter text: ")):

print(line)

**b) If Statements**

python

CopyEdit

if (n := len("Python")) > 3:

print(f"Length is {n}")

**Output:**

csharp

CopyEdit

Length is 6

👉 The variable n stores the length and is used in the condition.

**c) List Comprehensions**

python

CopyEdit

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

squares = [y for x in numbers if (y := x\*\*2) > 5]

print(squares)

**Output:**

csharp

CopyEdit

[9, 16, 25]

👉 y := x\*\*2 assigns the square to y and checks if it's greater than 5.

**6. Important Notes**

1. Works in **Python 3.8+ only**.
2. Should not be overused; can make code harder to read if nested too deeply.
3. **Scope:** The variable remains in scope after the expression.

**7. Real-World Example**

**Reading lines from a file**

python

CopyEdit

with open("data.txt") as file:

while (line := file.readline().strip()):

print(line)

👉 Reads each line until an empty string is encountered (end of file).

**8. Key Difference Between = and :=**

* = is a **statement** → only assigns.
* := is an **expression** → assigns and returns the value in one step.

**Example**

python

CopyEdit

# Normal assignment

x = 5

print(x) # 5

# Walrus operator

print(y := 10) # Assigns and prints 10 in one step