WHILE LOOP

WHILE

- When a `while` loop is encountered, Python checks the condition.
  - If the condition is `True`, the code block inside the loop is executed.
- After each execution of the code block, Python re-evaluates the condition.
  - If the condition remains `True`, the loop continues;
  - if it becomes `False`, the loop terminates, and the program moves on to the next statement after the loop.

LUT University

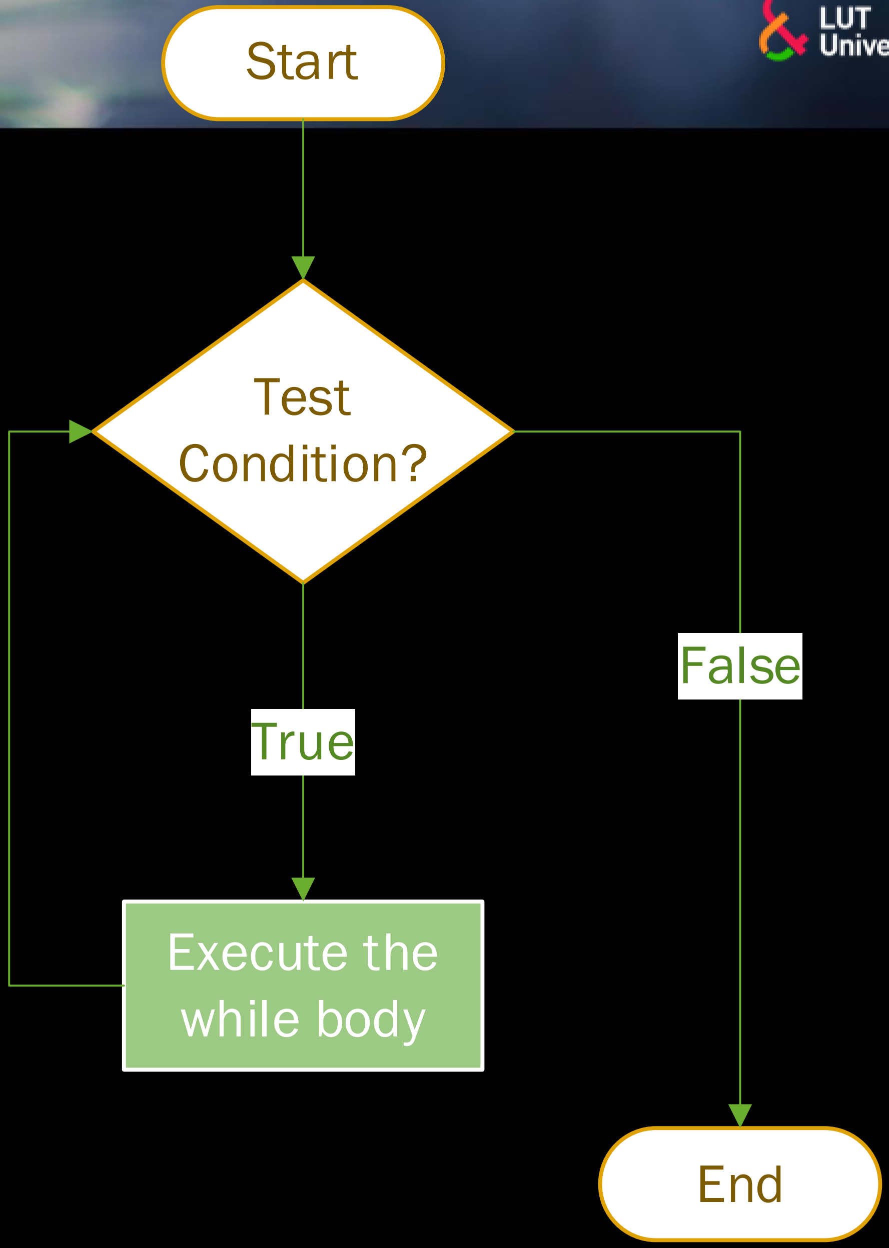
WHILE LOOP

A **while** loop is executed as follows:

- The loop starts & immediately checks if required conditions are met.
- The code block is then processed, after which the code checks for the condition again.

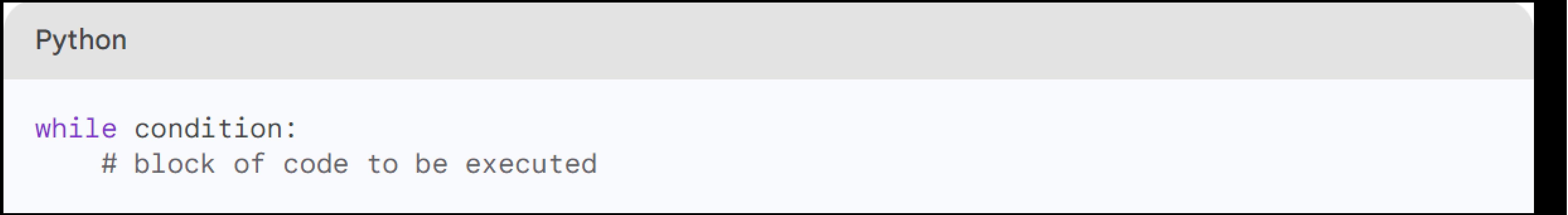
...

- Once the required condition is no longer true, it will stop running the code block.



WHILE LOOP

- The syntax for a while loop is as follows:



Python

```python
while condition:
    # block of code to be executed
```

- The condition is a Boolean expression that is evaluated before each iteration of the loop.
- If the condition is True, the block of code is executed.
- If the condition is False, the loop terminates.

WHILE ITERATION STRUCTURE

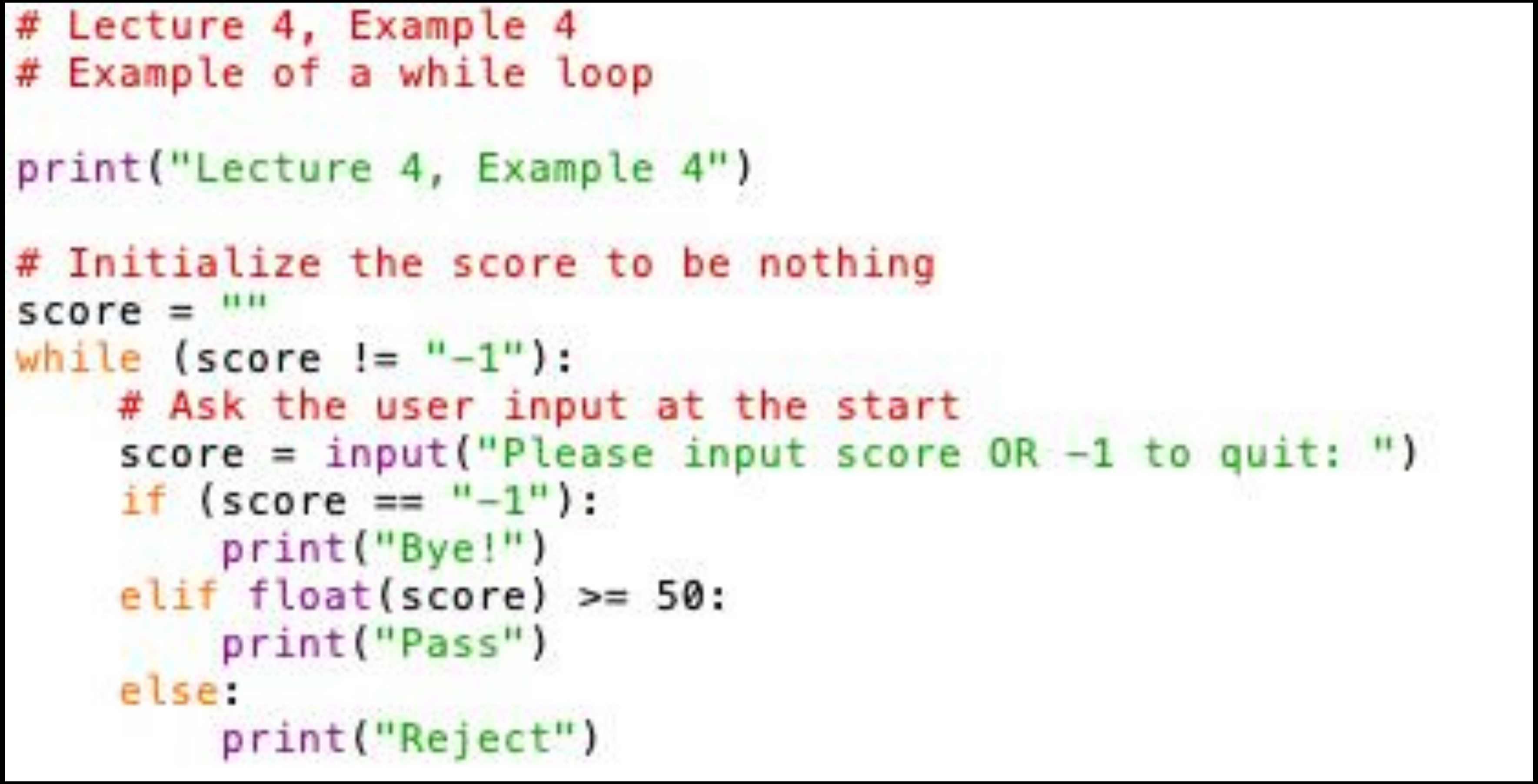
- Repetition based on user input

```
while <expression>:
   <do something>
```

Example:
Input score, print corresponding assessment result (Pass or Failure), until the user type “quit”

Score	result
Score>=50	Pass
Score<50	Failure

WHILE EXAMPLE



```
# Lecture 4, Example 4
# Example of a while loop

print("Lecture 4, Example 4")

# Initialize the score to be nothing
score = ""
while (score != "-1"):
    # Ask the user input at the start
    score = input("Please input score OR -1 to quit: ")
    if (score == "-1"):
        print("Bye!")
    elif float(score) >= 50:
        print("Pass")
    else:
        print("Reject")
```