

CTRL

FN



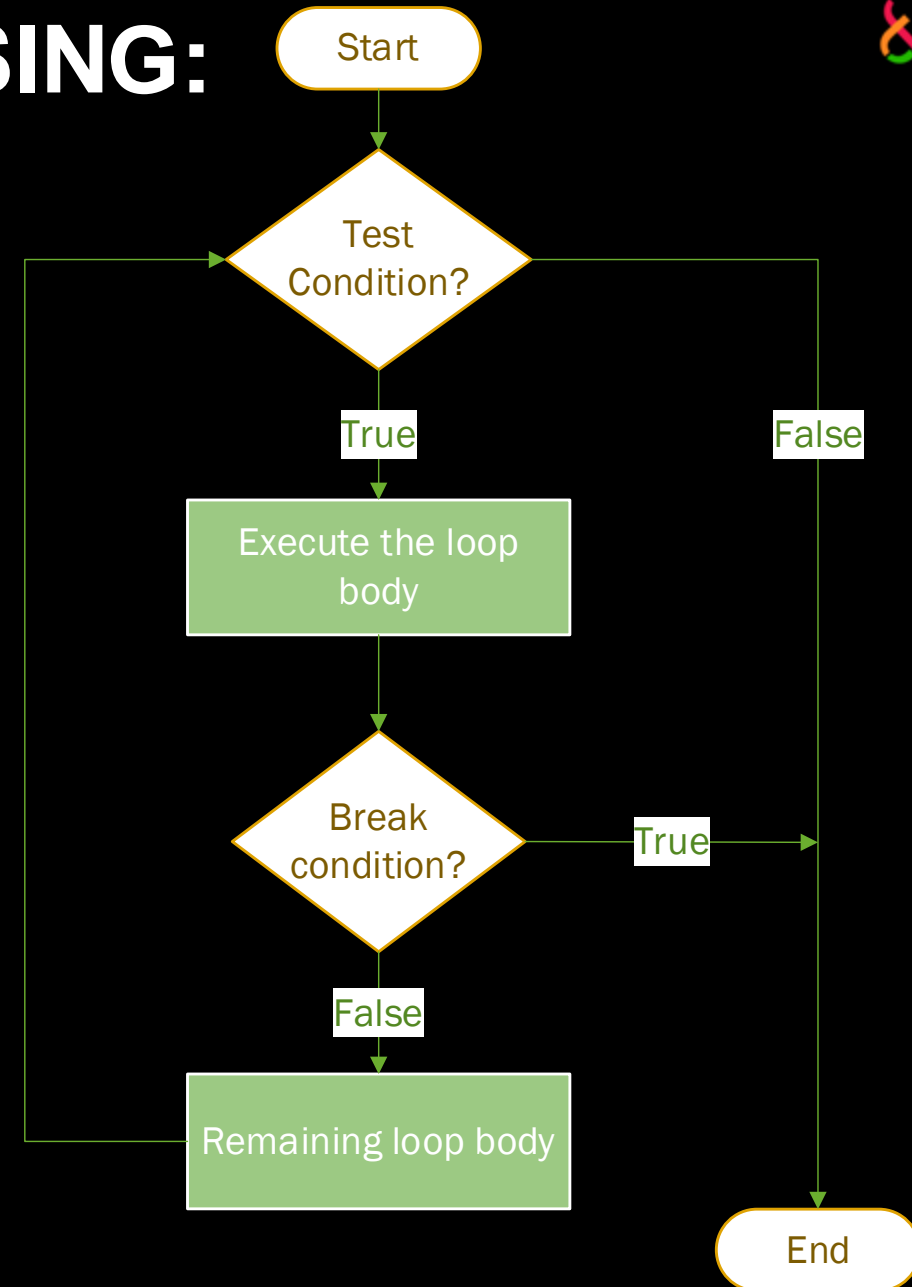
EVERYTHING

CONTROLLING LOOPS

<break>

CONTROLLING LOOPS USING: BREAK

- If the **break-command** is encountered, the iteration is terminated.
- If there are several nested iterations, loop within a loop, the innermost loop is ended.



EXAMPLE: BREAK

The program keeps going onwards from the user-given number one by one until it finds a number which is divisible by 13 (number % 13 == 0).

```
# Lecture 4, Example 5
# The loop keeps on going until number is divisible by 13
# This is done with check number % 13 == 0

print("Lecture 4, Example 5")

while True:
    n = int(input("Please enter a number: "))
    if (n % 13 == 0):
        print(n, "is divisible by 13")
        break
```



This cannot continue.

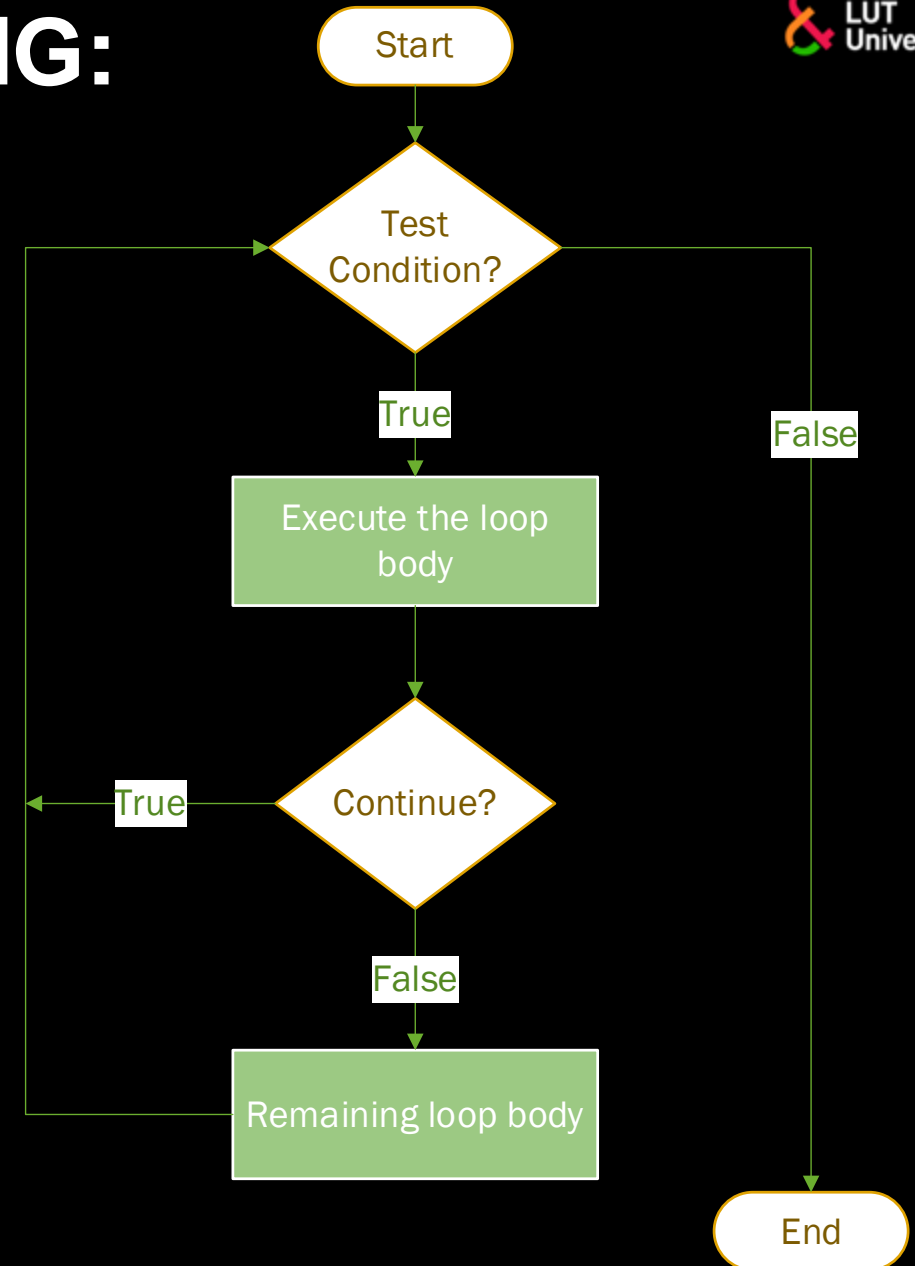
CONTROLLING LOOPS

<continue>

CONTROLLING LOOPS USING: CONTINUE

The **continue** Statement

- » The continue is also one of the iteration control commands.
- » The idea of continue is that it skips the rest of the iteration before starting a new round.
- » continue cannot exist outside iterative loop.



EXAMPLE: CONTINUE

```
# Lecture 4, Example 6
# Print the even numbers in the list

print("Lecture 4, Example 6")
list_numbers = list(range(2, 65, 3))
print("List: ", list_numbers)
print("Even numbers in the list: ")
for x in list_numbers:
    if x % 2 != 0:
        continue
    else:
        print(x, end=" ", " ")
```

```
Lecture 4, Example 6
List: [2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53, 56,
59, 62]
Even numbers in the list:
2, 8, 14, 20, 26, 32, 38, 44, 50, 56, 62,
```