



Lesson 3

# Python Fundamental

# Revisit

---

Using Random Library

---

Decision Making: if-elif-else

---

Logical Operators (==, !=, <, <=, >, >=)

---

Boolean Operators (and, or, not)

---

Boolean Values (True / False)

---

Functions: range(start, stop, step)

---

Control Structure: for loop

---

Lab: Calculate YRT fare

---

Exercise: a program use random, for loop, range(), and if – elif – else block

# Exercise 2 Review

```
#import random library
import random
#1. execute 10 times as a main loop
for n in range(0, 10):
    # 1.1 generate a random number between 1 and 100
    num = random.randint(1, 100)
    # Check if num mod 2 has remainder
    if num % 2 == 0:
        # Print for even number
        print(f"{num} - num is even")
    elif num > 50:
        print(f"{num} - High Odd")
    else:
        print(f"{num}")
```

# Lab – loop and decision making

Modify your dice program to create a "Guess the Number" game.

- The system generate a random number between 1 and 100.
- The user will have 5 attempts to guess the number.

After each guess, provide feedback to the user:

- If the guess is correct, print "BINGO!" and end the game.
- If the guess is incorrect and the user still has remaining attempts, prompt them to try again.
- If the user does not guess the number within 5 attempts, print "Fail" and reveal the correct number.

# while loop

---

- A while loop keeps running the same code again and again as long as the condition is True. True (upper T) is Boolean value in Python.
- The number of times a while loop runs depends on the condition being True, unlike a for loop, which runs a set number of times.

# Why Use a `while` Loop?

- Sometimes, we need to repeat an action until a certain condition is met such as displaying a menu until the user select an option to exit

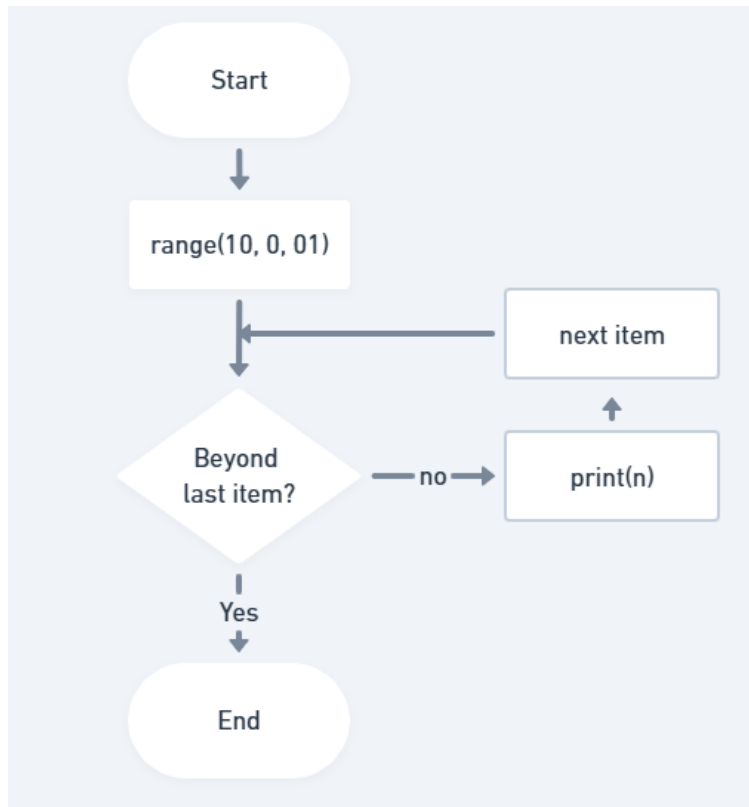
```
balance = 1000
while True:
    print("\nMenu:")
    print("1. Deposit")
    print("2. Withdraw")
    print("3. Check Balance")
    print("4. Exit")

    choice = input("Enter your choice: ")
    if choice == "4":
        print("Exiting. Goodbye!")
        break
```

# for Loop and while Loop Comparison

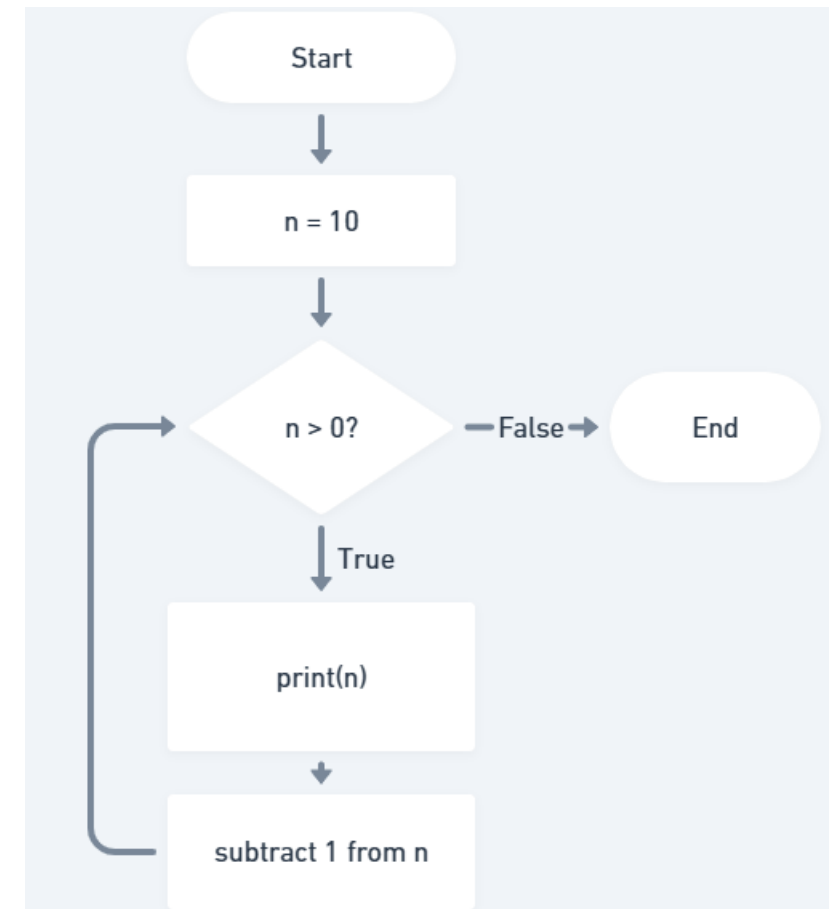
## Counting down with a for loop

```
for n in range(10, 0, -1):  
    print(n)
```



## Counting down with a while loop

```
n = 10  
while n > 0:  
    print(n)  
    n -= 1
```



# for Loop and while Loop Comparison

---

	<b>For Loop</b>	<b>While Loop</b>
Usage	Number of iterations is known upfront	Number of iterations is condition-based
Control Variable	Managed by the loop automatically	Update manually inside the loop
Example	Iterating through a range of numbers or a list	Waiting for a user input or reaching a condition



# Quiz 1

- You want to write a program that prints numbers from 1 to 20. Which type of loop (for or while) would be more suitable, and why?

## Quiz 2

- A student writes a loop that keeps running indefinitely, and their program never stops. Which type of loop (for or while) is more likely causing this issue, and why?

## Quiz 3

- A teacher wants students to write a loop that prints "Hello" exactly 5 times.
- Student A uses a for loop.
- Student B uses a while loop.
- Can both students achieve the same result? Why or why not?

# Lab

- Develop a program to:
  - Keep asking the user to enter numbers
  - Stop if the user enters a negative number
  - Print the sum of all positive numbers entered

- The output should look like:

Enter a number: 5

Enter a number: 10

Enter a number: -1

Sum of positive numbers: 15

- Hints: use `input()`, `print()`, `int()` functions in the while loop

# Exercise

- Create a program that helps plan a birthday party with a budget. Allow the user to buy party items, check remaining budget, and see their shopping list.
- Refer to exercise sheet for more details and hints