Python for DevOps

Bharath CY +919535955530

Introduction

- Python is a high-level, interpreted programming language known for its simplicity and readability.
- Created by Guido van Rossum and first released in 1991.
- Python is widely used in various fields such as
  - web development,
  - Cloud Computing
  - Automation
  - data analysis,
  - artificial intelligence,
  - scientific computing, and more.

Environment Variables

- **On linux / On Mac**
  - cat /etc/environment
  - cat /etc/paths

  - Bashrc
  - zshrc

- **On Windows**
  - Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Session Manager\Environment
  - sysdm.cpl
  - PATH

Setting Up Python

- Install Python
- Verify Installation
- Interactive Mode.
- Start Writing the Scripts

Basic Syntax and Concepts

- Comments
- Variables
- Data Types
- Control Flow
- Functions

Data Types

- Integers
- Floats
- Strings
- Lists
- Tuples
- Set
- Dictionaries
- None
- Boolean

Truthy and Falsy

- **Falsy values**:
  - False, None, 0, 0.0, 0j, '', [], (), {}, set(), range(0)

- **Truthy values**:
  - Everything else not listed above

Operators and Built-in Methods

Bharath CY +919535955530

Operators in python

1. **Arithmetic Operators**
2. **Comparison (Relational) Operators**
3. **Logical Operators**
4. **Bitwise Operators**
5. **Assignment Operators**
6. **Identity Operators**
7. **Membership Operators**
8. Ternary Operator

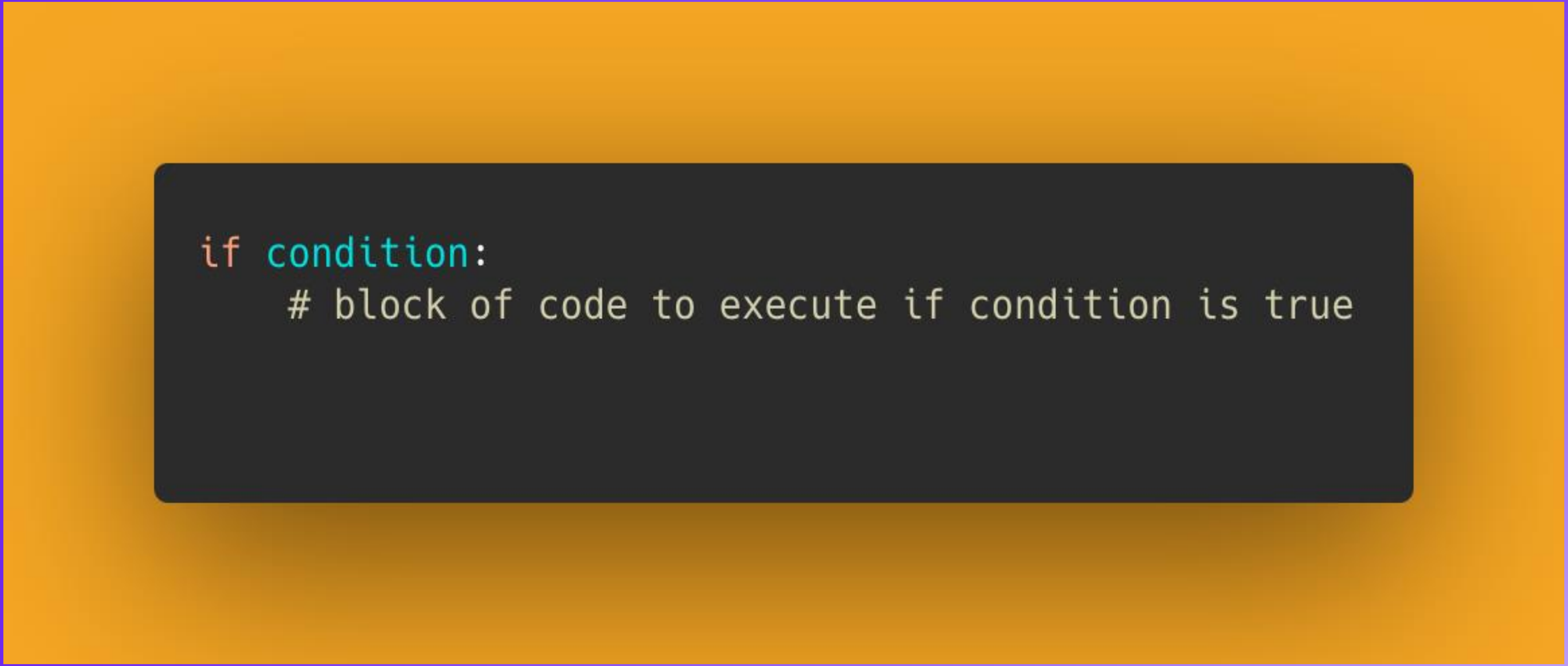
Control Flow

- Conditionals
- Loops

Conditionals

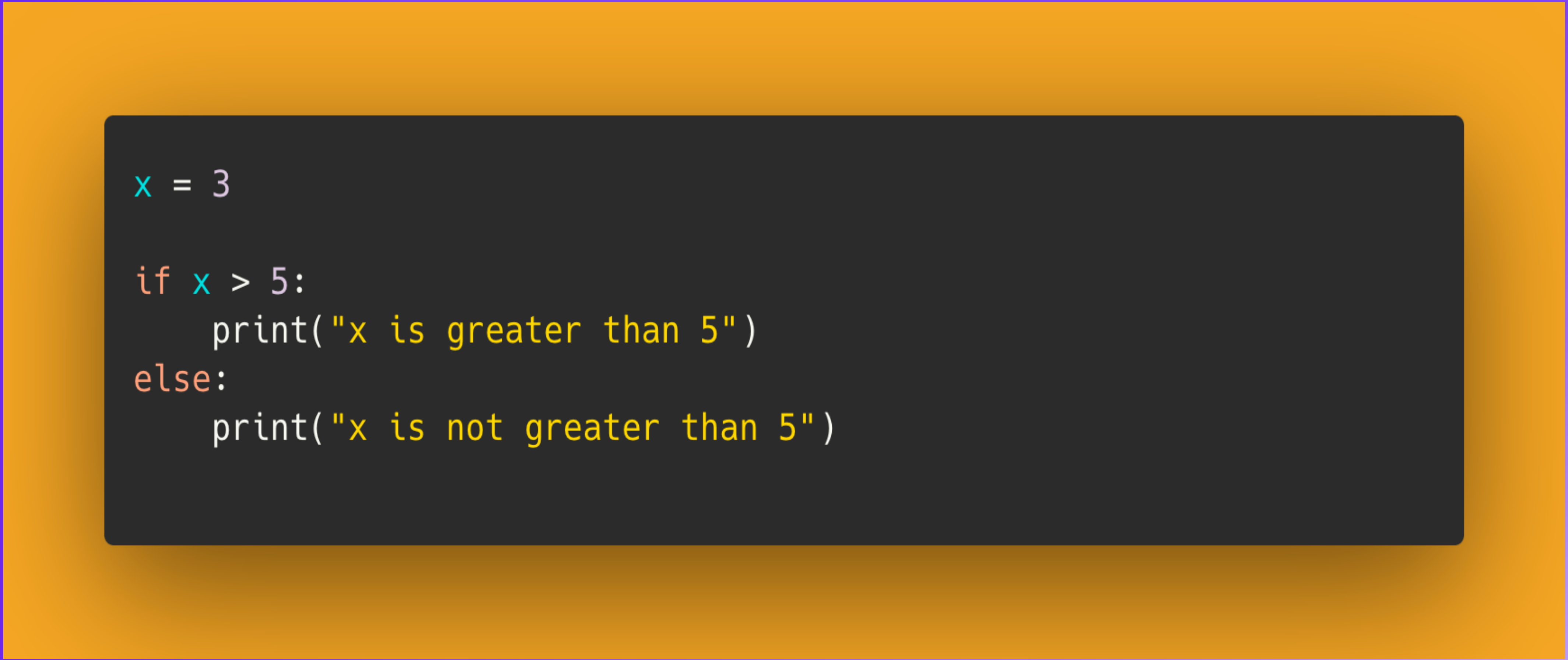
Bharath CY +919535955530

If Statements



```
if condition:
    # block of code to execute if condition is true
```

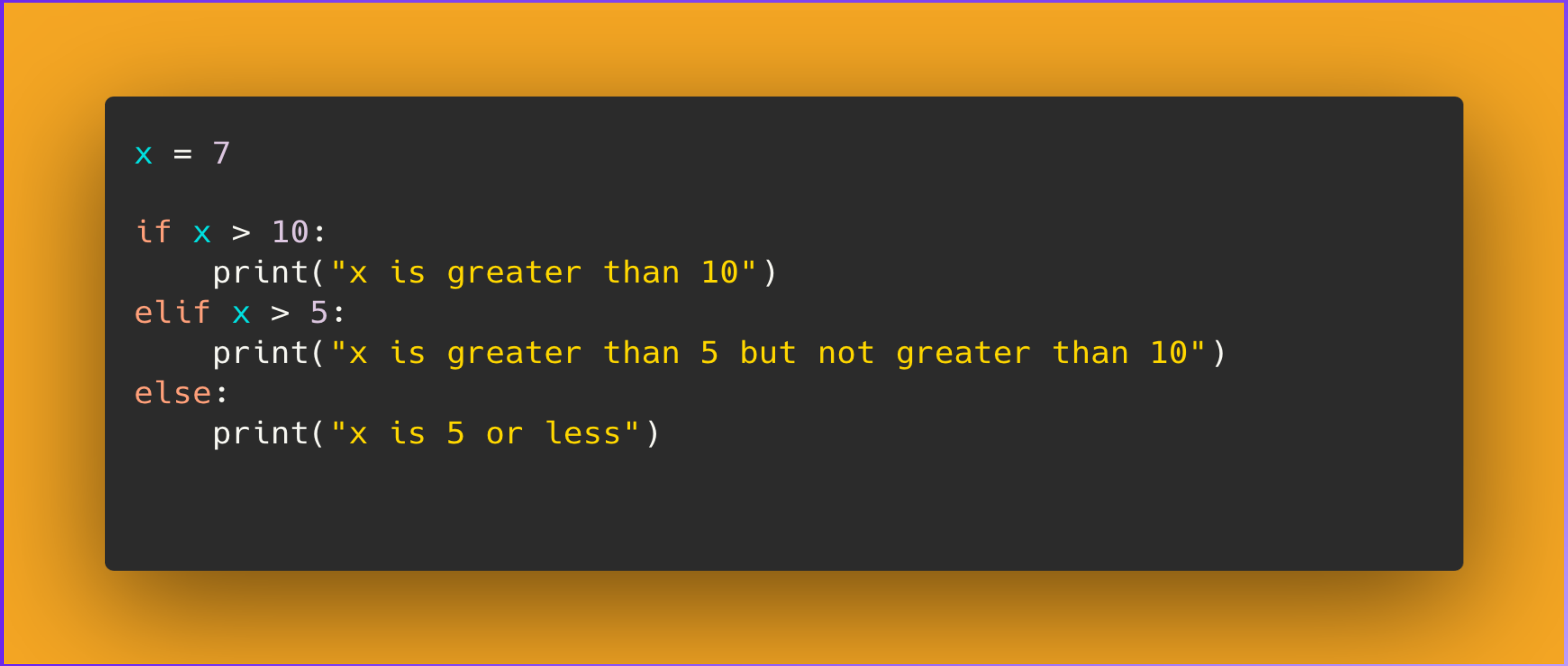
If else statements



```python
x = 3

if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
```

If elif else statements

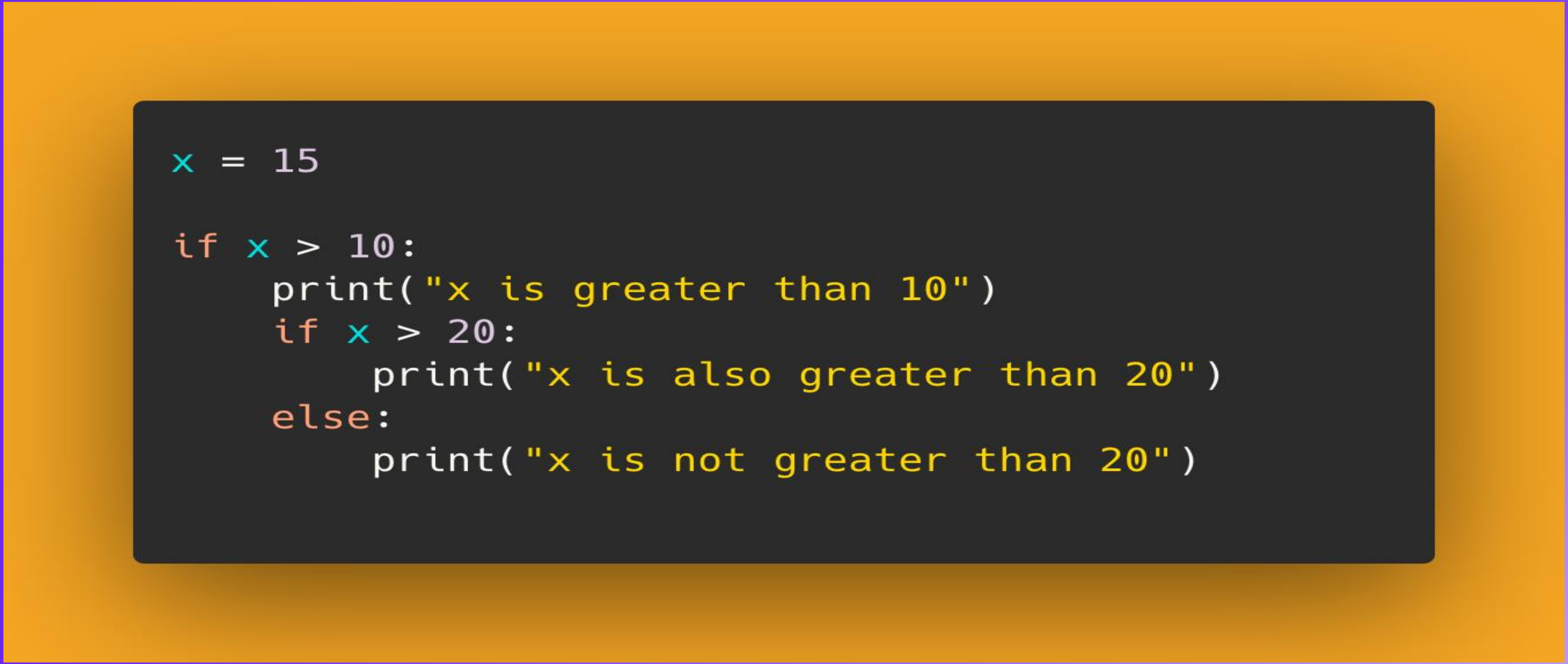


```python
x = 7

if x > 10:
    print("x is greater than 10")
elif x > 5:
    print("x is greater than 5 but not greater than 10")
else:
    print("x is 5 or less")
```

Bharath CY +919535955530

Nested If statements



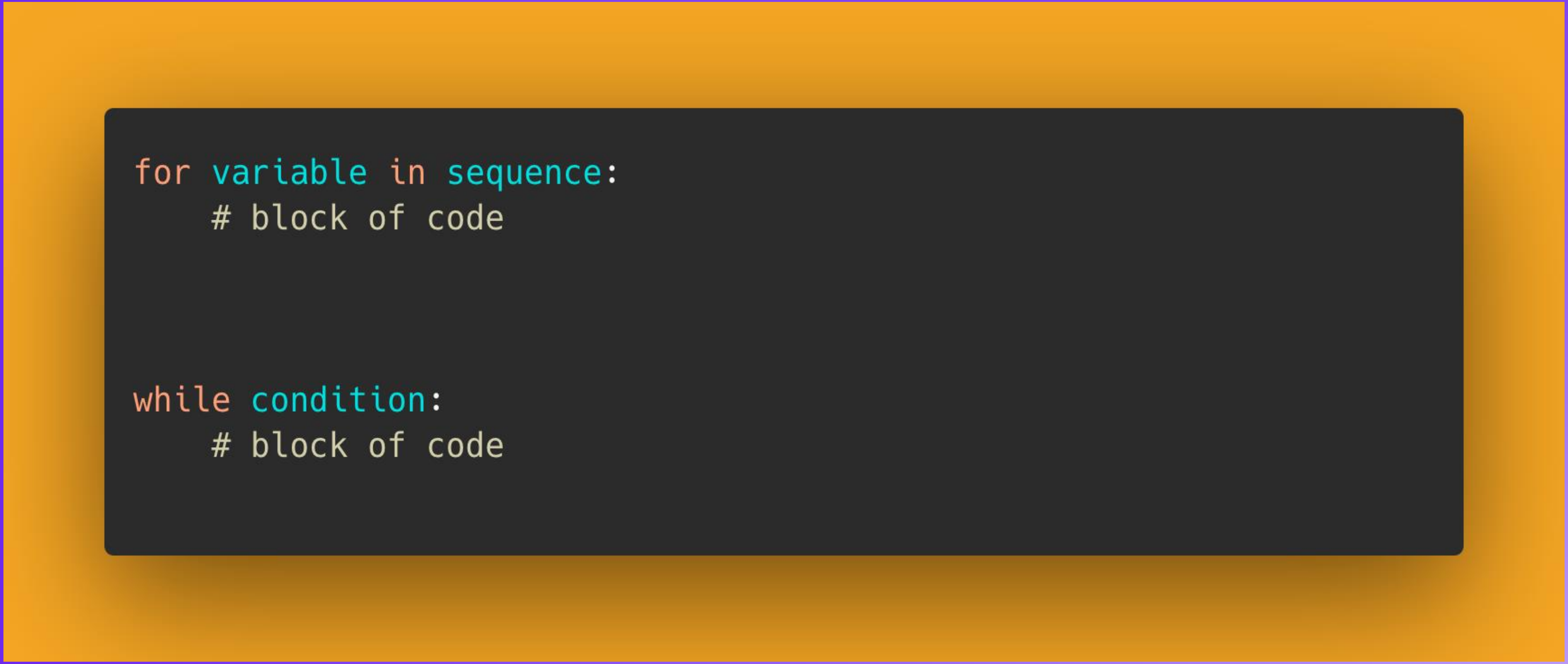
```
x = 15

if x > 10:
    print("x is greater than 10")
    if x > 20:
        print("x is also greater than 20")
    else:
        print("x is not greater than 20")
```

Loops

Bharath CY +919535955530

Loops in python



```
for variable in sequence:
    # block of code



while condition:
    # block of code
```

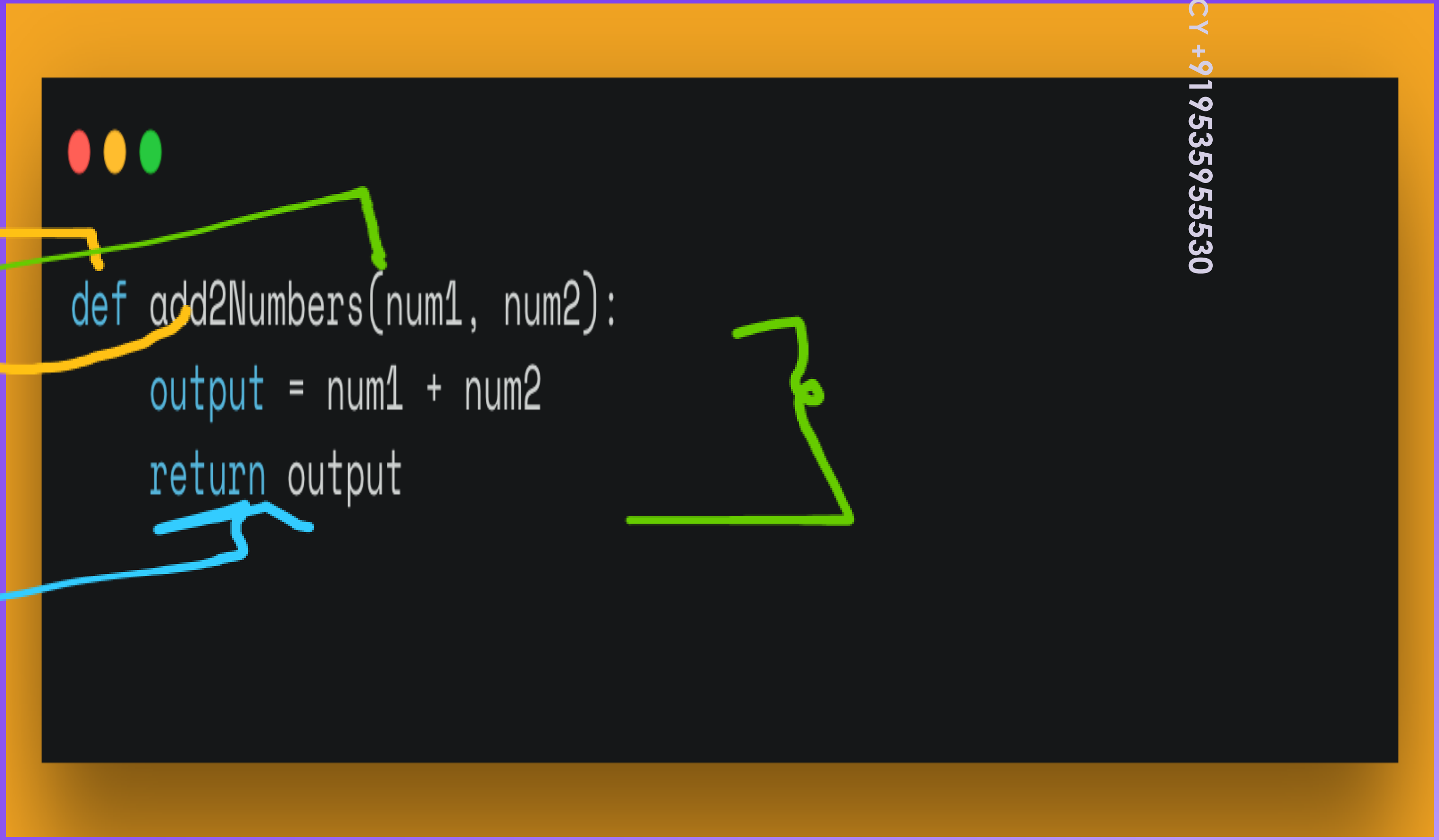
Bharath CY +919535955530

Functions

Bharath CY +919535955530

Functions

- Function Keyword
- Function Name
- Parameters
- Function Body
- Return Statement



```
def add2Numbers(num1, num2):
    output = num1 + num2
    return output
```

Once defined, this function can be called multiple times with different arguments

Modules

A file containing Python definitions and statements

- **Built-in Modules:**

  Python comes with a wide variety of modules that provide standard functionality (like mathematical operations, file I/O, etc.). These are part of the standard library.

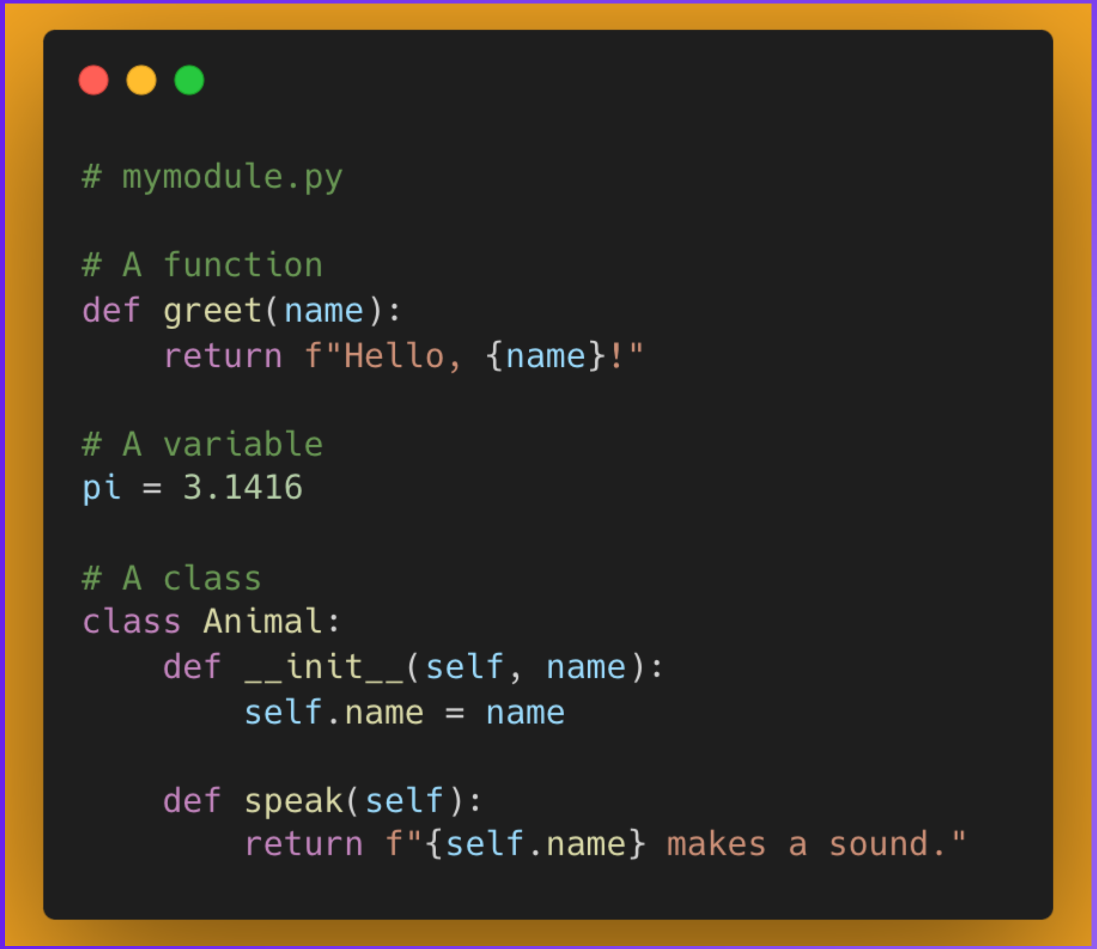
- **External Modules:**

  Modules that are installed via external packages (using tools like pip).

- **Custom Modules:**

  You can create your own modules by saving Python code in .py files and importing them into other programs.

Modules



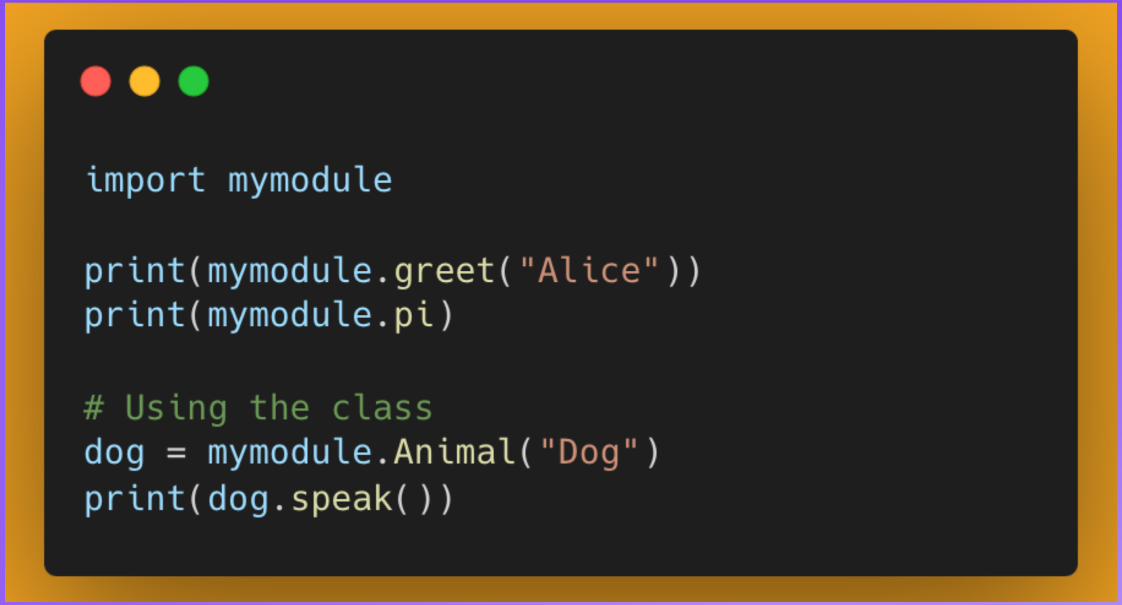
```python
# mymodule.py

# A function
def greet(name):
    return f"Hello, {name}!"

# A variable
pi = 3.1416

# A class
class Animal:
    def __init__(self, name):
        self.name = name

    def speak(self):
        return f"{self.name} makes a sound."
```



```python
import mymodule

print(mymodule.greet("Alice"))
print(mymodule.pi)

# Using the class
dog = mymodule.Animal("Dog")
print(dog.speak())
```

Key components of Boto3

1. **Session.**

2. **Client.**

3. **Resource.**

4. **Collections.**

5. **Waiters.**

6. **Paginators .**

Classes

Bharath CY +919535955530

Classes and Objects

- Classes provide a blueprint for creating objects and define the structure and behavior of those objects.

- Objects are concrete instances of classes, representing specific entities with their own state and behavior.