Artificial Intelligence (AI)

With Python

By: Saad B. Younis

saad@coie-nahrain.edu.iq

List OF Contents

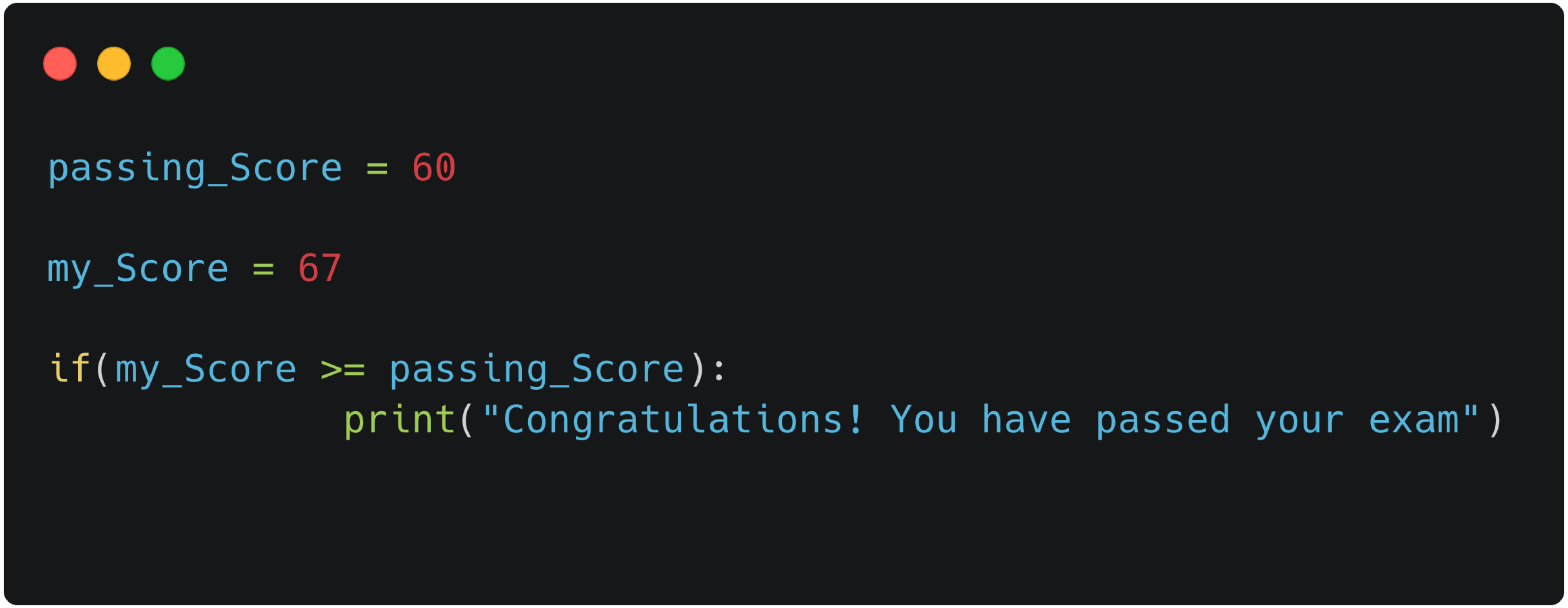
- If
- Elif
- Else
- Or
- And
- Match

IF

If statements

Python if statement is one of the most commonly used conditional statements in programming languages. It decides whether certain statements need to be executed or not. It checks for a given condition, if the condition is true, then the set of code present inside the ” if ” block will be executed otherwise not.

Example (1)



```
passing_Score = 60

my_Score = 67

if(my_Score >= passing_Score):
             print("Congratulations! You have passed your exam")
```

Output : Congratulations! You have passed your exam.

else

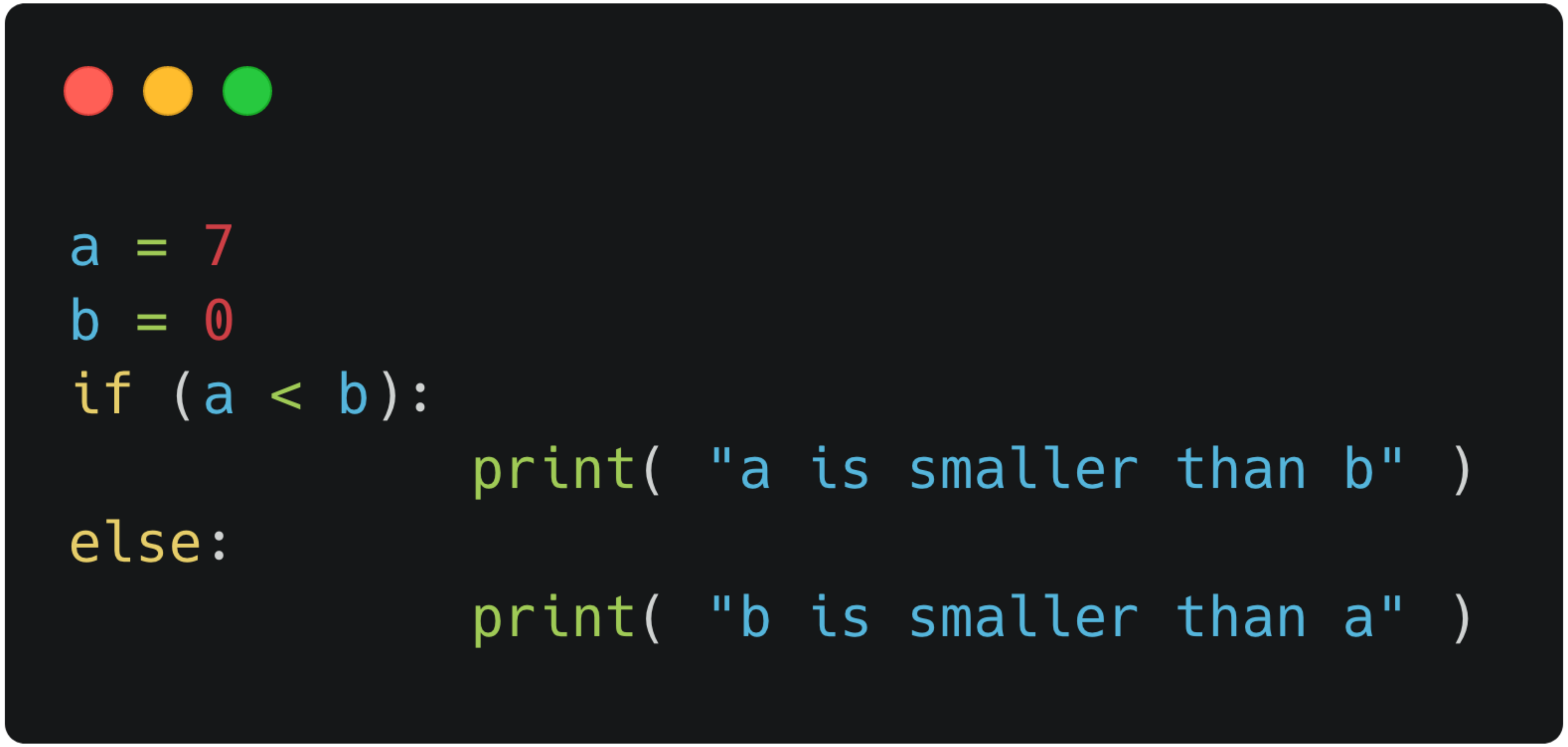
If-else statements

The statement itself says if a given condition is true then execute the statements present inside the “if block” and if the condition is false then execute the “else” block.

The “else” block will execute only when the condition becomes false. It is the block where you will perform some actions when the condition is not true.

if-else statement evaluates the Boolean expression. If the condition is TRUE then, the code present in the “ if “ block will be executed otherwise the code of the “else“ block will be executed

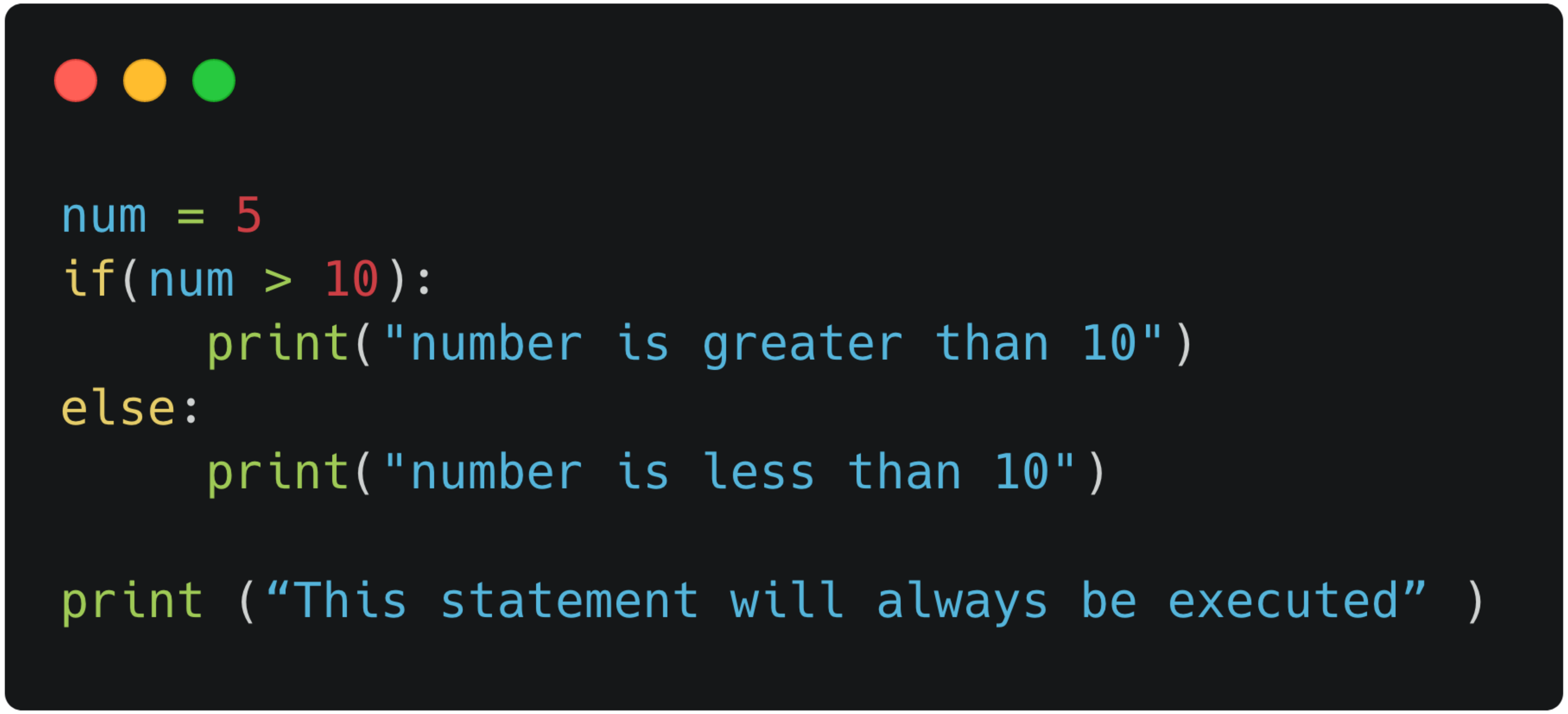
Example (2)



```python
a = 7
b = 0
if (a < b):
            print( "a is smaller than b" )
else:
            print( "b is smaller than a" )
```

Output : b is smaller than a

Example (3)



```
num = 5
if(num > 10):
     print("number is greater than 10")
else:
     print("number is less than 10")

print (“This statement will always be executed” )
```

Output :

```
number is less than 10.
This statement will always be executed.
```

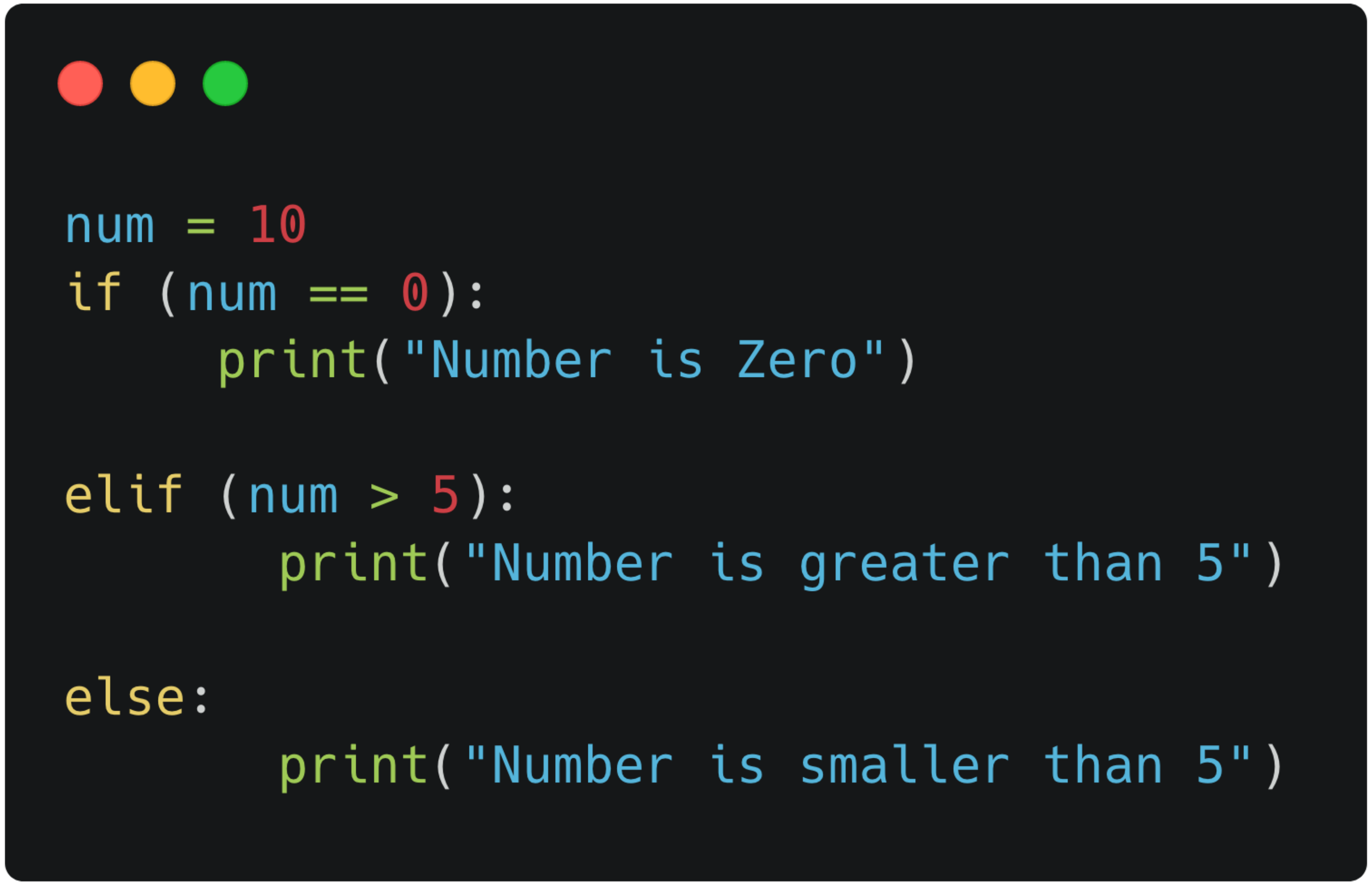
elif

elif statements

In Python, we have one more conditional statement called “elif” statements. “elif” statement is used to check multiple conditions only if the given condition is false. It’s similar to an “if-else” statement and the only difference is that in “else” we will not check the condition but in “elif” we will check the condition.

“elif” statements are similar to “if-else” statements but “elif” statements evaluate multiple conditions.

Example (4)



```
num = 10
if (num == 0):
     print("Number is Zero")

elif (num > 5):
       print("Number is greater than 5")

else:
       print("Number is smaller than 5")
```

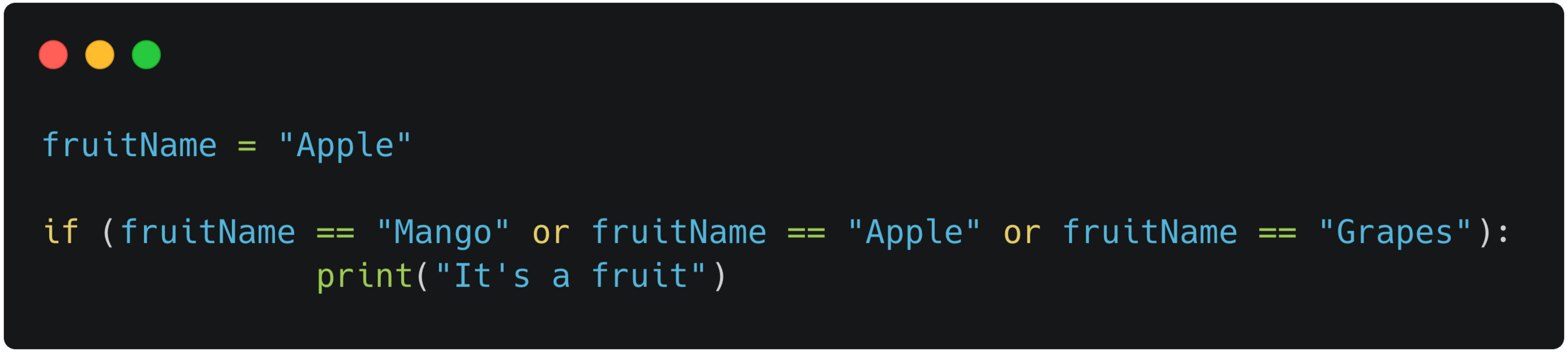
Output : Number is greater than 5

Or

or Keyword

Python’s **or** keyword is used to determine if at least one of the operands is truthy. An or statement returns the first operand if it is truthy and otherwise returns the second operand:

Example (5)



```
fruitName = "Apple"

if (fruitName == "Mango" or fruitName == "Apple" or fruitName == "Grapes"):
              print("It's a fruit")
```

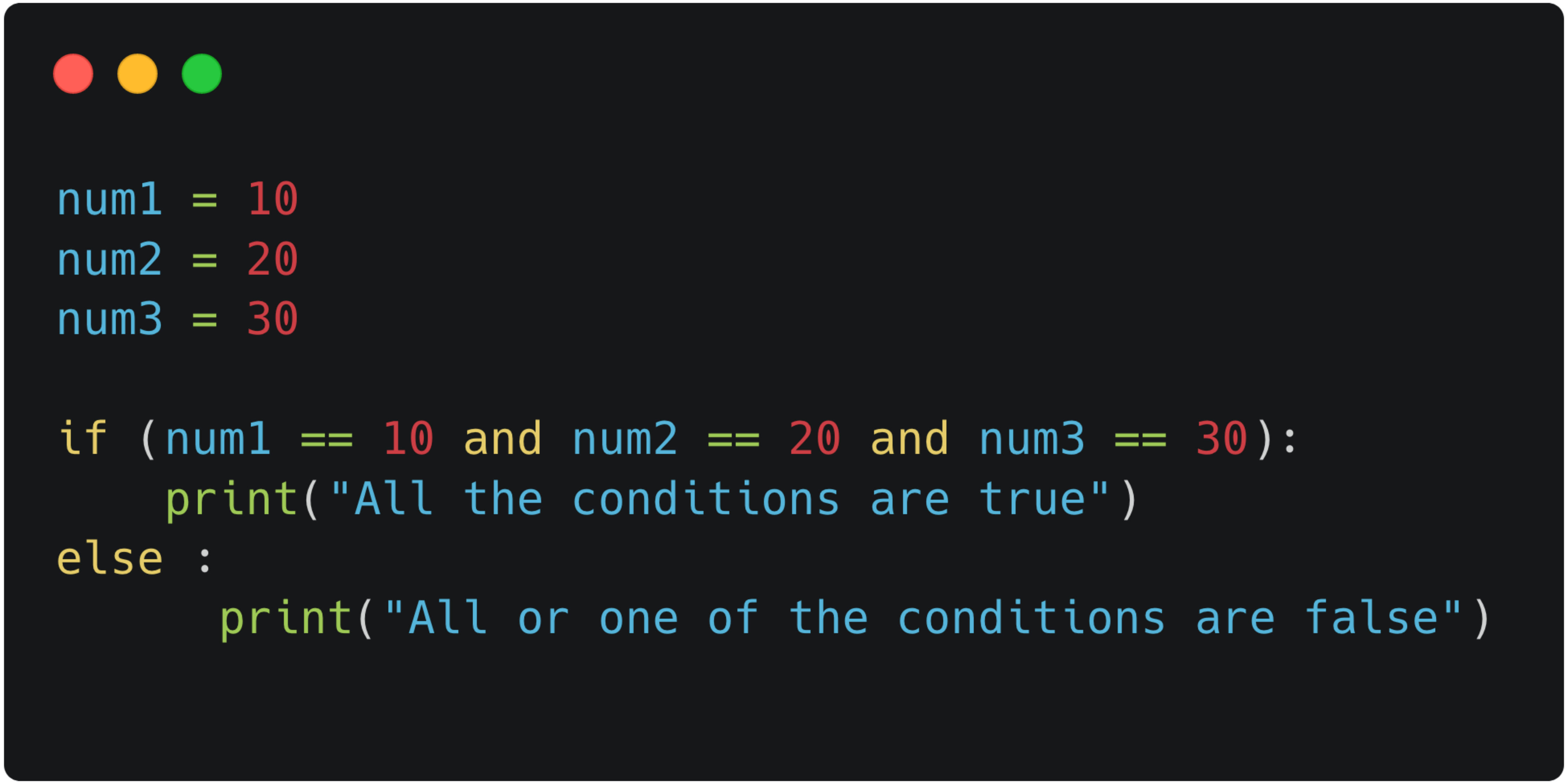
Output : It’s a fruit

And

and Keyword

Python’s and operator takes two **operands**, which can be Boolean expressions, objects, or a combination. With those operands, the and operator builds more elaborate expressions. The operands in an and expression are commonly known as **conditions**. If both conditions are true, then the and expression returns a true result. Otherwise, it returns a false result:

Example (6)

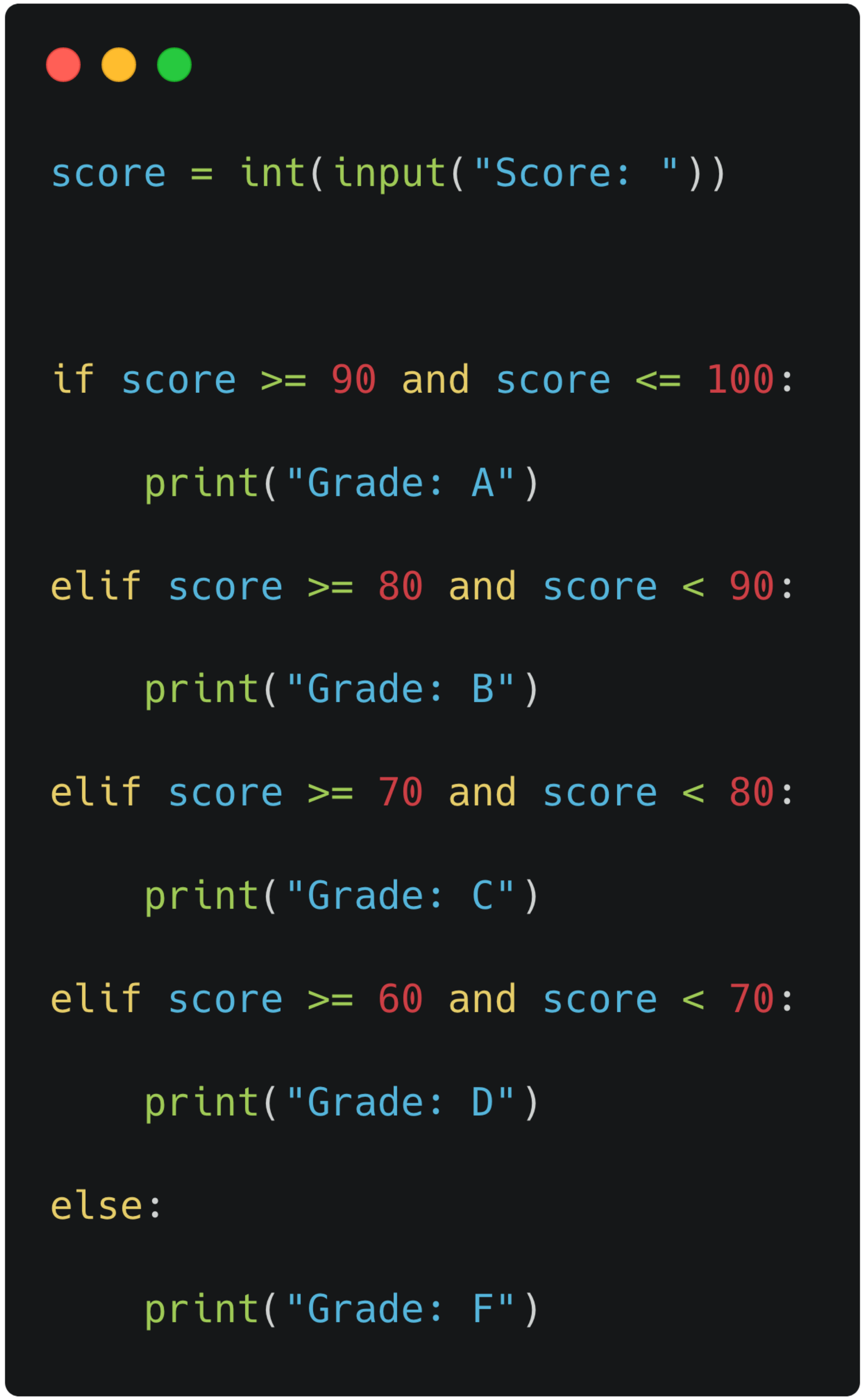


```
num1 = 10
num2 = 20
num3 = 30

if (num1 == 10 and num2 == 20 and num3 == 30):
    print("All the conditions are true")
else :
      print("All or one of the conditions are false")
```

Output: All the conditions are true

Example (7)



```python
score = int(input("Score: "))

if score >= 90 and score <= 100:
    print("Grade: A")
elif score >= 80 and score < 90:
    print("Grade: B")
elif score >= 70 and score < 80:
    print("Grade: C")
elif score >= 60 and score < 70:
    print("Grade: D")
else:
    print("Grade: F")
```

Match

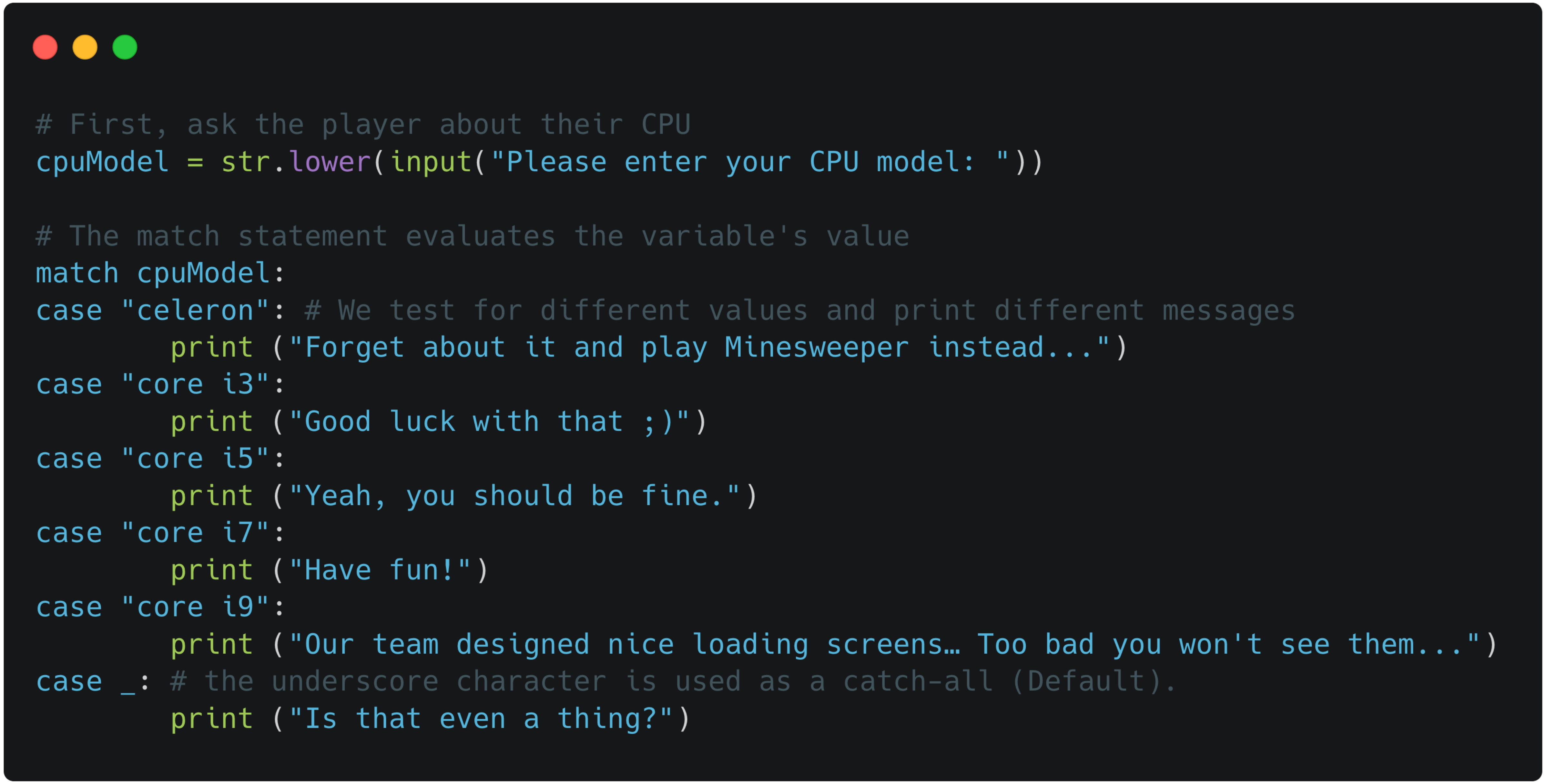
Match

To implement switch-case like characteristics and if-else functionalities, we use a **match case in python. A match statement will compare a given variable's value to different shapes, also referred to as the pattern.** The main idea is to keep on comparing the variable with all the present patterns until it fits into one.

Example (8)

let’s say you’re building a program to check a computer’s processor. Based on the result, the program will let the gamer know if their processor is compatible with a certain video game. Here’s how our program would look:

Example (8) - Con..



```
# First, ask the player about their CPU
cpuModel = str.lower(input("Please enter your CPU model: "))

# The match statement evaluates the variable's value
match cpuModel:
case "celeron": # We test for different values and print different messages
        print ("Forget about it and play Minesweeper instead...")
case "core i3":
        print ("Good luck with that ;)")
case "core i5":
        print ("Yeah, you should be fine.")
case "core i7":
        print ("Have fun!")
case "core i9":
        print ("Our team designed nice loading screens… Too bad you won't see them...")
case _: # the underscore character is used as a catch-all (Default).
        print ("Is that even a thing?")
```

Input: Please enter your CPU model: core i9

Output: Our teams designed nice loading screens... Too bad you won't see them...