LAB REPORT # 1

NAME : AMISH BABAR

REG NO: B22F1253AI092

SUBJECT: PROGRAMMING OF AI

SECTION: AI BLUE F22

PROGRAM: (AI)

LAB 4 AND 5: PROGRAMMING FOR AI:

Task no 1:

Differentiate between assignment operator and equality operatorANSWER:

The assignment operator (=) and the equality operator (==) serve distinct purposes in programming. The assignment operator is used to assign a value to a variable, while the equality operator is used to compare two values for equality.

Assignment Operator (=):

The assignment operator is denoted by a single equal sign (=).

It assigns the value on the right side to the variable on the left side.

Example: x = 5 assigns the value 5 to the variable x.

It is used for setting or resetting values stored in variables.

In programming, = is used for assignment, not for comparison.

Equality Operator (==):

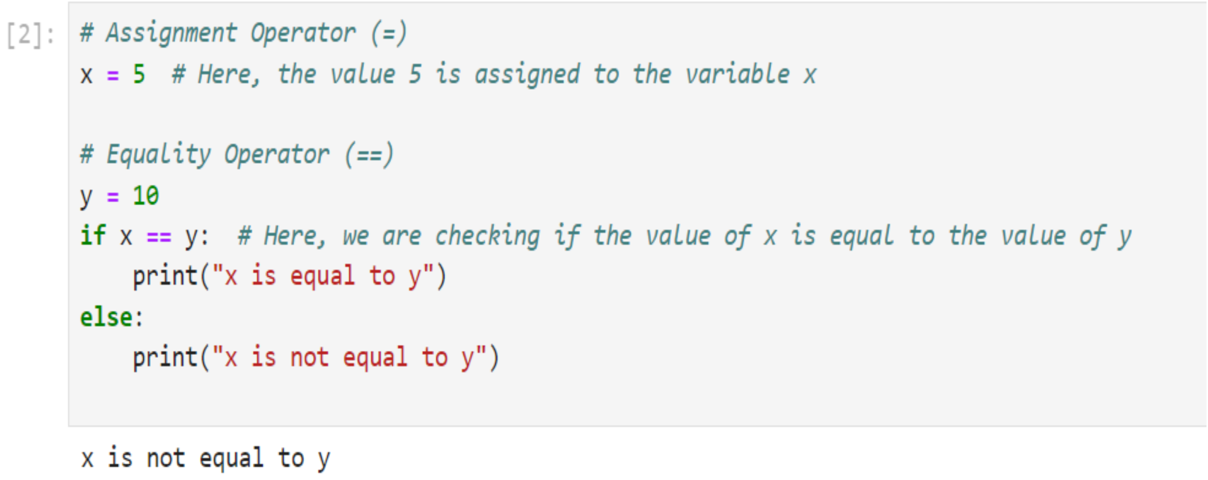
The equality operator is denoted by two consecutive equal signs (==).

It checks whether two given operands are equal or not.

If the operands are equal, it returns true; otherwise, it returns false.Example: a

== b tests if the value stored in variable a is equal to b.

It is a relational or comparison operator used for comparing two values.



[2]:

```
# Assignment Operator (=)
x = 5  # Here, the value 5 is assigned to the variable x

# Equality Operator (==)
y = 10
if x == y:  # Here, we are checking if the value of x is equal to the value of y
    print("x is equal to y")
else:
    print("x is not equal to y")
```

```
x is not equal to y
```

TASK 2:

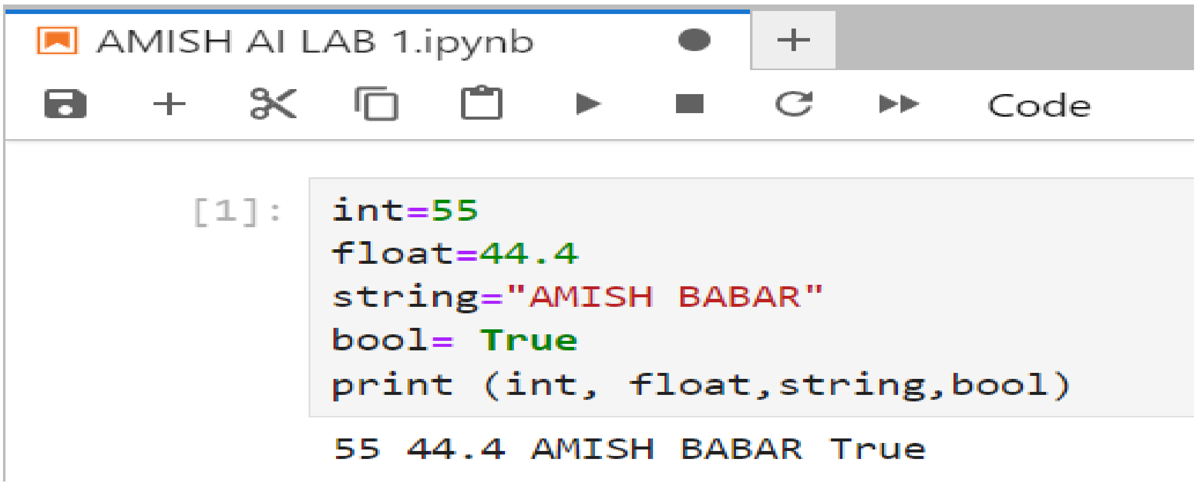
1. Explore all kinds /data types of variables. And write code for each datatype.

# Integer variable

# Floating-point variable

# String variable

# Boolean variable



```
[1]: int=55
     float=44.4
     string="AMISH BABAR"
     bool= True
     print (int, float,string,bool)
```

```
55 44.4 AMISH BABAR True
```

Task 3:

Explain the rules for variables name and use different variables name including special characters , numbers etc.

Answer:

RULES:

Variable names must begin with a letter, dollar sign ($), or underscore (_). They can only contain alphanumeric characters (a-z, A-Z, 0-9) and underscores.

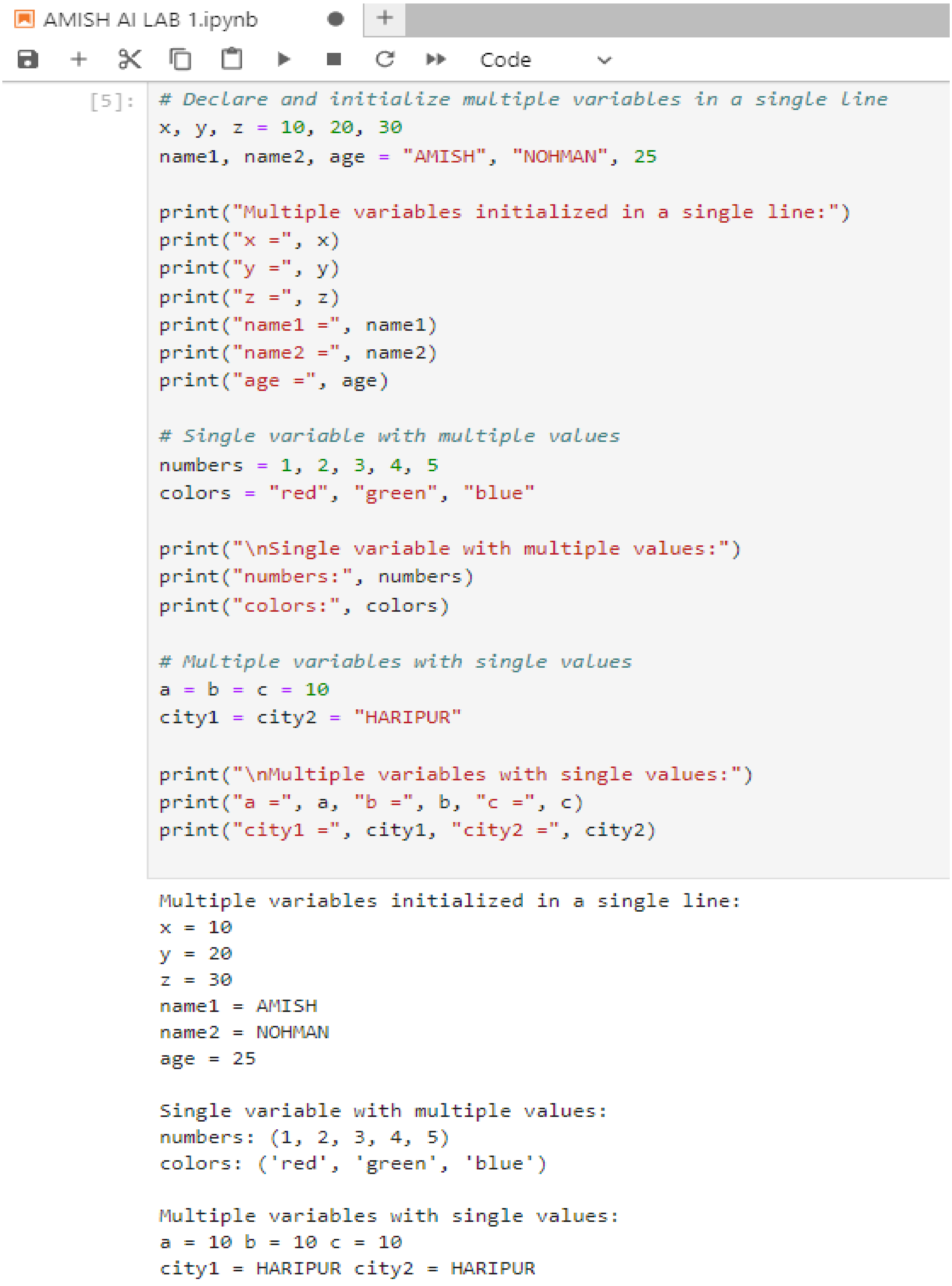
Variable names are case-sensitive.

No spaces are allowed in variable names.

Variable names cannot be any programming language keywords.

TASK 4:

Declare and initialize multiple variables in a single line . Single variable with multiple values ,multiple variables with single values. Each having at least two examples



```
# Declare and initialize multiple variables in a single line
x, y, z = 10, 20, 30
name1, name2, age = "AMISH", "NOHMAN", 25

print("Multiple variables initialized in a single line:")
print("x =", x)
print("y =", y)
print("z =", z)
print("name1 =", name1)
print("name2 =", name2)
print("age =", age)

# Single variable with multiple values
numbers = 1, 2, 3, 4, 5
colors = "red", "green", "blue"

print("\nSingle variable with multiple values:")
print("numbers:", numbers)
print("colors:", colors)

# Multiple variables with single values
a = b = c = 10
city1 = city2 = "HARIPUR"

print("\nMultiple variables with single values:")
print("a =", a, "b =", b, "c =", c)
print("city1 =", city1, "city2 =", city2)
```

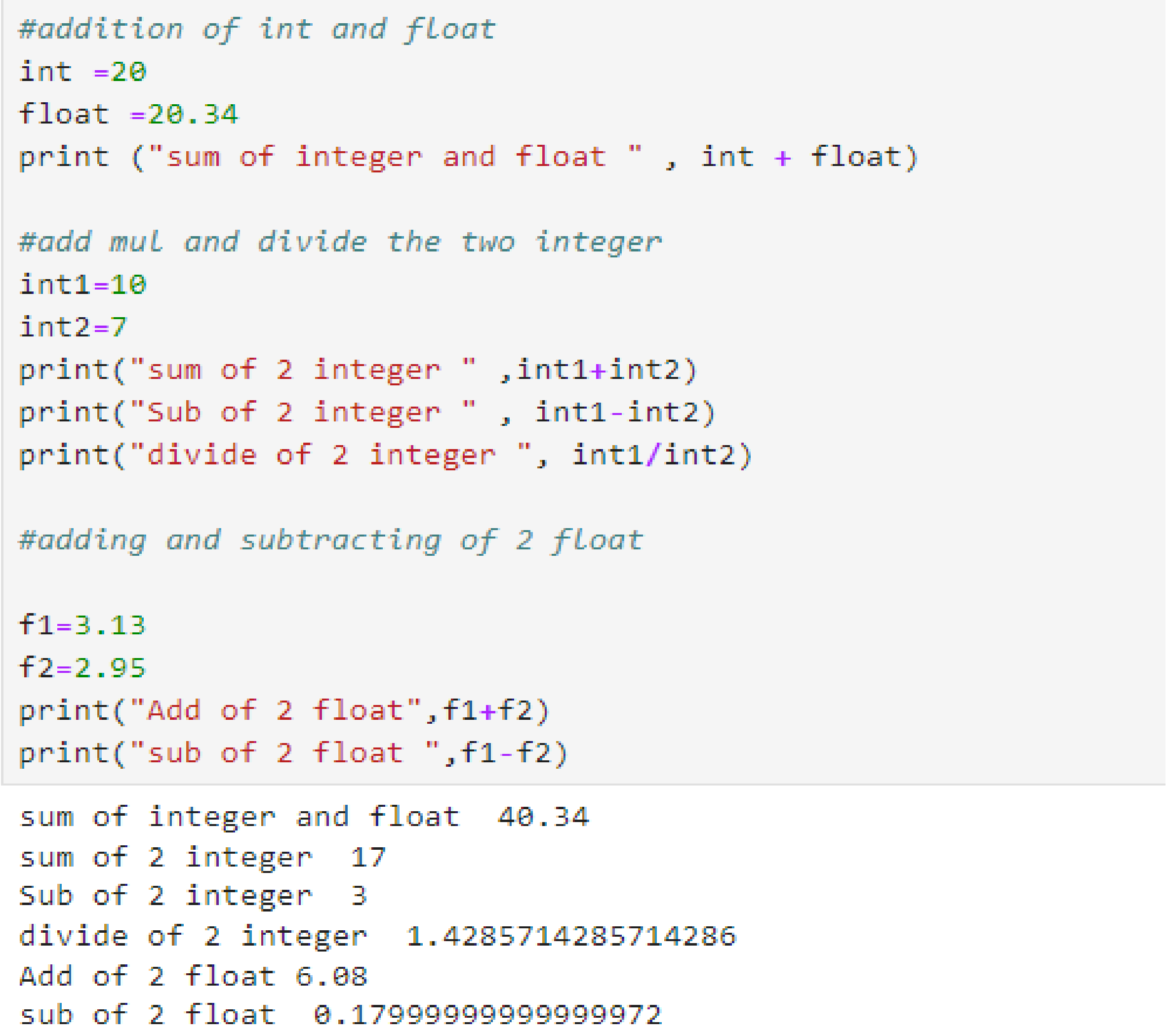
```
Multiple variables initialized in a single line:
x = 10
y = 20
z = 30
name1 = AMISH
name2 = NOHMAN
age = 25

Single variable with multiple values:
numbers: (1, 2, 3, 4, 5)
colors: ('red', 'green', 'blue')

Multiple variables with single values:
a = 10 b = 10 c = 10
city1 = HARIPUR city2 = HARIPUR
```

Task 5:

Perform arithmetic operations on integers and floating-point numbers
# Adding, Subtracting, Multiplication and dividing two integer
# Adding, Subtracting, Multiplication and dividing floating-point variables.



```
#addition of int and float
int =20
float =20.34
print ("sum of integer and float " , int + float)

#add mul and divide the two integer
int1=10
int2=7
print("sum of 2 integer " ,int1+int2)
print("Sub of 2 integer " , int1-int2)
print("divide of 2 integer ", int1/int2)

#adding and subtracting of 2 float

f1=3.13
f2=2.95
print("Add of 2 float",f1+f2)
print("sub of 2 float ",f1-f2)
```

```
sum of integer and float  40.34
sum of 2 integer  17
Sub of 2 integer  3
divide of 2 integer  1.4285714285714286
Add of 2 float 6.08
sub of 2 float  0.17999999999999972
```

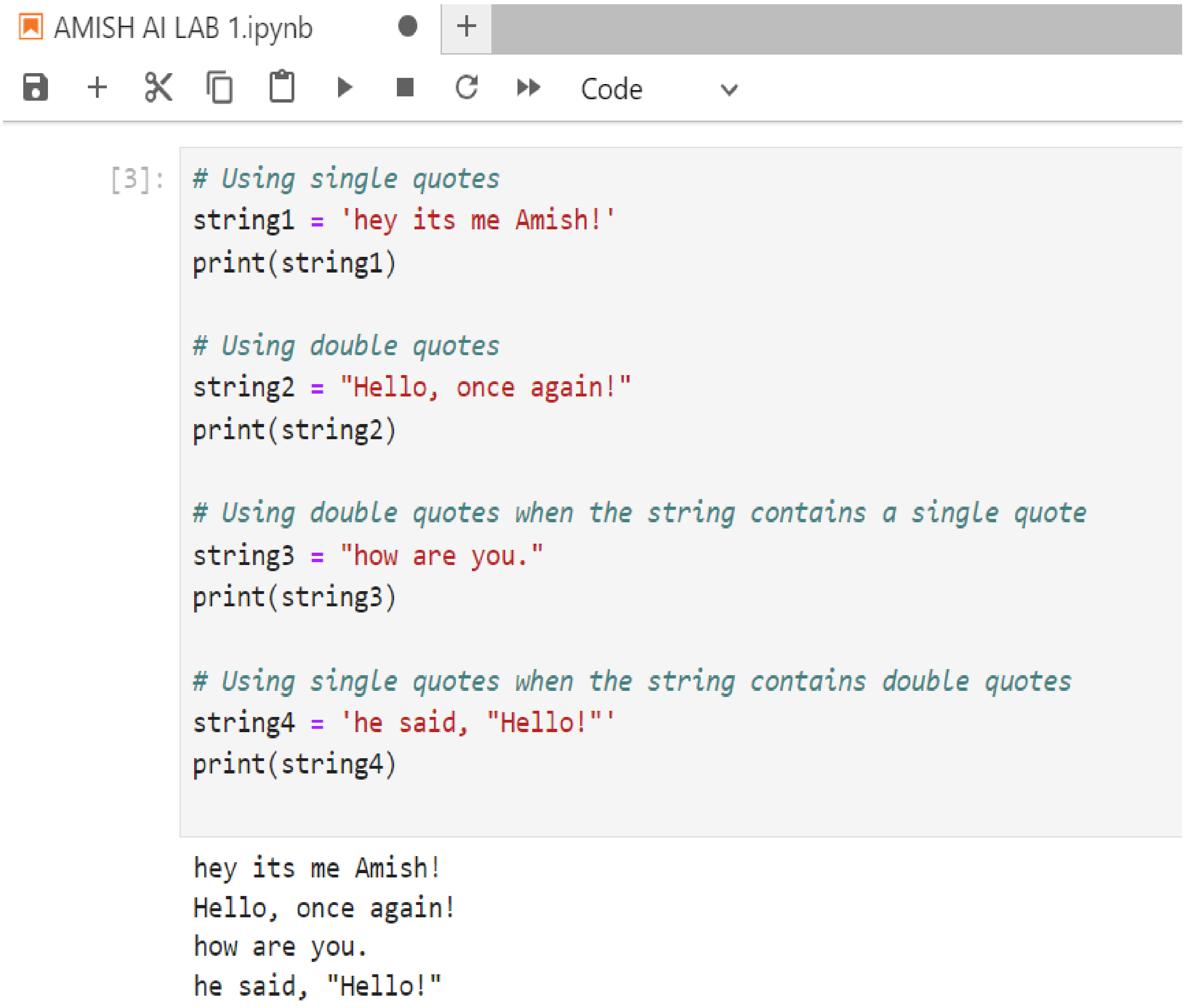
Task 6:

A string literal is a sequence of characters enclosed in quotes. In Python, we can use either single quotes (‘…’) or double quotes (“…”) to create a string.

Using single quotes

Using double quotes

Using double quotes when the string contains a single quote # Using single quotes when the string contains double quotes



```
[3]: # Using single quotes
     string1 = 'hey its me Amish!'
     print(string1)

     # Using double quotes
     string2 = "Hello, once again!"
     print(string2)

     # Using double quotes when the string contains a single quote
     string3 = "how are you."
     print(string3)

     # Using single quotes when the string contains double quotes
     string4 = 'he said, "Hello!"'
     print(string4)
```

```
hey its me Amish!
Hello, once again!
how are you.
he said, "Hello!"
```

Task 7:

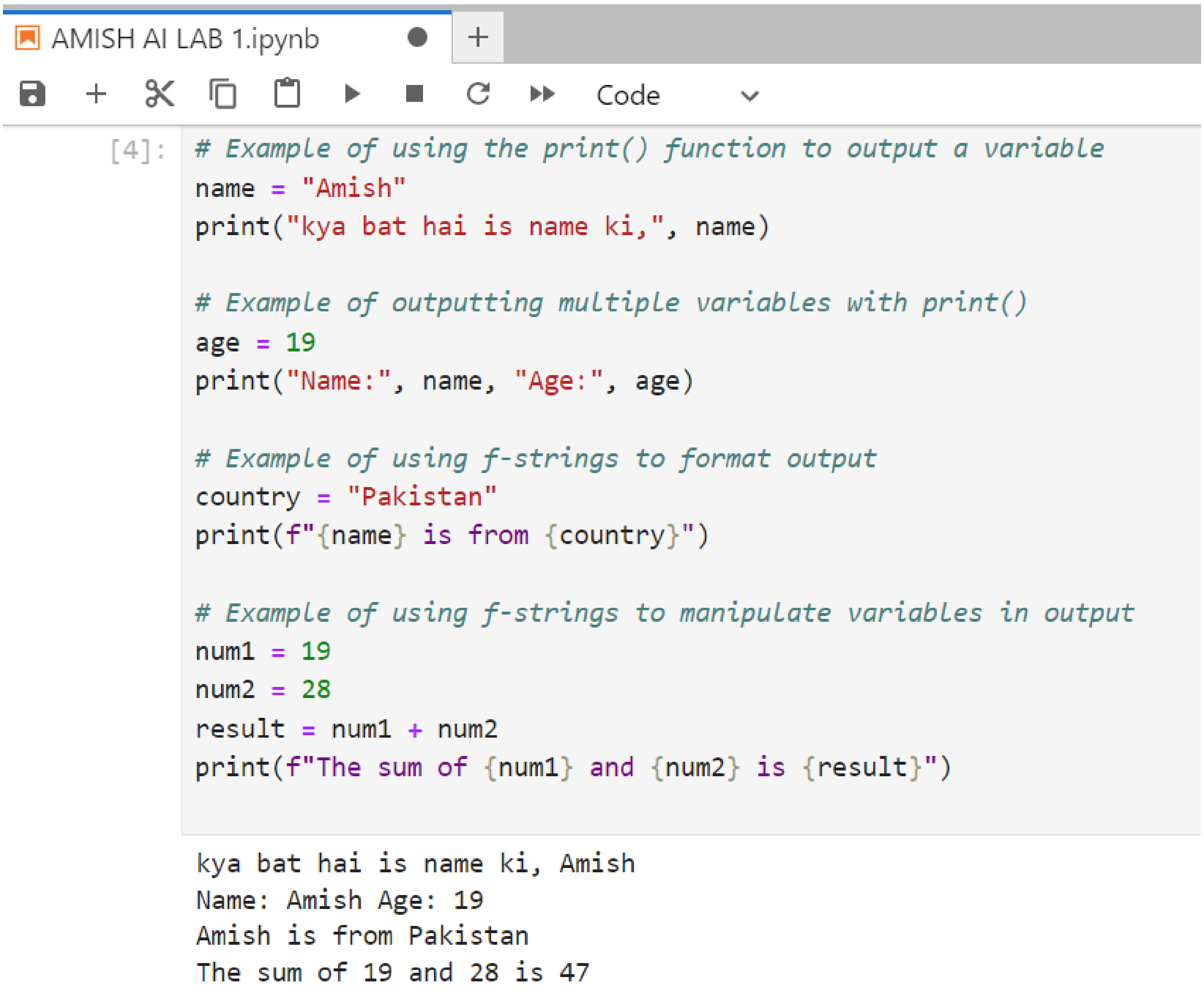
The print() function is a built-in function in Python that allows you to output variables and other data to the console.

Example of using the print() function to output a variable# Example of

outputting multiple variables with print()

Example of using f-strings to format output

Example of using f-strings to manipulate variables in output



```
# Example of using the print() function to output a variable
name = "Amish"
print("kya bat hai is name ki,", name)

# Example of outputting multiple variables with print()
age = 19
print("Name:", name, "Age:", age)

# Example of using f-strings to format output
country = "Pakistan"
print(f"{name} is from {country}")

# Example of using f-strings to manipulate variables in output
num1 = 19
num2 = 28
result = num1 + num2
print(f"The sum of {num1} and {num2} is {result}")
```

```
kya bat hai is name ki, Amish
Name: Amish Age: 19
Amish is from Pakistan
The sum of 19 and 28 is 47
```