PYTHON IS FOR EVERYONE

Tutorial 9:

PYTHON PROGRAMMING - STRING MANIPULATION IN GOOGLE COLAB

Jeff Gentry

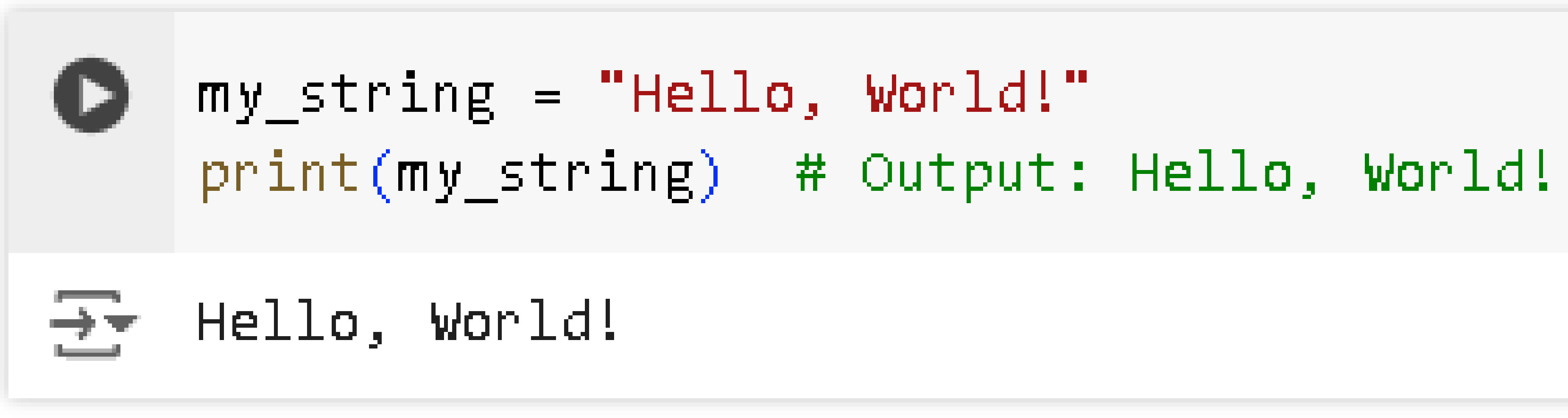
@www.linkedin.com/in/jefferycharlesgentry

Objectives

- Understand string data types and how to manipulate them.
- Learn about common string methods.
- Explore string formatting techniques.
- Practice string manipulation through hands-on exercises.

Understanding Strings

A string is a sequence of characters enclosed in quotes (single, double, or triple quotes). Strings are immutable, meaning they cannot be changed after they are created.



```
my_string = "Hello, World!"
print(my_string)  # Output: Hello, World!
```

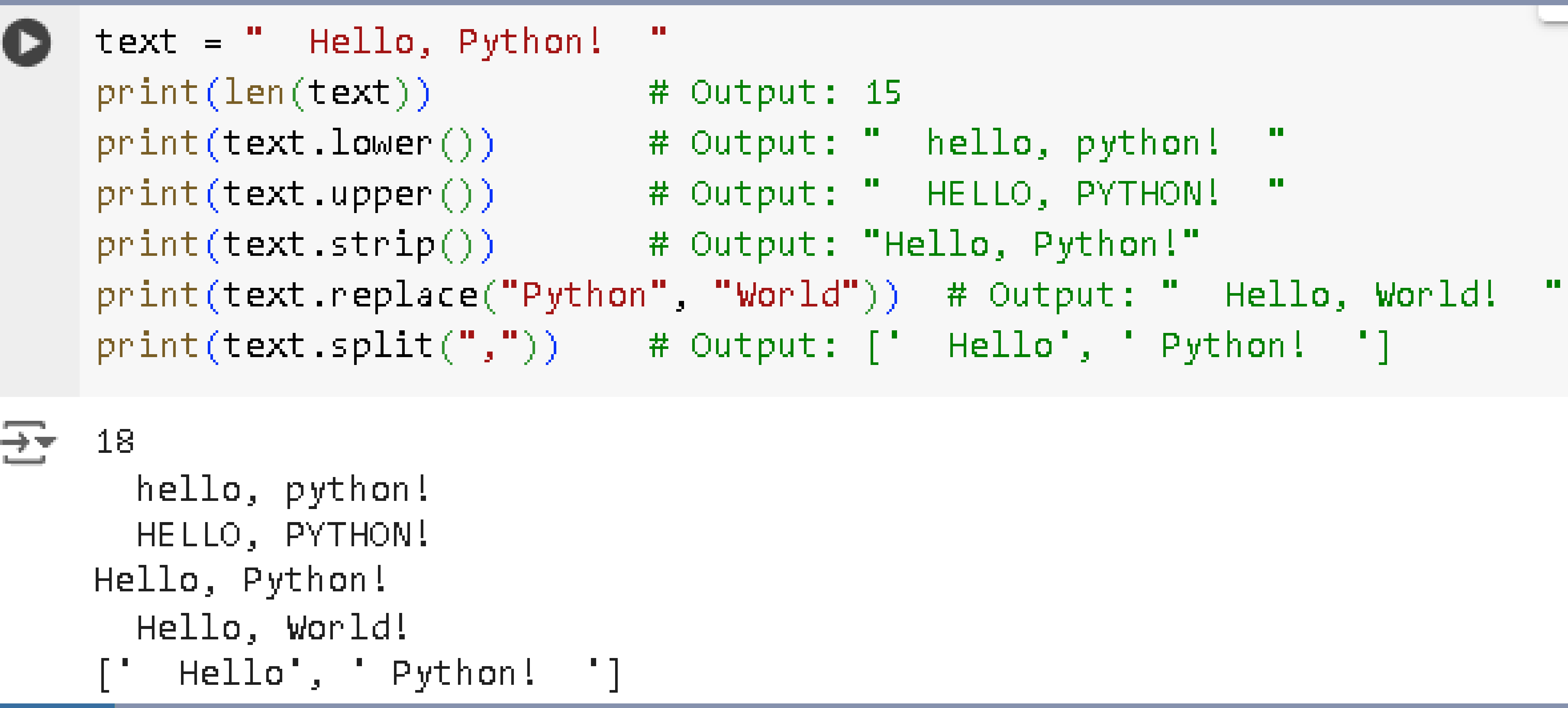
```
Hello, World!
```

Common String Methods

Python provides several built-in methods for string manipulation. Here are some commonly used string methods:

- len(): Returns the length of the string.
- lower(): Converts the string to lowercase.
- upper(): Converts the string to uppercase.
- strip(): Removes leading and trailing whitespace.
- replace(old, new): Replaces occurrences of a substring with another substring.
- split(separator): Splits the string into a list based on a separator.
- join(iterable): Joins elements of an iterable into a single string.

Examples



```
text = "  Hello, Python!  "
print(len(text))          # Output: 15
print(text.lower())        # Output: "  hello, python!  "
print(text.upper())        # Output: "  HELLO, PYTHON!  "
print(text.strip())        # Output: "Hello, Python!"
print(text.replace("Python", "World"))  # Output: "  Hello, World!  "
print(text.split(","))    # Output: ['  Hello', ' Python!  ']
```

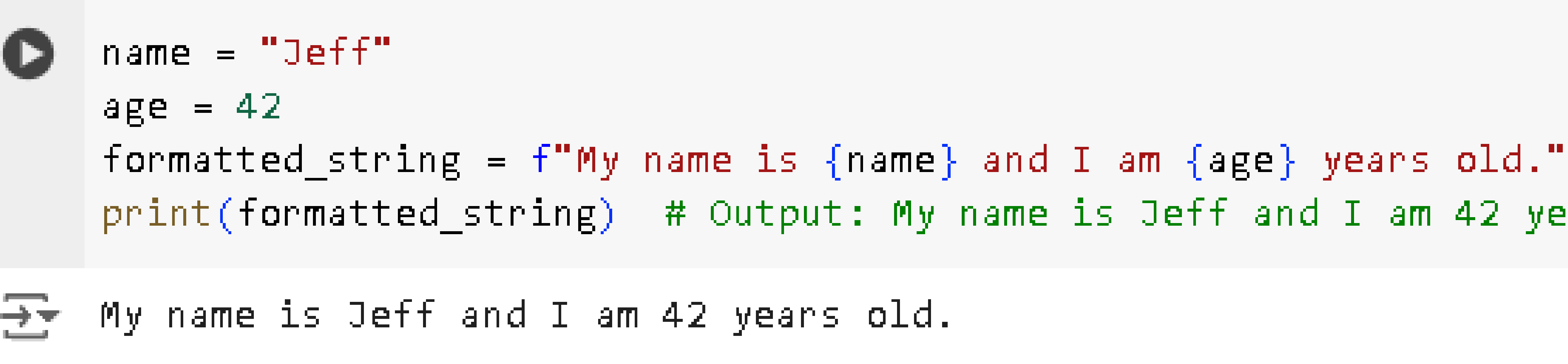
```
18
  hello, python!  
  HELLO, PYTHON!  
Hello, Python!
  Hello, World!  
['  Hello', ' Python!  ']
```

String Formatting

String formatting allows you to create strings that include variables or expressions. There are several ways to format strings in Python:

- f-strings (Python 3.6+)
- the “format()” method
- the % operator

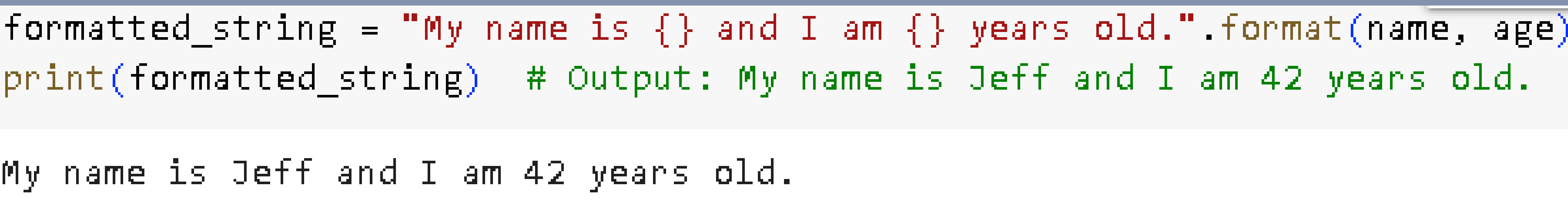
Using f-strings (Python 3.6+)



```
name = "Jeff"
age = 42
formatted_string = f"My name is {name} and I am {age} years old."
print(formatted_string)  # Output: My name is Jeff and I am 42 ye
```

```
My name is Jeff and I am 42 years old.
```

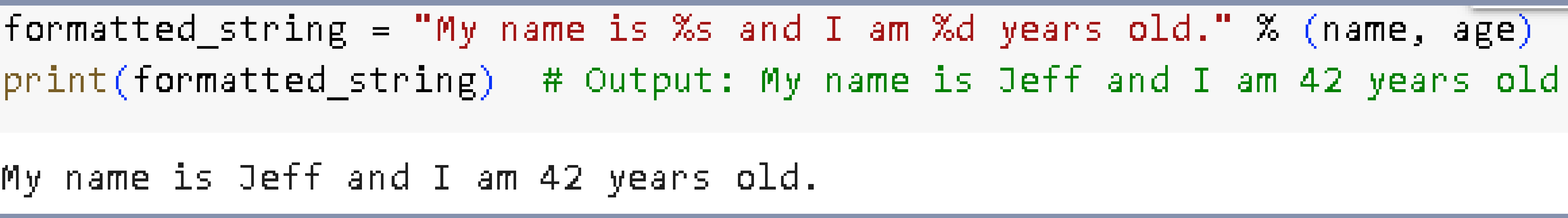
Using the “format()” method



```
formatted_string = "My name is {} and I am {} years old.".format(name, age)
print(formatted_string)  # Output: My name is Jeff and I am 42 years old.
```

```
My name is Jeff and I am 42 years old.
```

Using the % operator

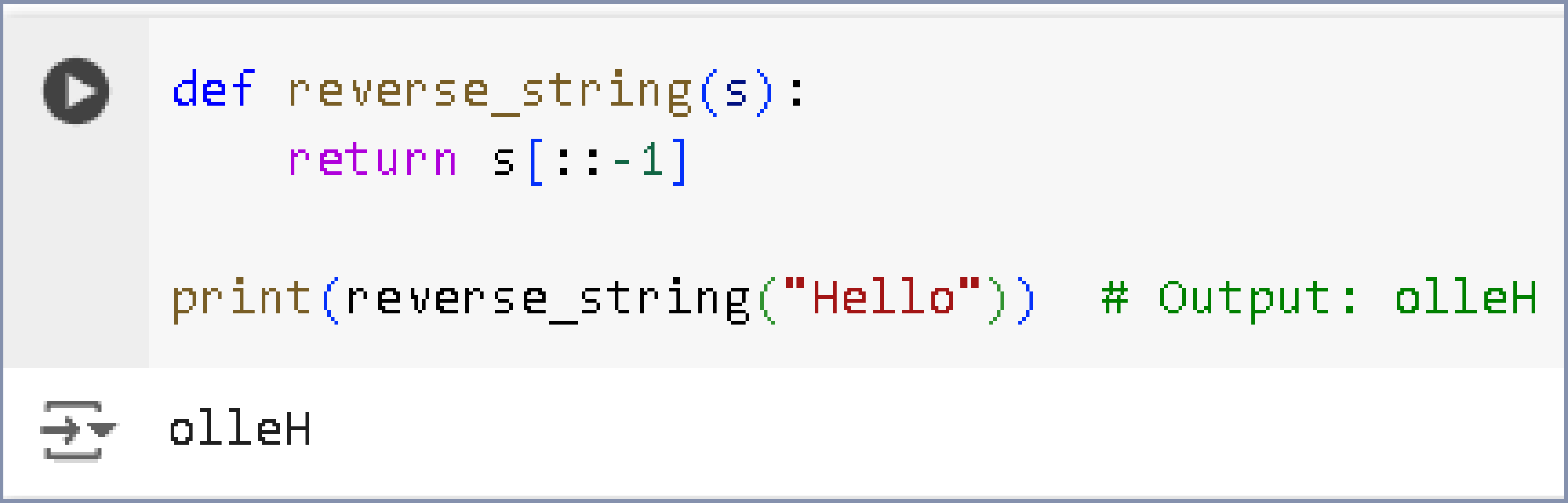


```
formatted_string = "My name is %s and I am %d years old." % (name, age)
print(formatted_string)  # Output: My name is Jeff and I am 42 years old
```

```
My name is Jeff and I am 42 years old.
```

Practice Exercises

Reverse a String: Write a program that reverses a given string.



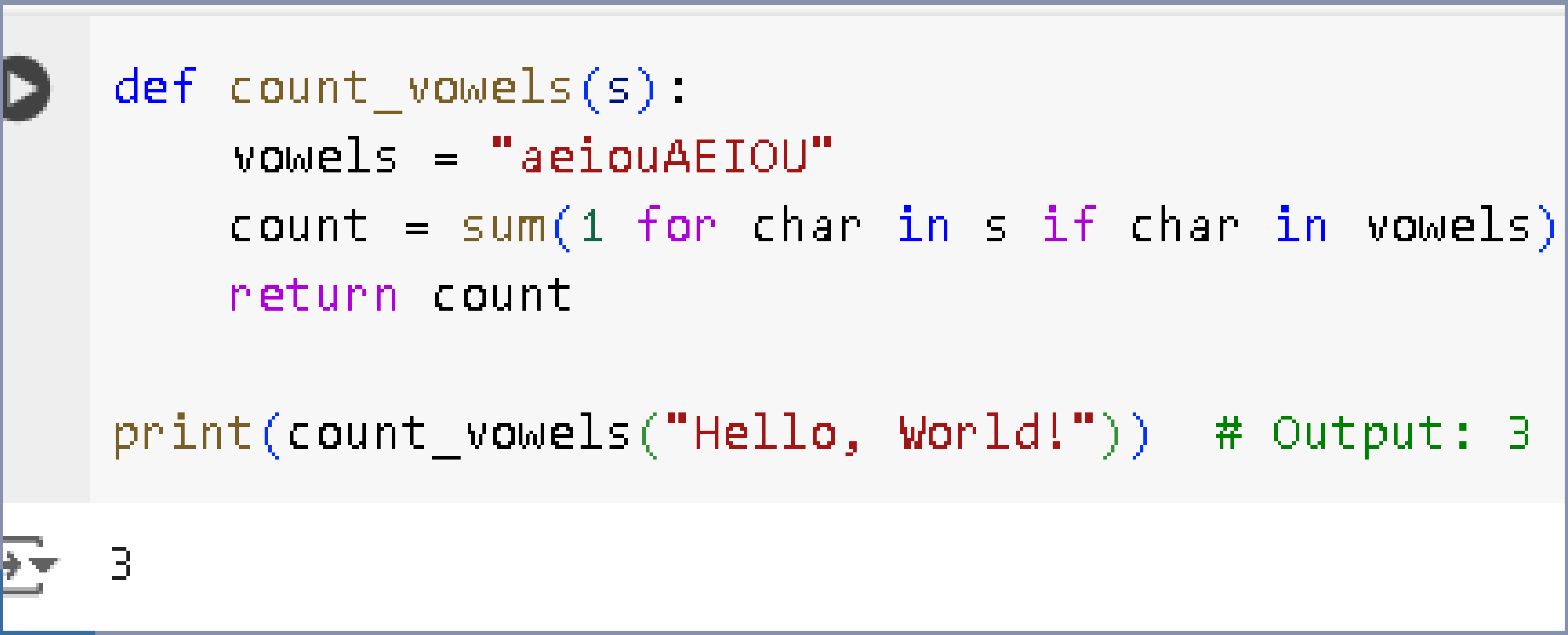
```python
def reverse_string(s):
    return s[::-1]

print(reverse_string("Hello"))  # Output: olleH
```

```
olleH
```

Practice Exercises

Count Vowels: Write a program that counts the number of vowels in a string.



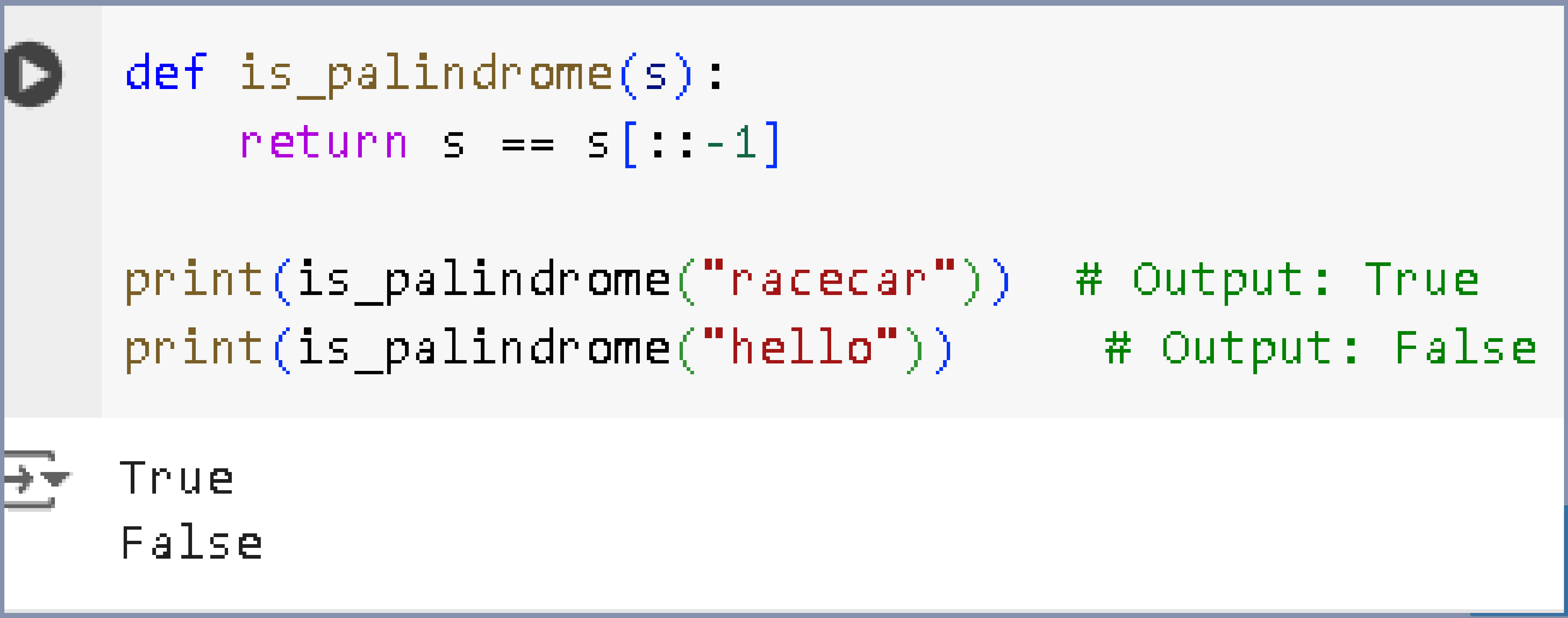
```python
def count_vowels(s):
    vowels = "aeiouAEIOU"
    count = sum(1 for char in s if char in vowels)
    return count

print(count_vowels("Hello, World!"))  # Output: 3
```

```
3
```

Practice Exercises

Check Palindrome: Write a program that checks if a string is a palindrome (reads the same forwards and backwards).



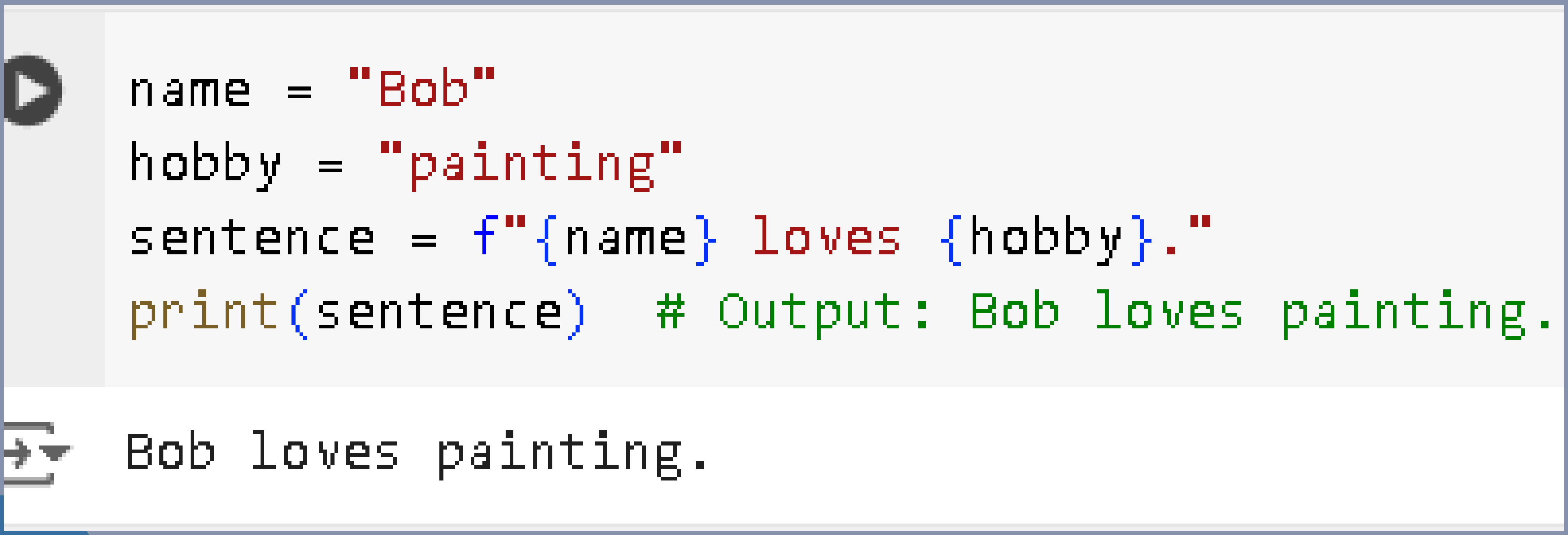
```python
def is_palindrome(s):
    return s == s[::-1]

print(is_palindrome("racecar"))  # Output: True
print(is_palindrome("hello"))     # Output: False
```

```
True
False
```

Practice Exercises

Format a Sentence: Write a program that formats a sentence using variables.

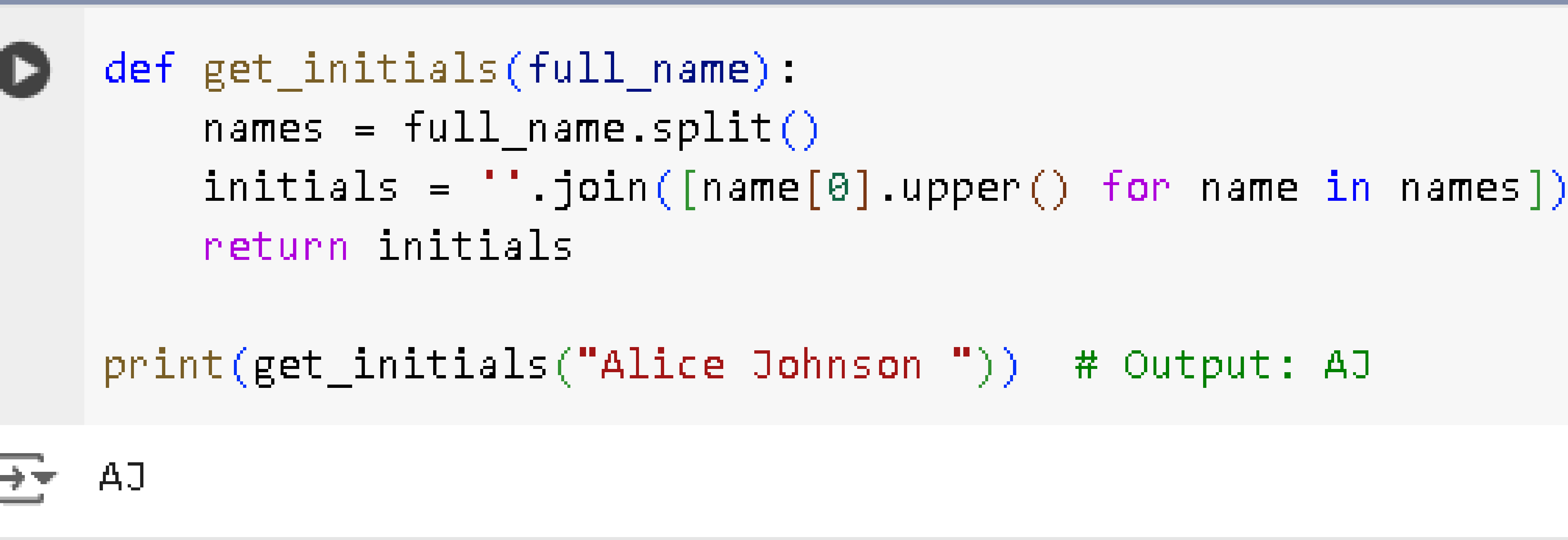


```python
name = "Bob"
hobby = "painting"
sentence = f"{name} loves {hobby}."
print(sentence)  # Output: Bob loves painting.
```

```
Bob loves painting.
```

Practice Exercises

Extract Initials: Write a program that extracts the initials from a full name.



```python
def get_initials(full_name):
    names = full_name.split()
    initials = ''.join([name[0].upper() for name in names])
    return initials

print(get_initials("Alice Johnson "))  # Output: AJ
```

```
AJ
```

Conclusion

In this tutorial, you learned about string manipulation in Python, including common string methods and various formatting techniques. String manipulation is a crucial skill in programming, allowing you to handle and process text data effectively.

Next Steps

In tutorial 10, we will cover how to read from and write to files, which is essential for data persistence in applications.

FOLLOW ME

for more tips you didn’t know you needed

Jeff Gentry

@www.linkedin.com/in/jefferycharlesgentry