Unit 4 :Strings

Syllabus :

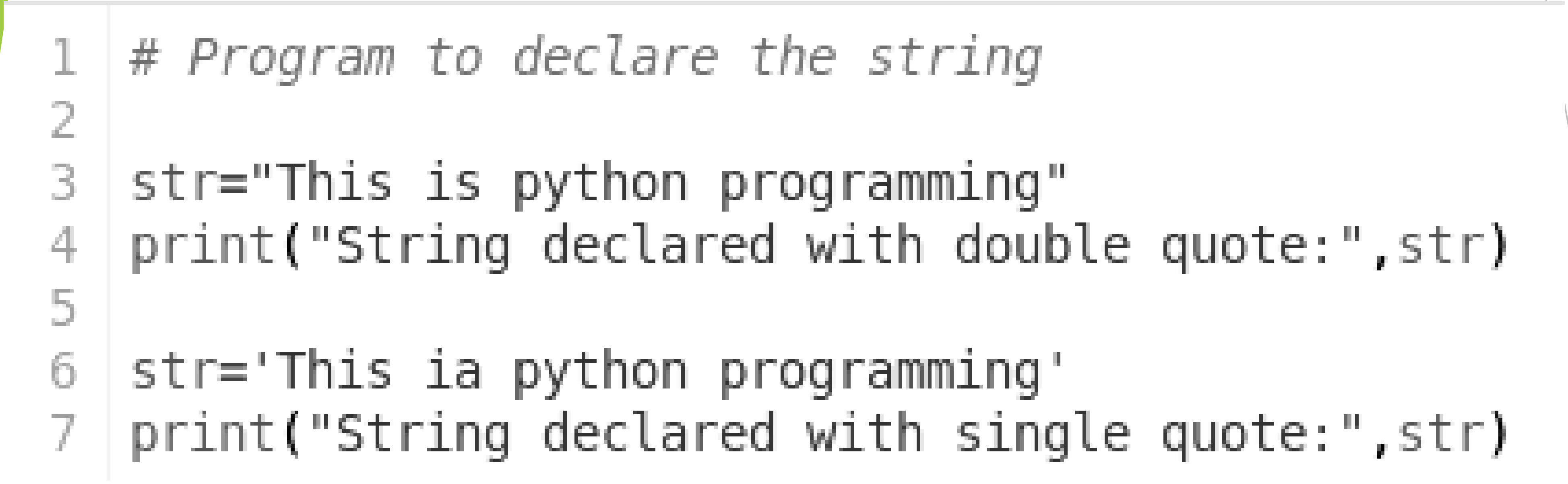
Strings and Operations :

- ✓ concatenation, appending, multiplication and slicing.
- ✓ Strings are immutable, strings formatting operator,
- ✓ built in string methods and functions.
- ✓ Slice operation, ord() and chr() functions, in and not in operators,
- ✓ comparing strings,
- ✓ Iterating strings,
- ✓ the string module.

Strings :

- A Python string is a sequence of zero or more characters.
- There is a unique code provided to all existing characters.
- Space, comma everything inside those quotes will be a character.
- There is a built-in class ‘str’ for handling Python string. You can prove this with the type() function.

Strings :



```python
# Program to declare the string

str="This is python programming"
print("String declared with double quote:",str)

str='This ia python programming'
print("String declared with single quote:",str)
```

Strings :

String Accessing

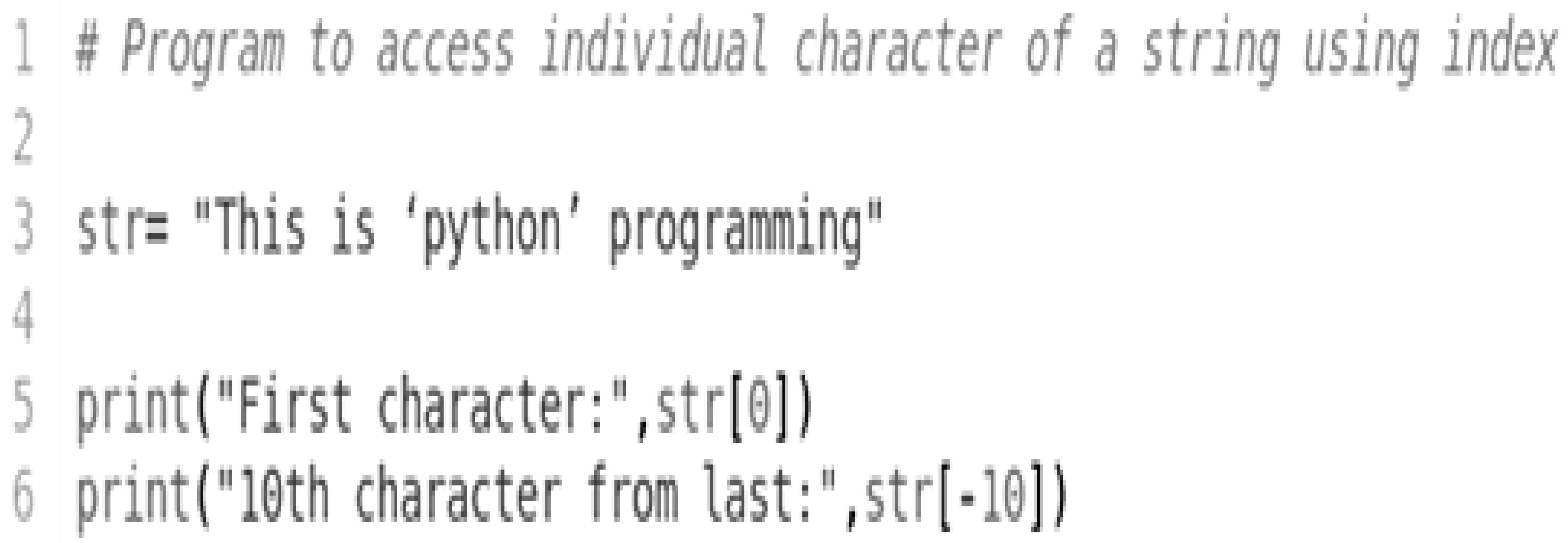
- We can access each individual character of a string by using index.
- Each character can be accessed in two ways - from the front, or from the rear end.

Strings :

String Accessing

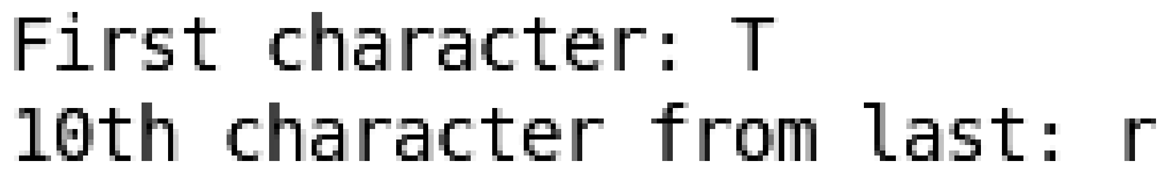
• Characters	P	Y	T	H	O	N
• Forward index	0	1	2	3	4	5
• Backward index	-6	-5	-4	-3	-2	-1

Strings :



```python
# Program to access individual character of a string using index

str= "This is 'python' programming"

print("First character:",str[0])
print("10th character from last:",str[-10])
```

Output :

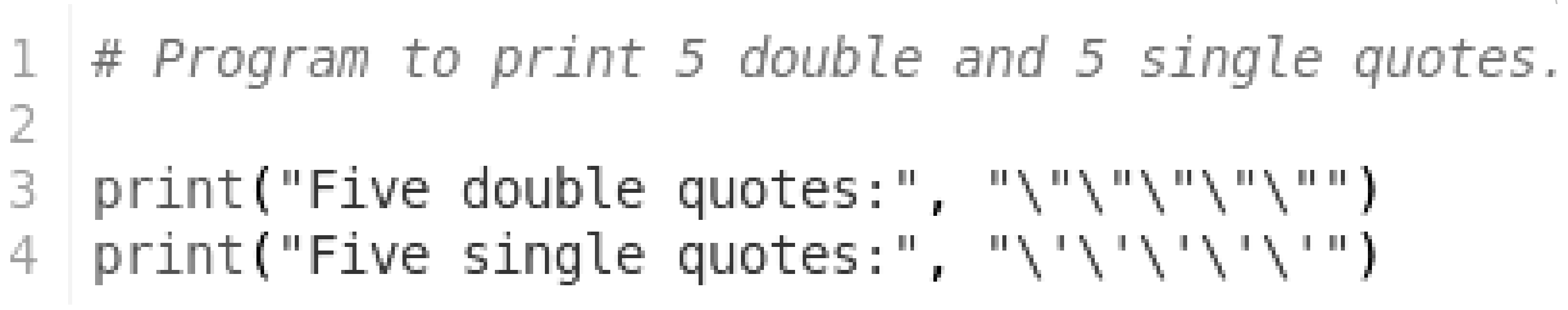


```
First character: T
10th character from last: r
```

Escape Sequence or a Back-slash:

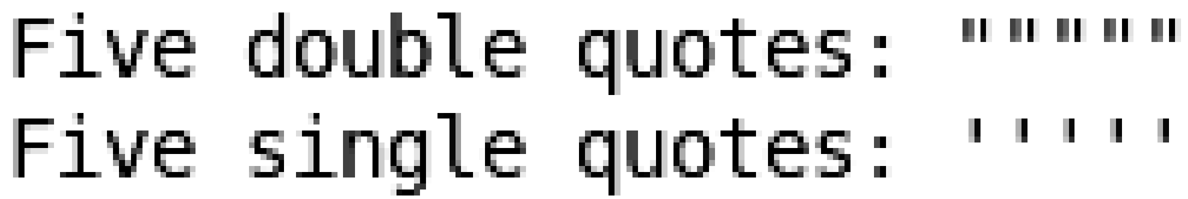
- **escape sequence or a back-slash will inform the compiler to simply print whatever you type and not try to compile it.**
- **You must use one escape sequence for one character. For example, in order to print 5 double quotes, we will have to use 5 backslashes, one before each quote.**
- **Python also supports the \n - linefeed and \t - tab sequences.**

Strings :



```python
# Program to print 5 double and 5 single quotes.

print("Five double quotes:", "\"\"\"\"\"")
print("Five single quotes:", "\'\'\'\'\'")
```

Output:



```
Five double quotes: """""
Five single quotes: '''''
```

String Slicing :

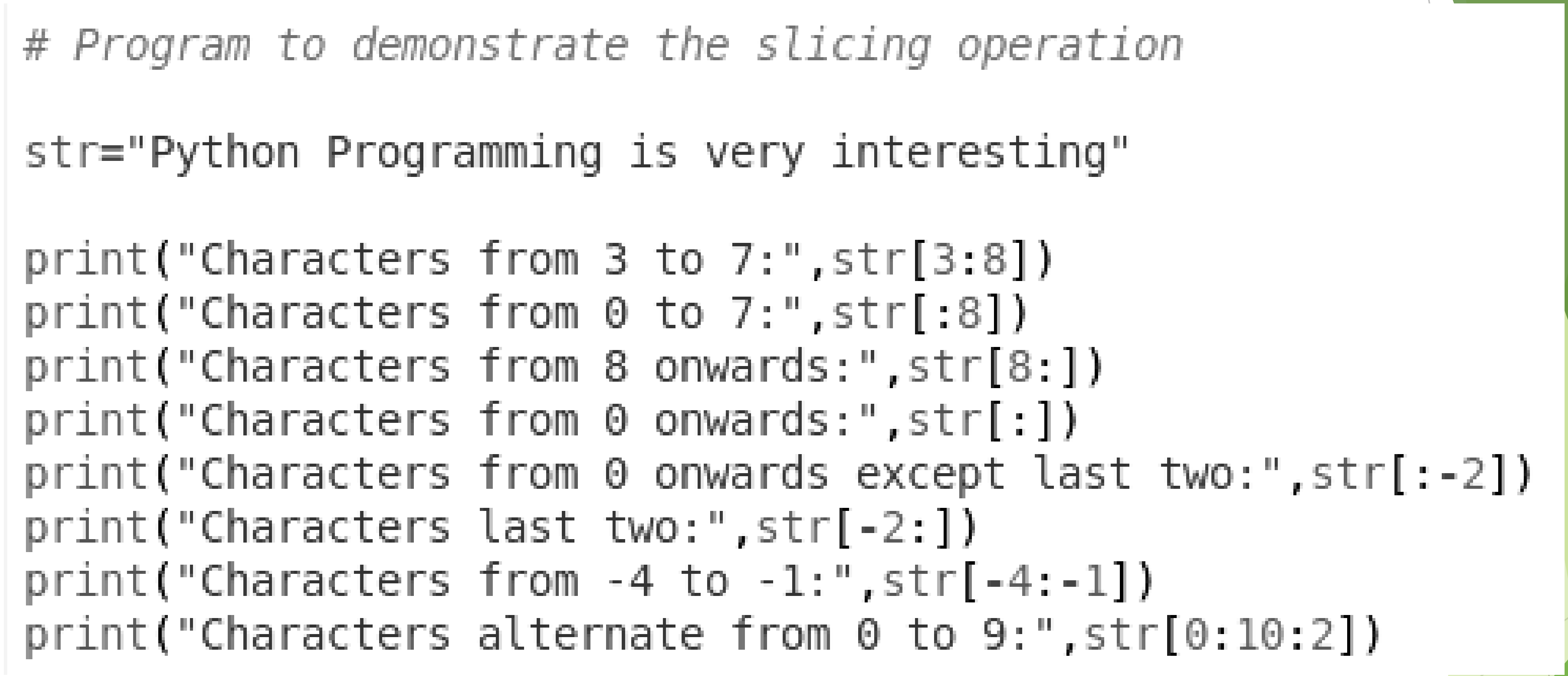
- Slicing is a string operation.
- Slicing is used to extract a part of any string based on a start index and an end index.
- The extracted portions of Python strings called substrings.
- In slicing colon : is used. An integer value will appear on either side of colon.

String Slicing :

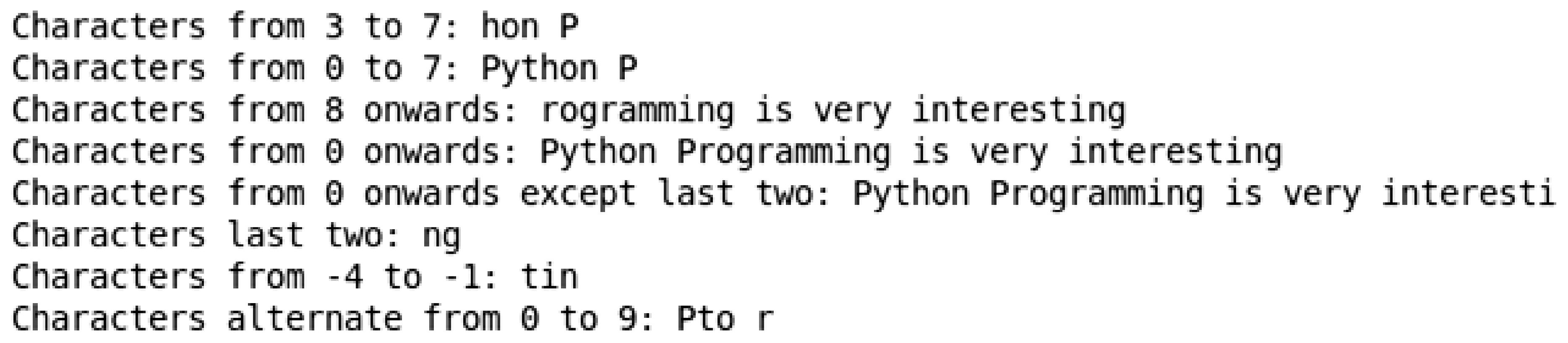
Syntax :

string_name[starting_index : finishing_index : character_iterate]

- **String_name is the name of the variable holding the string.**
- **starting_index is the index of the beginning character which you want in your sub-string.**
- **finishing_index is one more than the index of the last character that you want in your substring.**
- **character_iterate is how many characters to extract in each jump.**



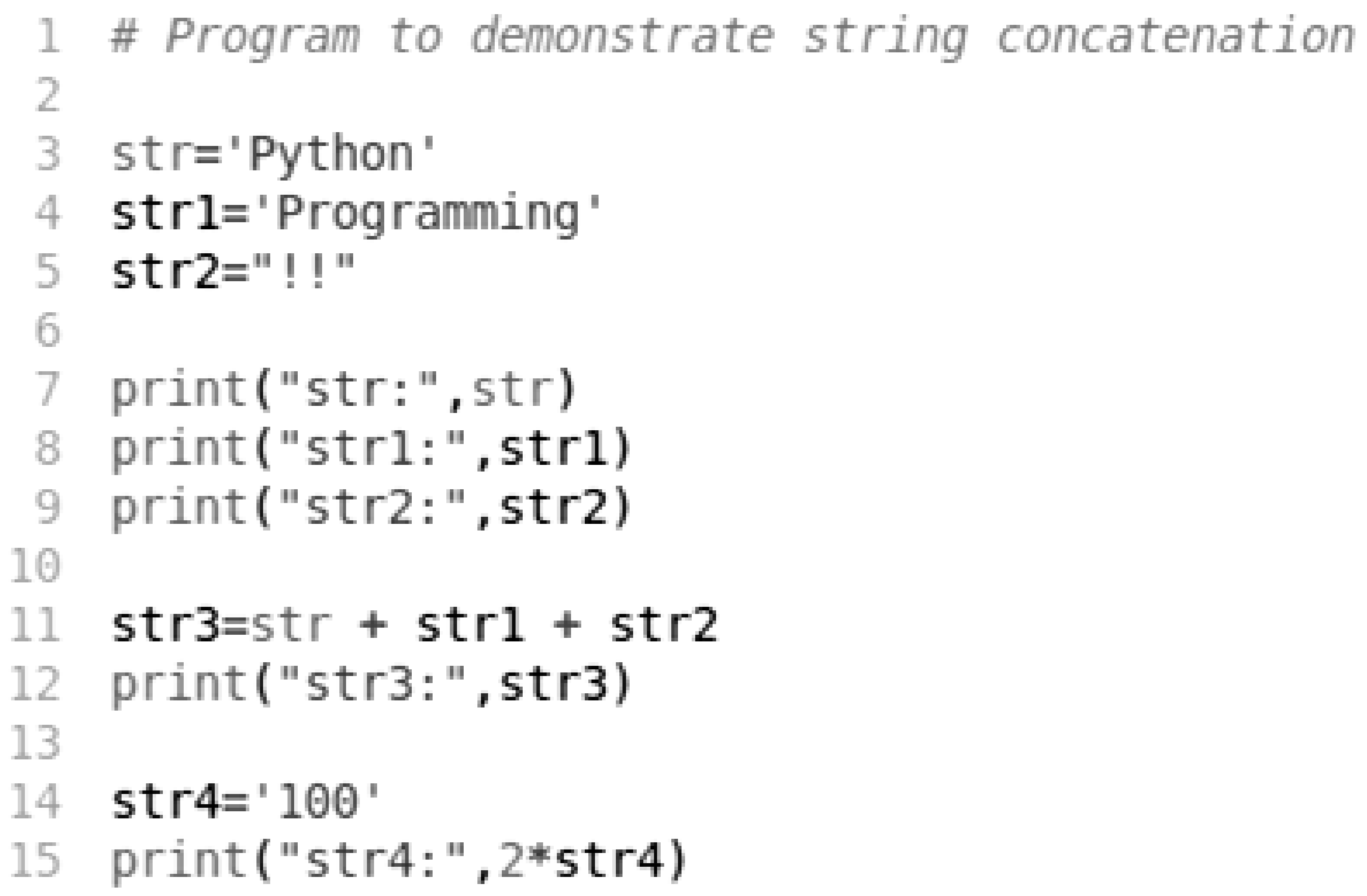
```python
# Program to demonstrate the slicing operation

str="Python Programming is very interesting"

print("Characters from 3 to 7:",str[3:8])
print("Characters from 0 to 7:",str[:8])
print("Characters from 8 onwards:",str[8:])
print("Characters from 0 onwards:",str[:])
print("Characters from 0 onwards except last two:",str[:-2])
print("Characters last two:",str[-2:])
print("Characters from -4 to -1:",str[-4:-1])
print("Characters alternate from 0 to 9:",str[0:10:2])
```



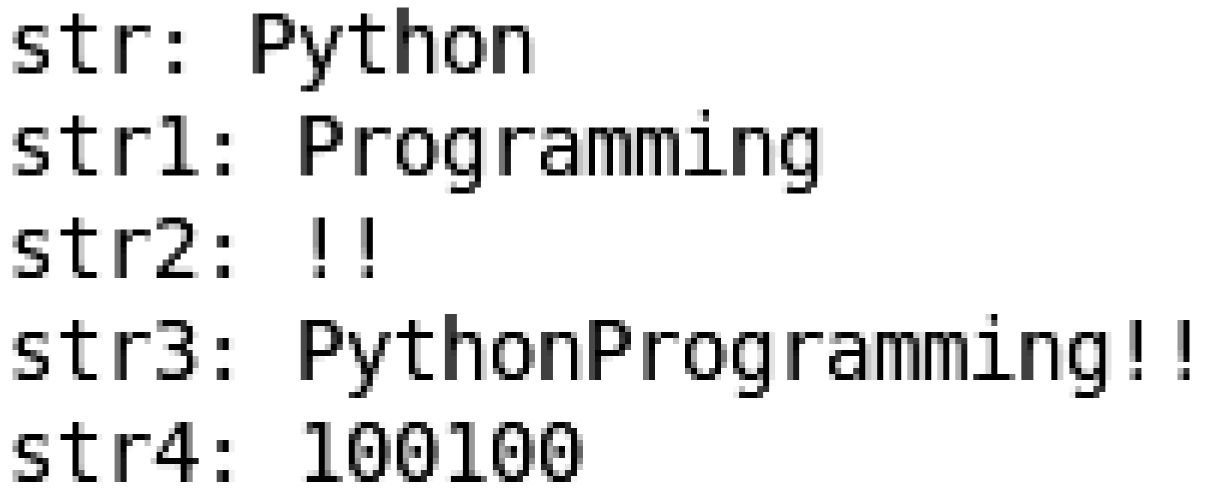
```
Characters from 3 to 7: hon P
Characters from 0 to 7: Python P
Characters from 8 onwards: rogramming is very interesting
Characters from 0 onwards: Python Programming is very interesting
Characters from 0 onwards except last two: Python Programming is very interesti
Characters last two: ng
Characters from -4 to -1: tin
Characters alternate from 0 to 9: Pto r
```

String Concatenation :

- Concatenation is the operation of joining two strings together.
- Python Strings can join using the concatenation operator +.



```python
# Program to demonstrate string concatenation

str='Python'
str1='Programming'
str2="!!"

print("str:",str)
print("str1:",str1)
print("str2:",str2)

str3=str + str1 + str2
print("str3:",str3)

str4='100'
print("str4:",2*str4)
```

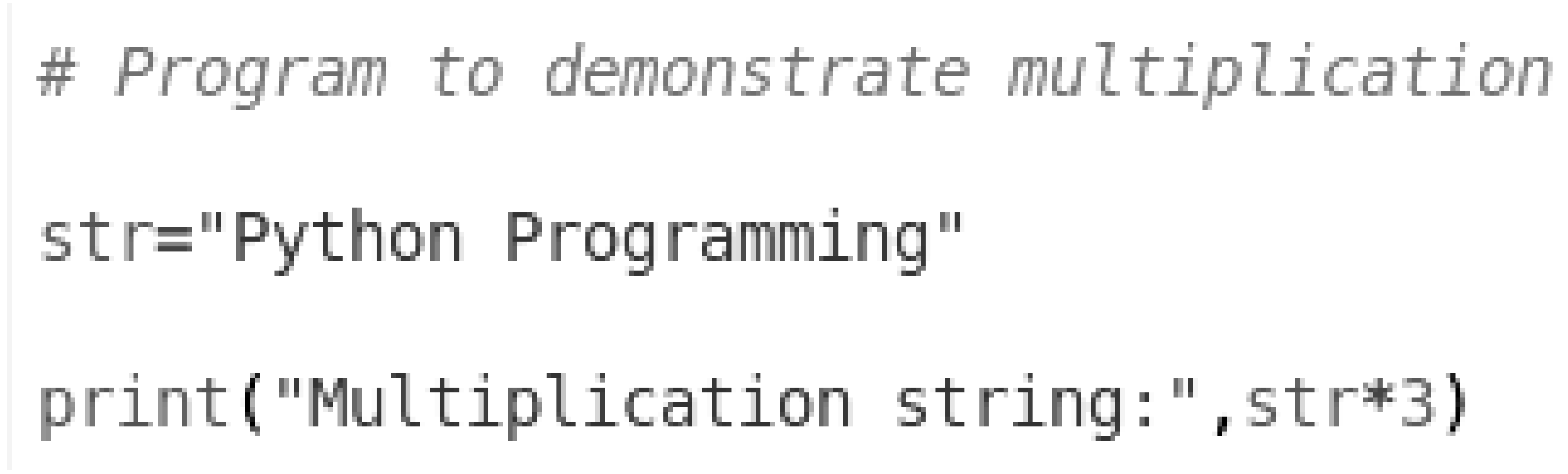


```
str: Python
str1: Programming
str2: !!
str3: PythonProgramming!!
str4: 100100
```

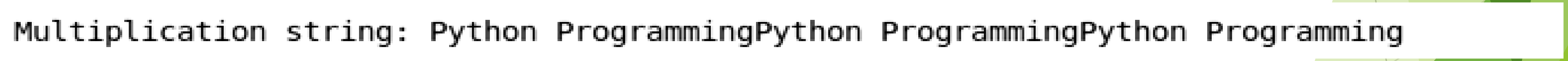
Multiplication or Repetition :

- To write the same text multiple times, multiplication or repetition is used.

- Multiplication can be done using operator



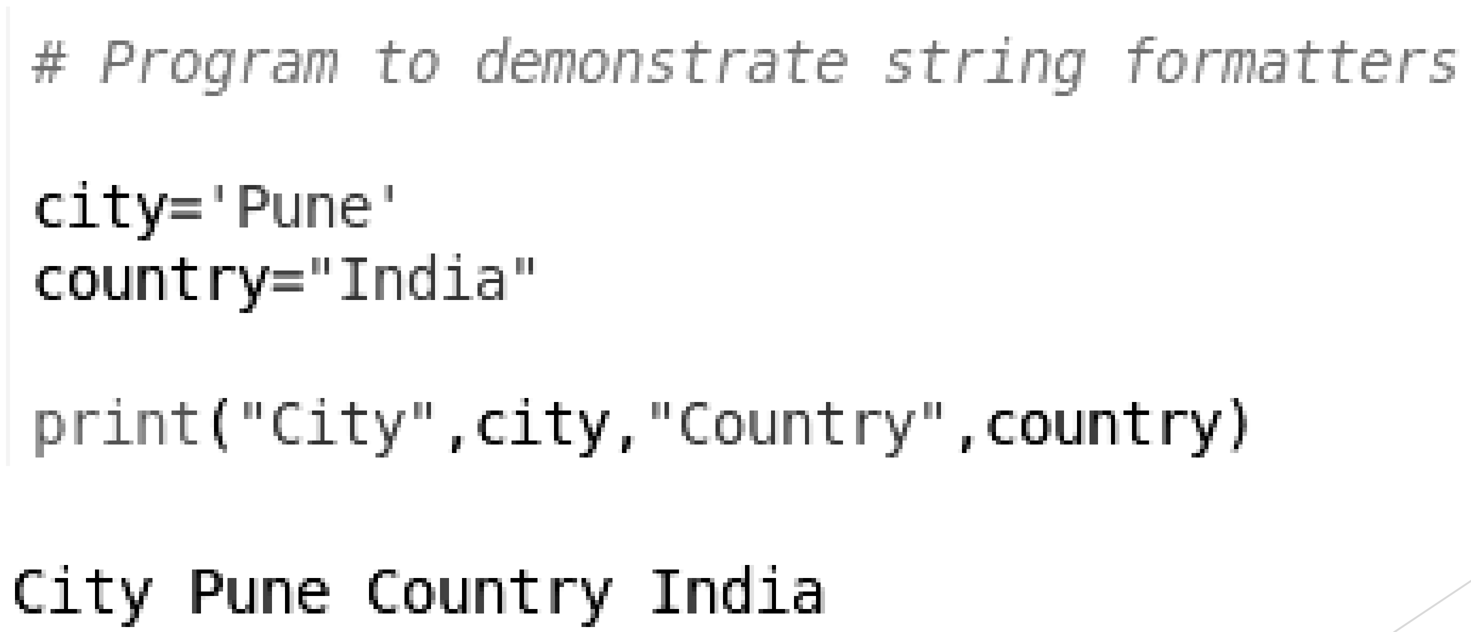
```python
# Program to demonstrate multiplication

str="Python Programming"

print("Multiplication string:",str*3)
```



```
Multiplication string: Python ProgrammingPython ProgrammingPython Programming
```

String Formatters :

- **Sometimes, you may want to print variables along with a string.**
- **You can either use commas, or use string formatters for the same.**

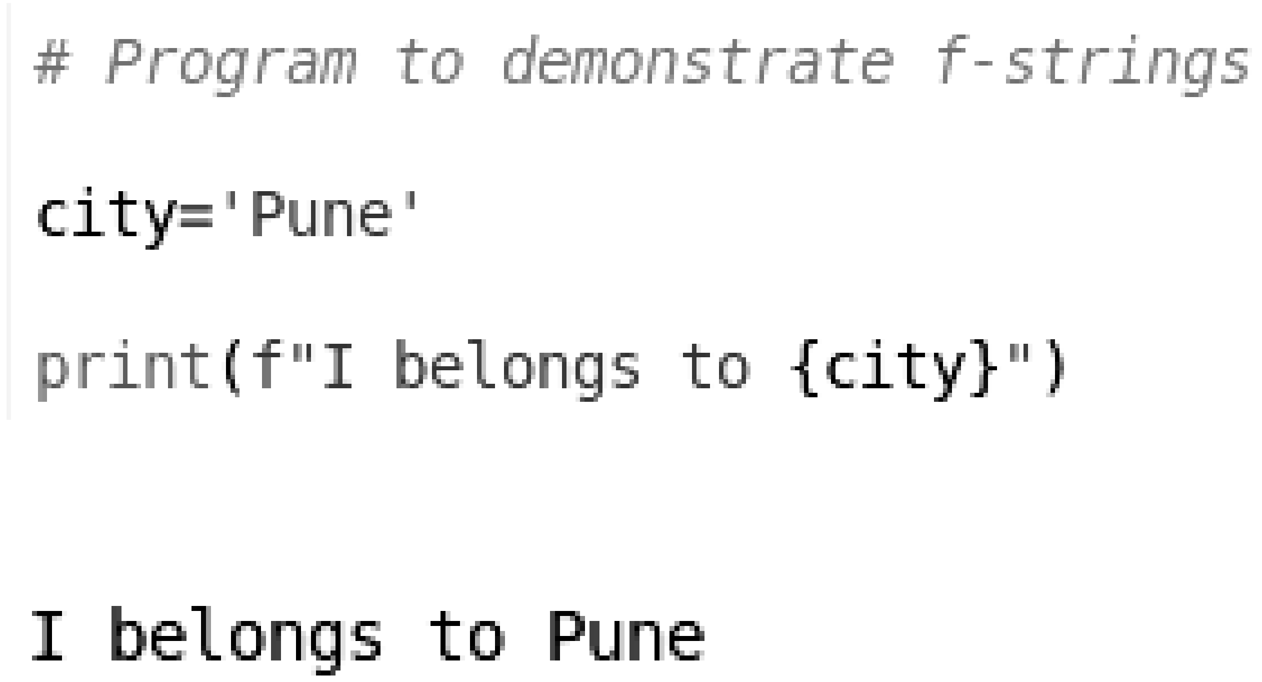


```
# Program to demonstrate string formatters

city='Pune'
country="India"

print("City",city,"Country",country)
```

```
City Pune Country India
```

String Formatters : f-strings

- The letter ‘f’ precedes the string, and the variables are mentioned in curly braces in their places.

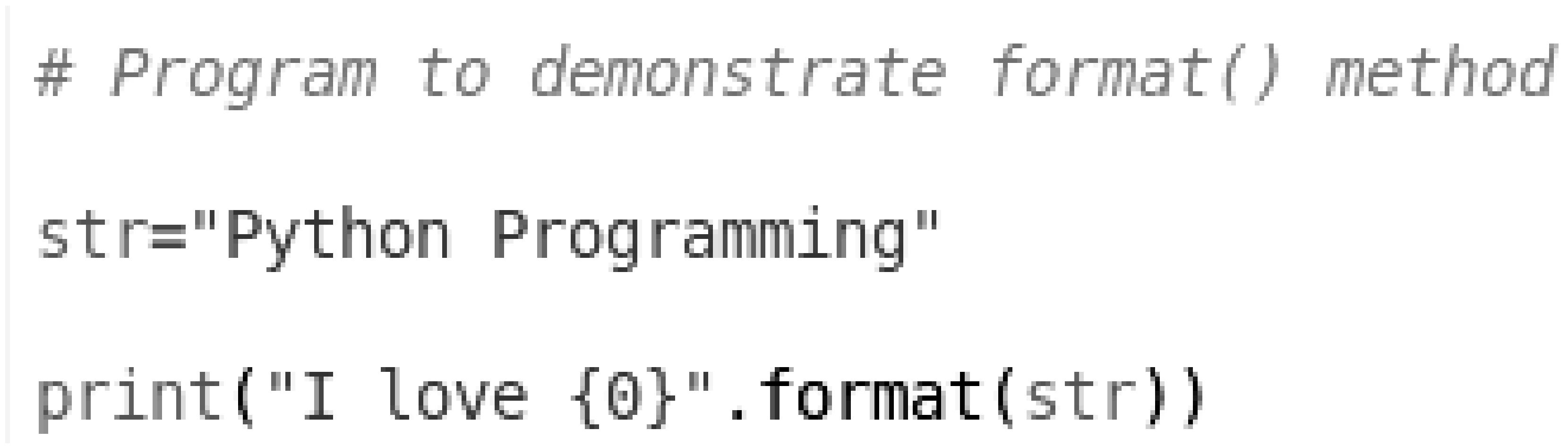


```python
# Program to demonstrate f-strings

city='Pune'

print(f"I belongs to {city}")
```

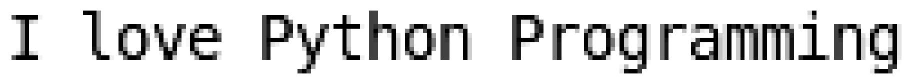
```
I belongs to Pune
```

String Formatters : Format() method

- You can use the format() method to do the same. It succeeds the string.
- It has the variables as arguments separated by commas. In the string, use curly braces to position the variables.
- Inside the curly braces, you can either put 0,1,.. or the variables.
- You must assign values to them in the format method.

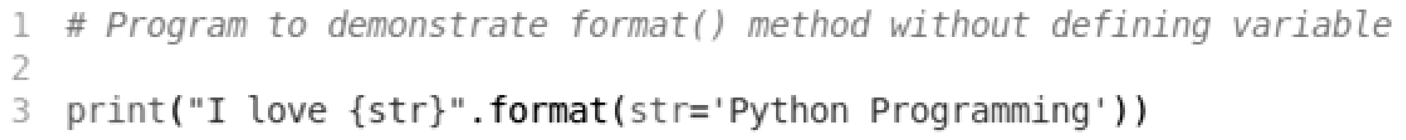


```python
# Program to demonstrate format() method

str="Python Programming"

print("I love {0}".format(str))
```

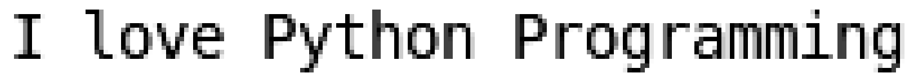


```
I love Python Programming
```

Program to demonstrate format() method without defining variable :



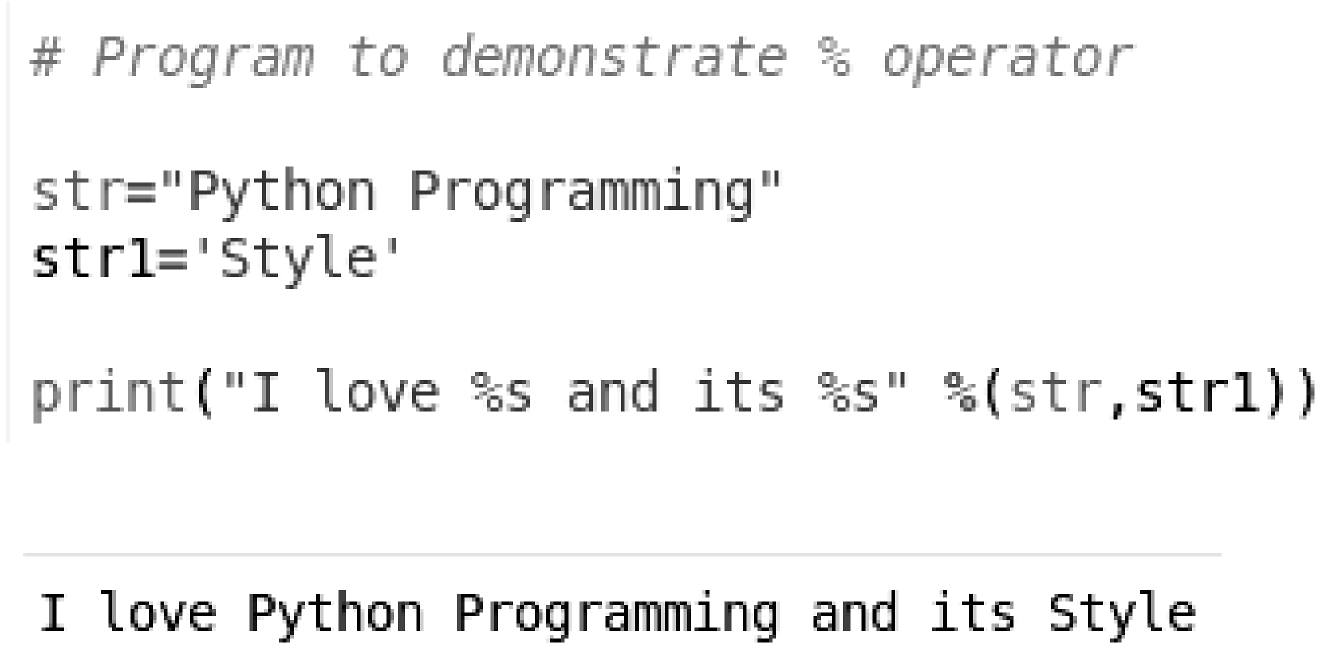
```python
# Program to demonstrate format() method without defining variable

print("I love {str}".format(str='Python Programming'))
```



```
I love Python Programming
```

String Formatters : % Operator

- The % operator goes where the variables go in a string. %s is for string. What follows the string is the operator and variables in parentheses in a tuple.

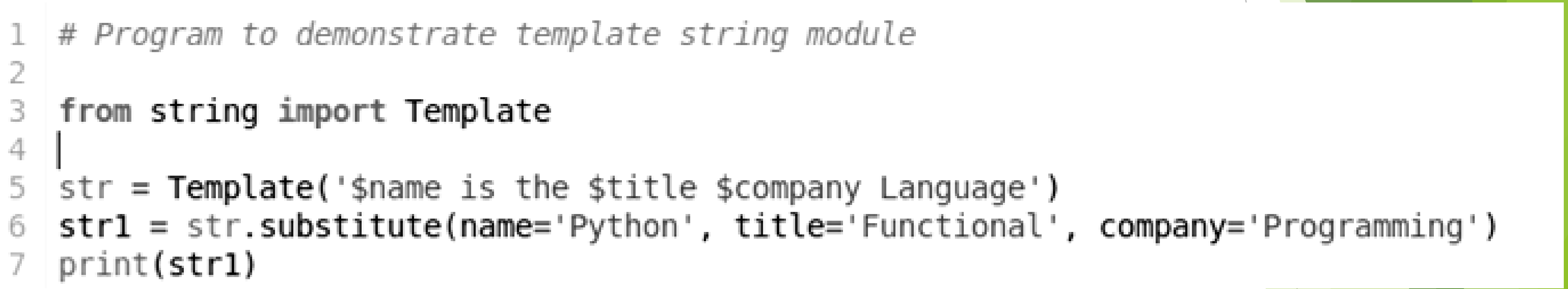


```
# Program to demonstrate % operator

str="Python Programming"
str1='Style'

print("I love %s and its %s" %(str,str1))
```

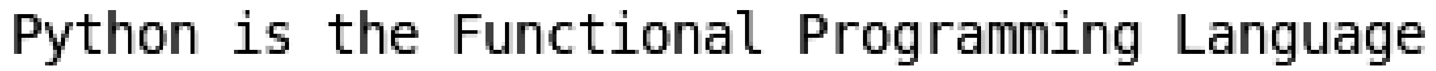
```
I love Python Programming and its Style
```

String Formatters : Template

This class is used to create a string template for simpler string substitutions



```python
# Program to demonstrate template string module

from string import Template

str = Template('$name is the $title $company Language')
str1 = str.substitute(name='Python', title='Functional', company='Programming')
print(str1)
```



```
Python is the Functional Programming Language
```

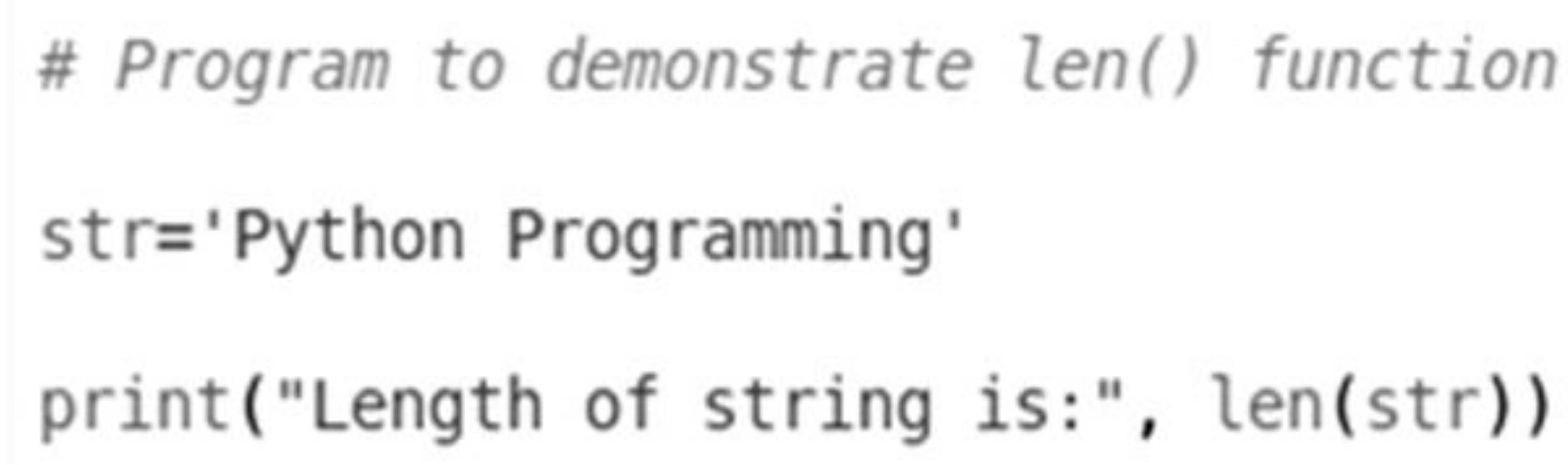
String Methods and Functions :

- Python provides us with a number of functions that we can apply on strings or to create strings.
- Different functions are described as below.

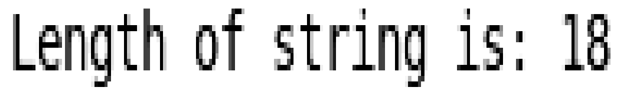
String Methods and Functions :

len() :

- **The len() function returns the length of a string**



```
# Program to demonstrate len() function

str='Python Programming'

print("Length of string is:", len(str))
```

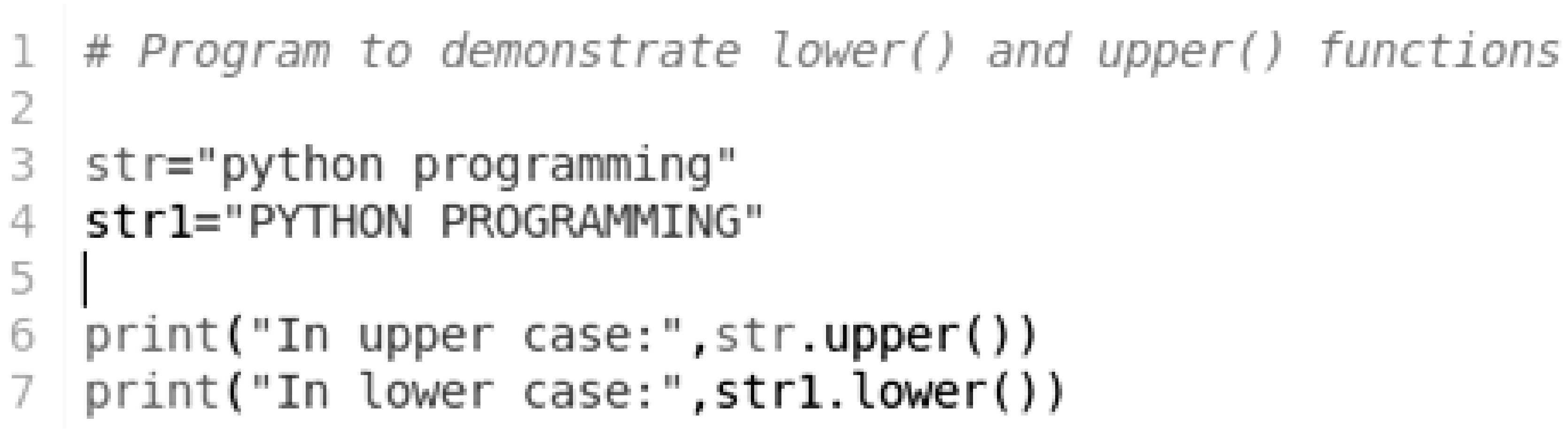


```
Length of string is: 18
```

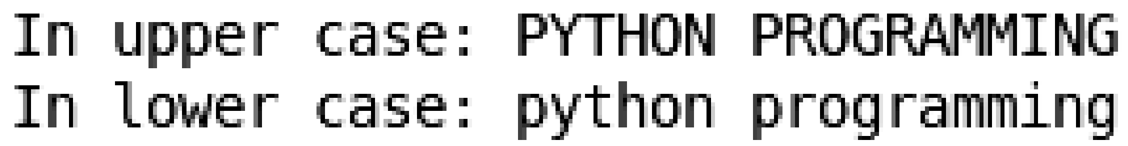
String Methods and Functions :

lower() and upper() :

- These methods return the string in lowercase and uppercase, respectively.



```
# Program to demonstrate lower() and upper() functions

str="python programming"
str1="PYTHON PROGRAMMING"

print("In upper case:",str.upper())
print("In lower case:",str1.lower())
```

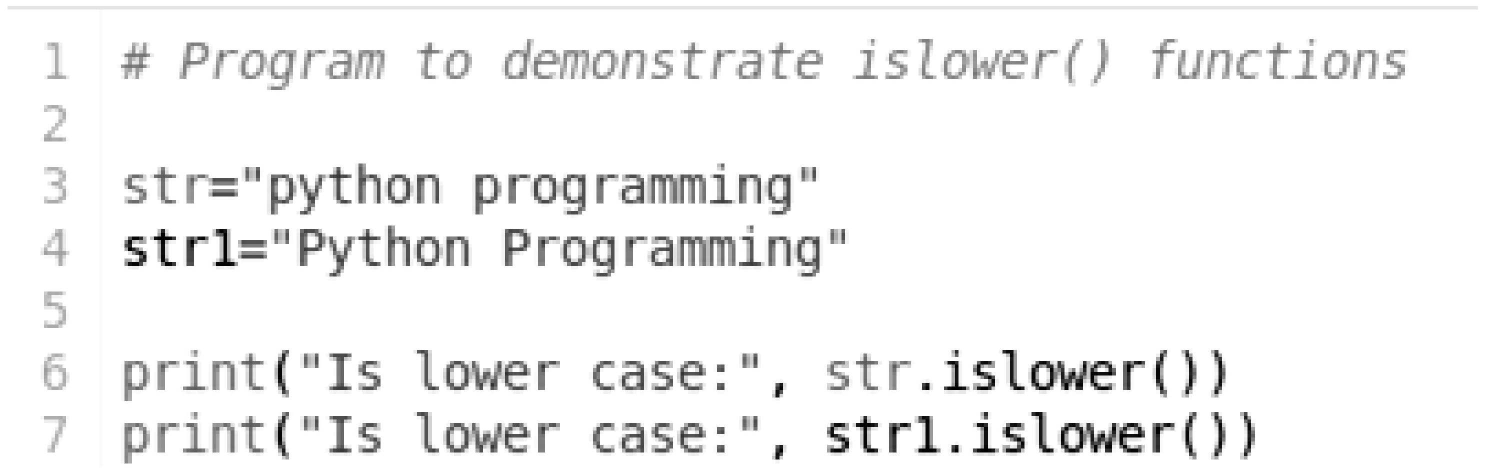


```
In upper case: PYTHON PROGRAMMING
In lower case: python programming
```

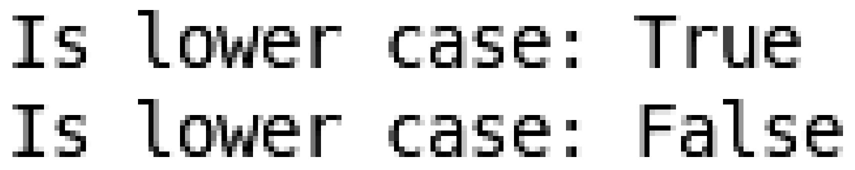
String Methods and Functions :

islower() :

- **The method islower() return true if the Python string contains only lower cased character(s) otherwise return false. .**



```
# Program to demonstrate islower() functions

str="python programming"
str1="Python Programming"

print("Is lower case:", str.islower())
print("Is lower case:", str1.islower())
```

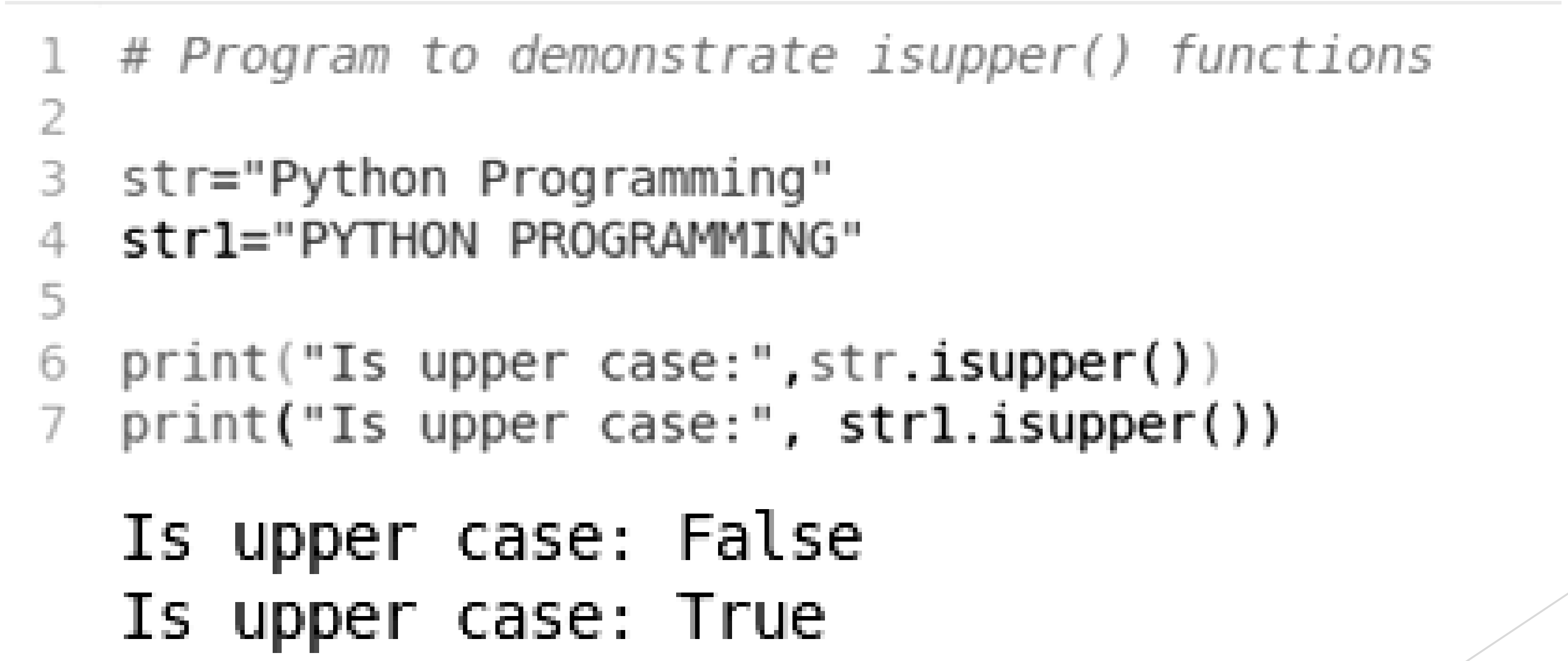


```
Is lower case: True
Is lower case: False
```

String Methods and Functions :

isupper() :

- **The method isupper() return true if the Python string contains only upper cased character(s) otherwise return false.**



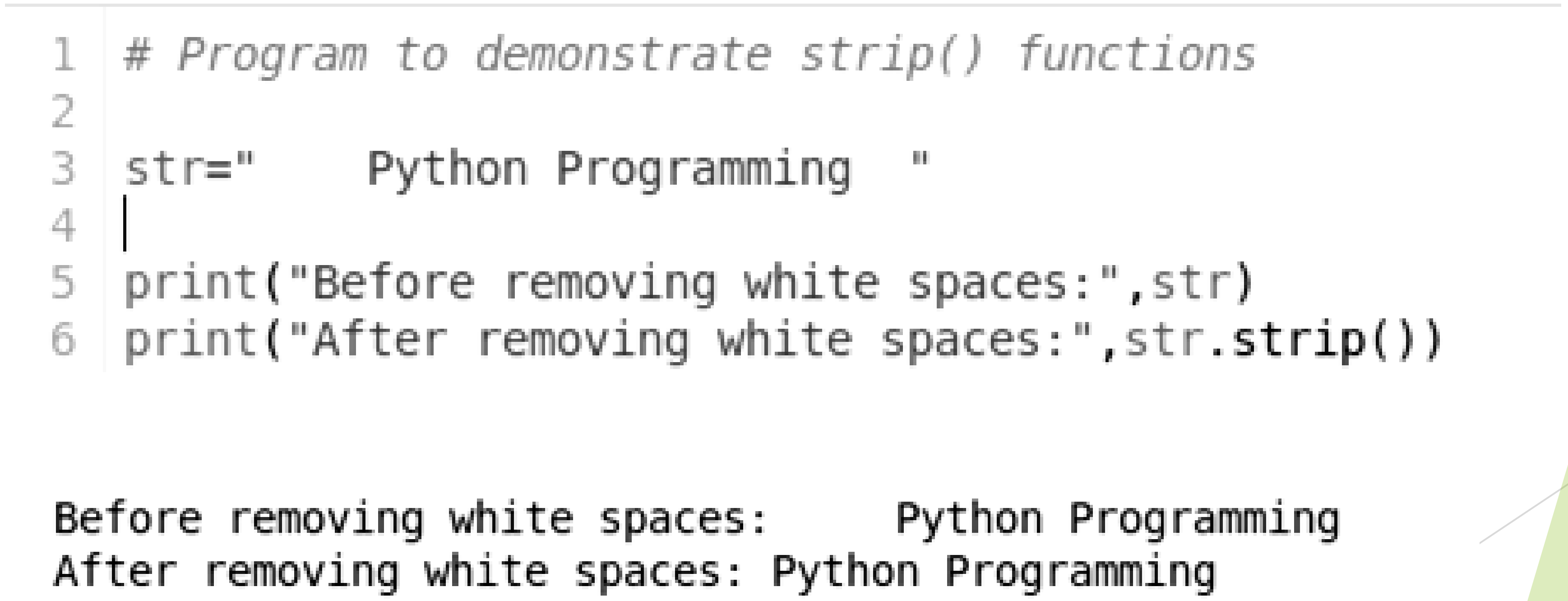
```
# Program to demonstrate isupper() functions

str="Python Programming"
str1="PYTHON PROGRAMMING"

print("Is upper case:",str.isupper())
print("Is upper case:", str1.isupper())
```

```
Is upper case: False
Is upper case: True
```

String Methods and Functions :

strip() :

- **It removes whitespaces from the beginning and end of the string.**



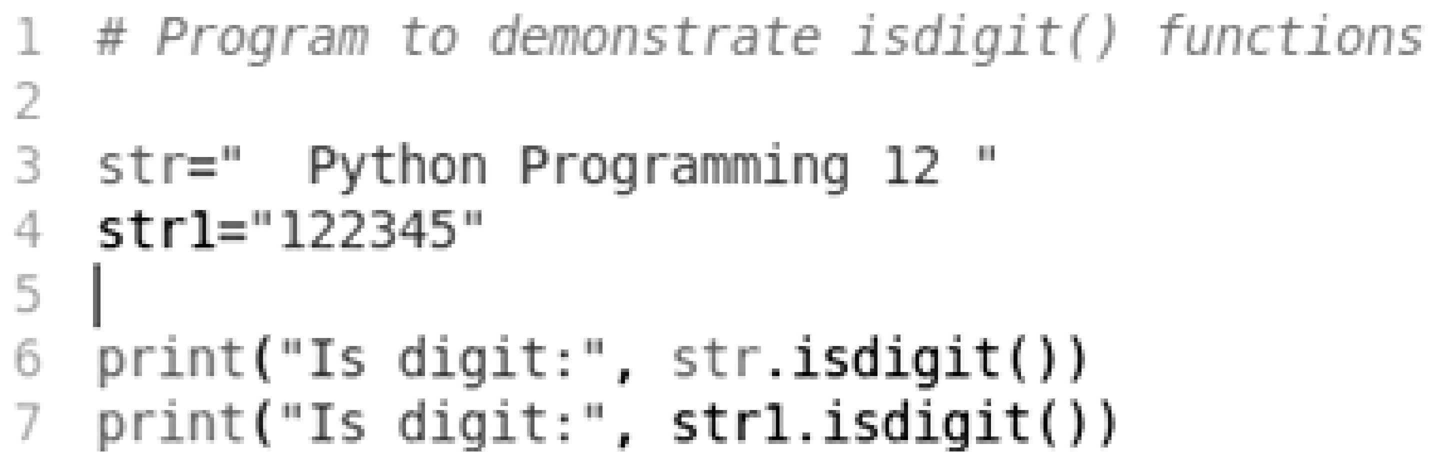
```python
# Program to demonstrate strip() functions

str="    Python Programming  "

print("Before removing white spaces:",str)
print("After removing white spaces:",str.strip())
```

```
Before removing white spaces:     Python Programming
After removing white spaces: Python Programming
```

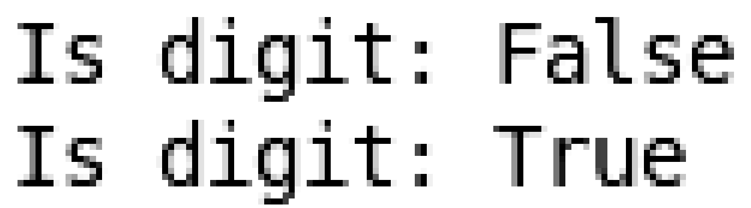
String Methods and Functions :

isdigit() :

- **Returns True if all characters in a string are digits.**



```
# Program to demonstrate isdigit() functions

str="  Python Programming 12 "
str1="122345"

print("Is digit:", str.isdigit())
print("Is digit:", str1.isdigit())
```

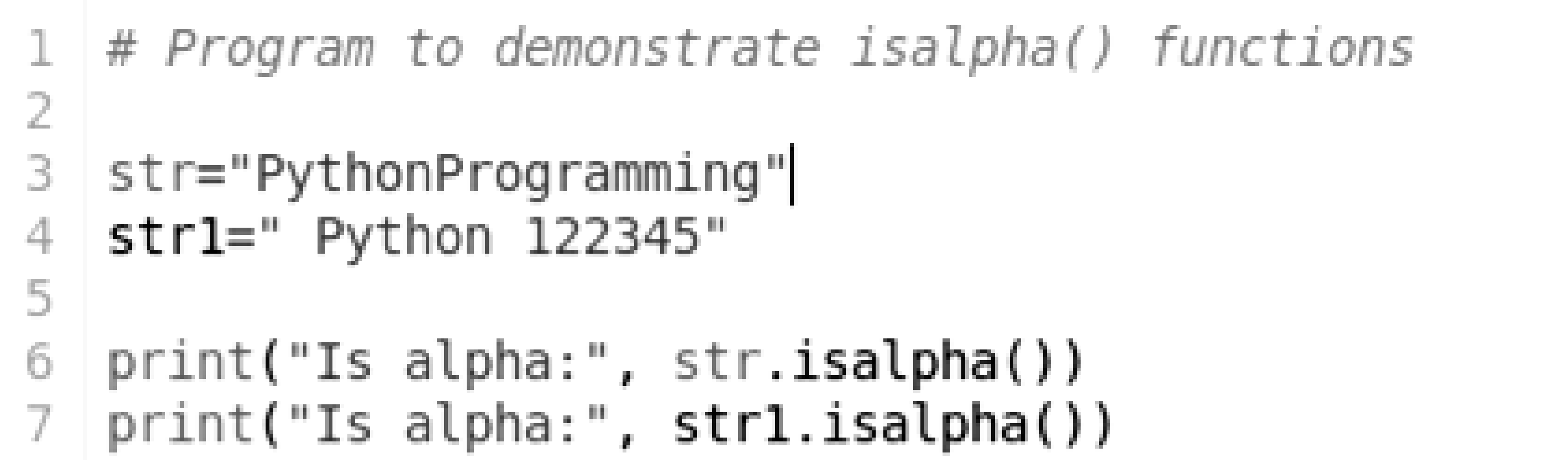


```
Is digit: False
Is digit: True
```

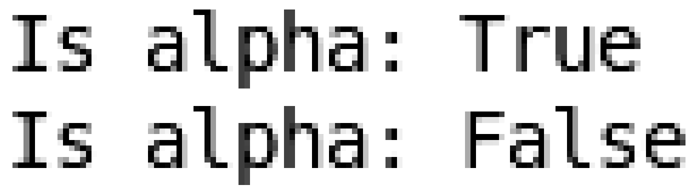
String Methods and Functions :

isalpha() :

- **Returns True if all characters in a string are characters from an alphabet.**



```
# Program to demonstrate isalpha() functions

str="PythonProgramming"
str1=" Python 122345"

print("Is alpha:", str.isalpha())
print("Is alpha:", str1.isalpha())
```

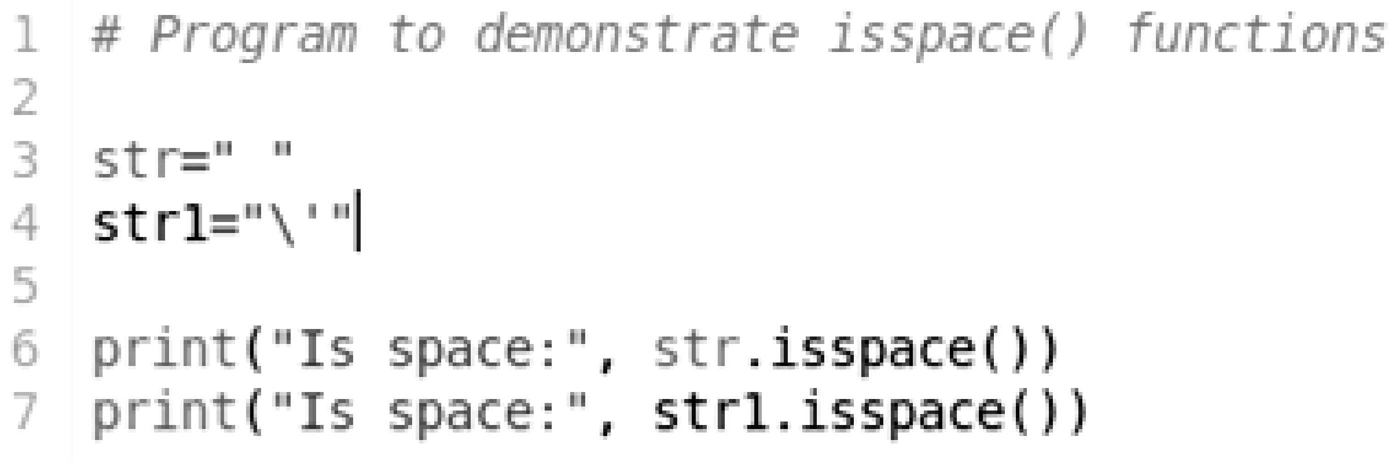


```
Is alpha: True
Is alpha: False
```

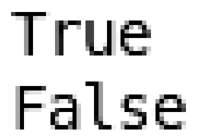
String Methods and Functions :

isspace() :

- **Returns True if all characters in a string are spaces.**



```python
# Program to demonstrate isspace() functions

str=" "
str1="\'"

print("Is space:", str.isspace())
print("Is space:", str1.isspace())
```

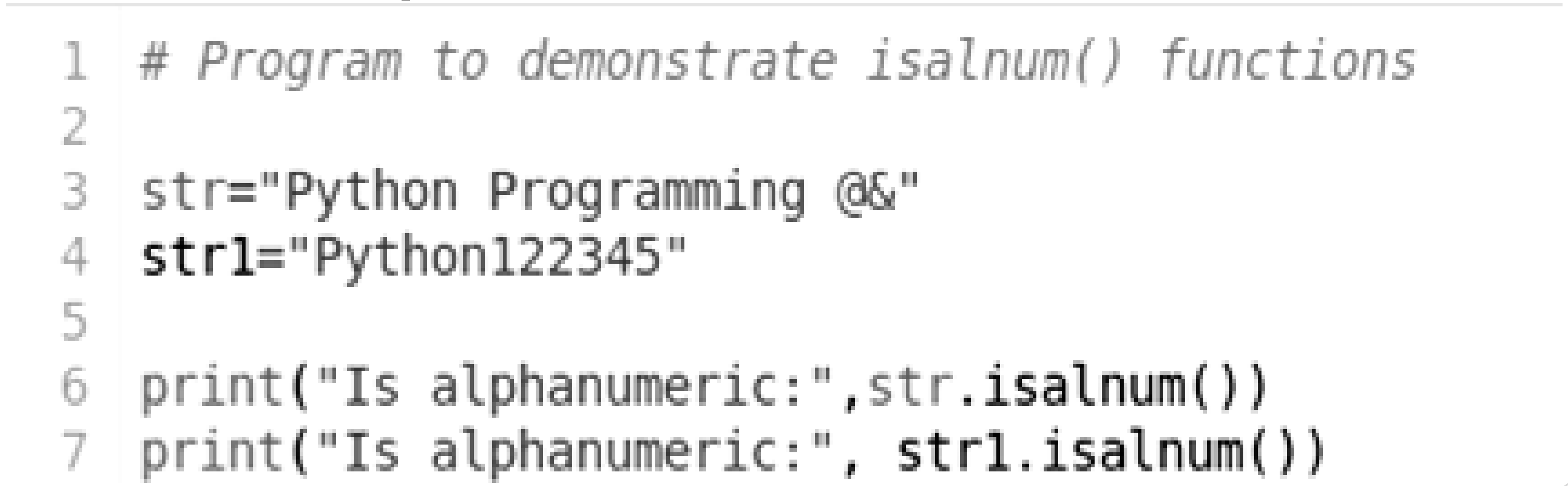


```
True
False
```

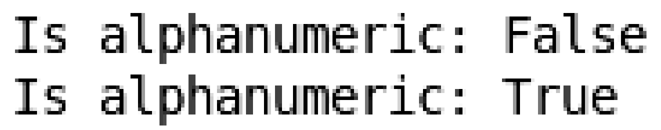
String Methods and Functions :

isalnum() :

- **The method isalnum() is used to determine whether the Python string consists of alphanumeric characters.**



```python
# Program to demonstrate isalnum() functions

str="Python Programming @&"
str1="Python122345"

print("Is alphanumeric:",str.isalnum())
print("Is alphanumeric:", str1.isalnum())
```

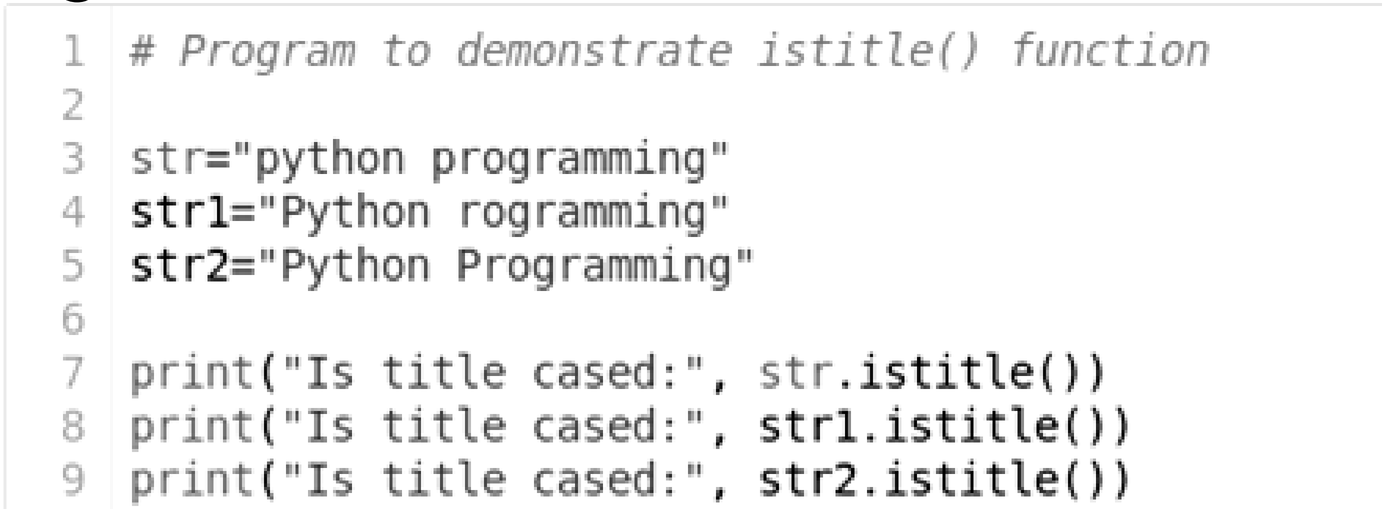


```
Is alphanumeric: False
Is alphanumeric: True
```

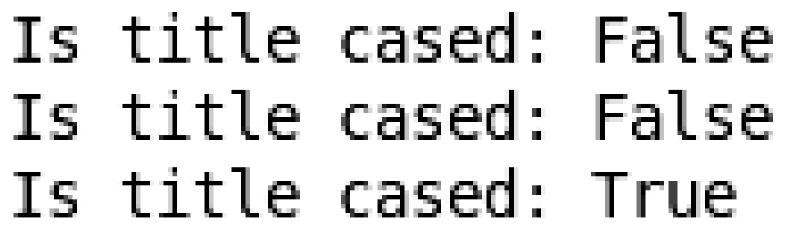
String Methods and Functions :

istitle() :

- **The method istitle() return true if the string is a title cased.**



```
# Program to demonstrate istitle() function

str="python programming"
str1="Python rogramming"
str2="Python Programming"

print("Is title cased:", str.istitle())
print("Is title cased:", str1.istitle())
print("Is title cased:", str2.istitle())
```

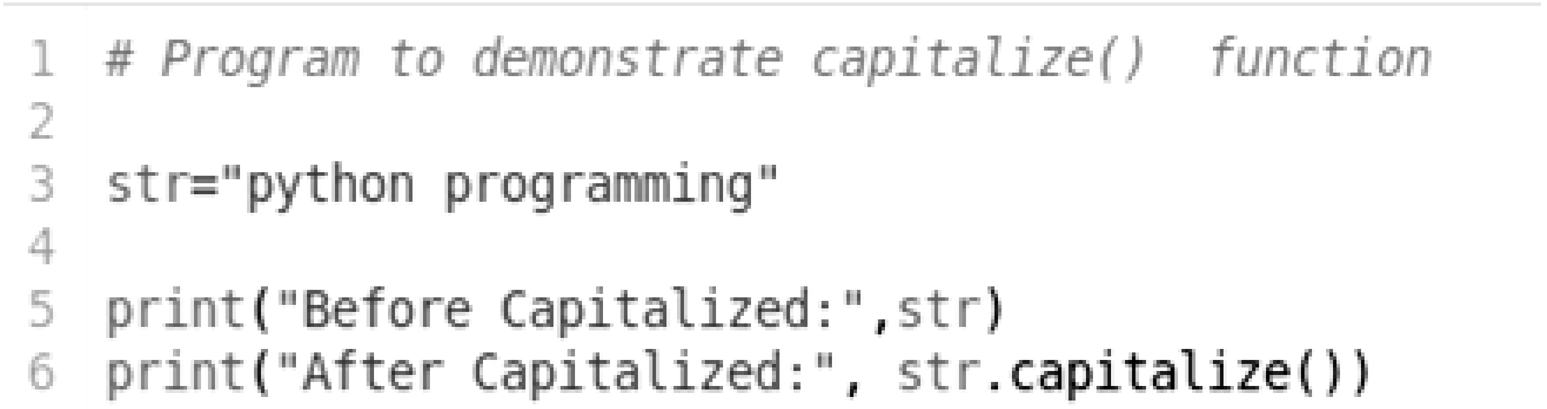


```
Is title cased: False
Is title cased: False
Is title cased: True
```

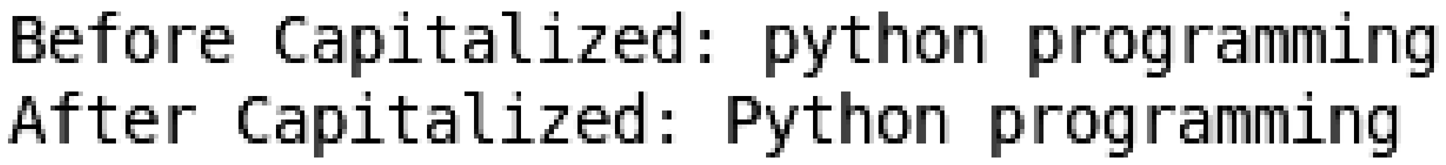
String Methods and Functions :

capitalize() :

- **This function capitalizes first letter of string.**



```
# Program to demonstrate capitalize()  function

str="python programming"

print("Before Capitalized:",str)
print("After Capitalized:", str.capitalize())
```

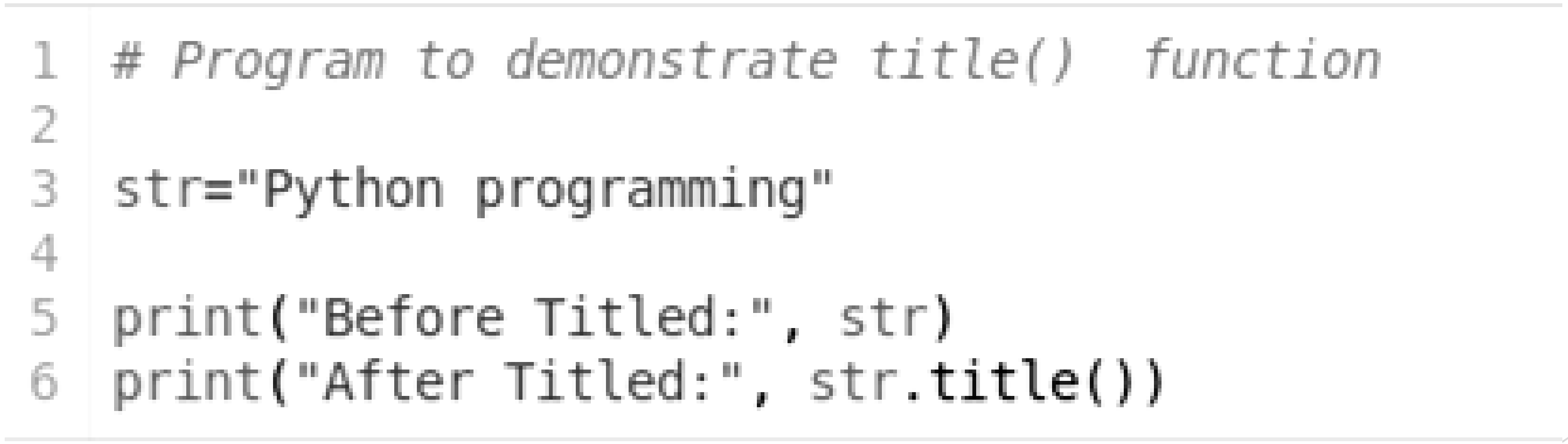


```
Before Capitalized: python programming
After Capitalized: Python programming
```

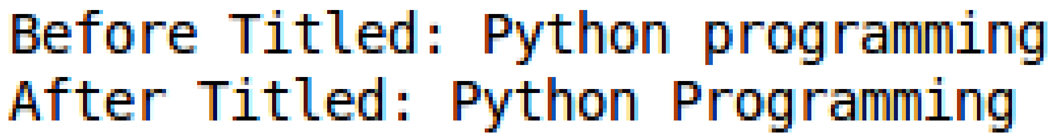
String Methods and Functions :

title() :

- **The method title() returns a copy of the string in which first character of all words of string are capitalised.**



```
# Program to demonstrate title()  function

str="Python programming"

print("Before Titled:", str)
print("After Titled:", str.title())
```

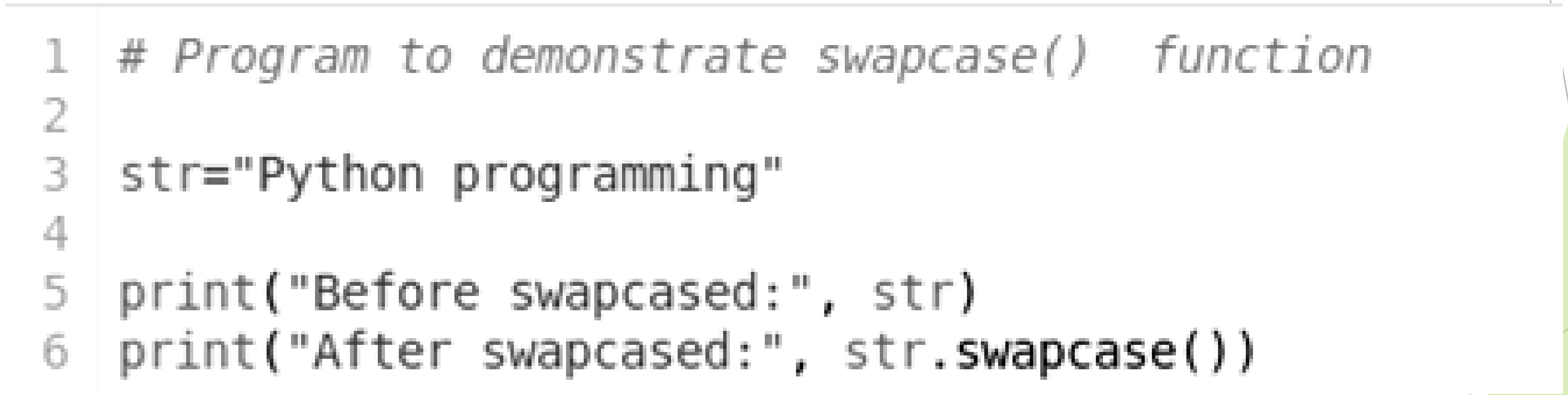


```
Before Titled: Python programming
After Titled: Python Programming
```

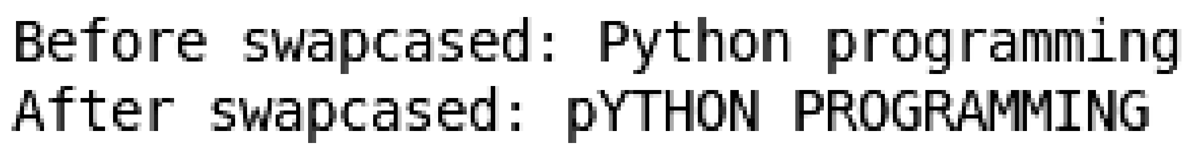
String Methods and Functions :

swapcase() :

- **The method swapcase() returns a copy of the string in which all case based character swap their case.**



```python
# Program to demonstrate swapcase()  function

str="Python programming"

print("Before swapcased:", str)
print("After swapcased:", str.swapcase())
```

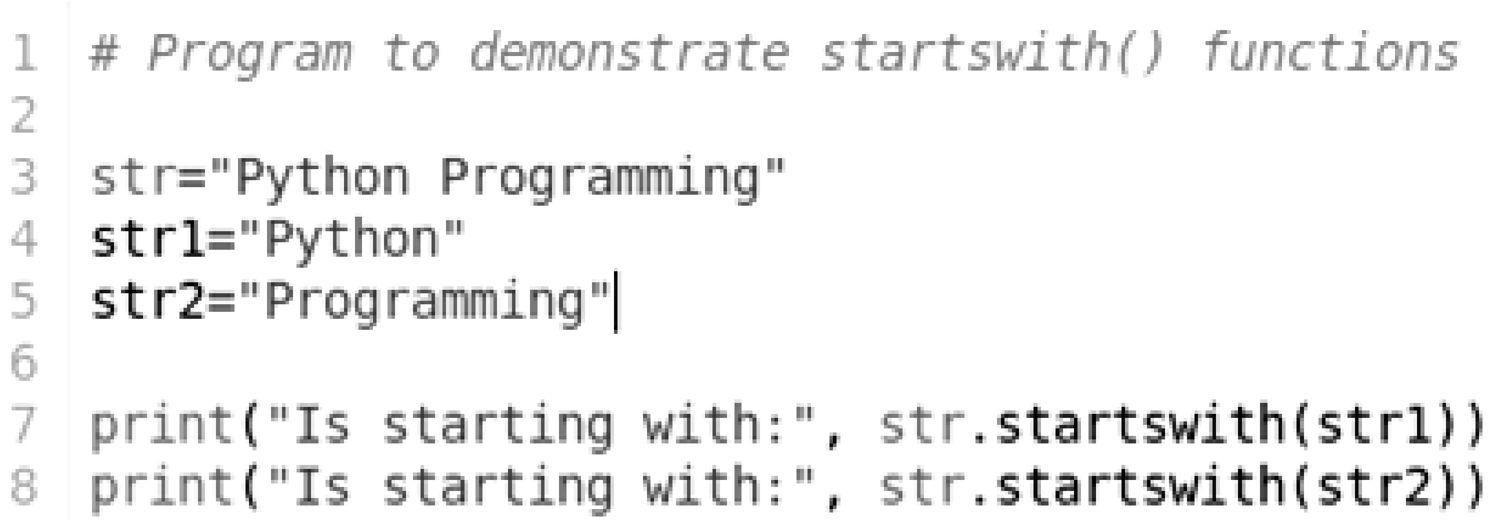


```
Before swapcased: Python programming
After swapcased: pYTHON PROGRAMMING
```

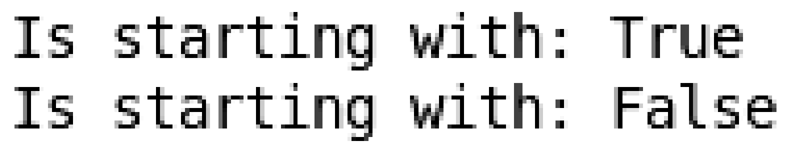
String Methods and Functions :

startswith() :

- **It takes a string as an argument, and returns True is the string it is applied on begins with the string in the argument.**



```python
# Program to demonstrate startswith() functions

str="Python Programming"
str1="Python"
str2="Programming"

print("Is starting with:", str.startswith(str1))
print("Is starting with:", str.startswith(str2))
```



```
Is starting with: True
Is starting with: False
```