www.thegordon.edu.au

Python Programming

Fine-tuning the print() function output

UEE60411- Advanced Diploma in Computer Systems Engineering.

UEENEED111A - Develop, Implement and Test Object Oriented Code

For further information:

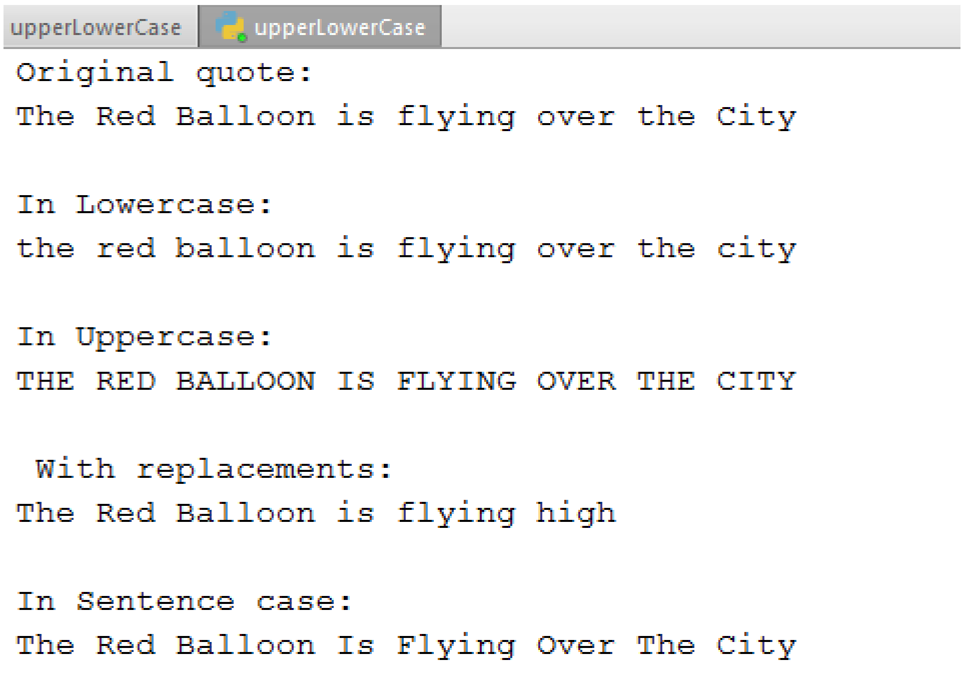
Leanne Matheson
ICT – Creative Business Enterprise
Ph: (03) 5225 0616
Email: lmatheson1@gordontafe.edu.au

Fine-tuning the print() function output.

You can convert collected string data to upper, lower or sentence case. This is also handy if you wish to react to the same word typed in with different cases from the keyboard. Try out the example below.

```
# Quotation manipulation, dealing with upper and lower case input
#Original Quote

quote = “ The Red Balloon is flying over the City”

print(“Original quote: “)
print(quote)

print(“\nIn uppercase: “)
print(quote.upper())

print(“\nIn lowercase: “)
print(quote.lowercase())

print(“\nAs a title: “)
print(quote.title())

print(“\nWith a little replacement”)
print(quote.replace(“over the city”,”high”)

print(“\nThe original quote is unchanged”)
print(quote)

input(“\n\nPress the enter key to exit.”)
```

The output should look like below:



```
Original quote:
The Red Balloon is flying over the City

In Lowercase:
the red balloon is flying over the city

In Uppercase:
THE RED BALLOON IS FLYING OVER THE CITY

 With replacements:
The Red Balloon is flying high

In Sentence case:
The Red Balloon Is Flying Over The City
```

You will notice that the code above has called up some built-in methods that python can recognize.

These are upper(), lower(), sentence() and replace(). These methods cannot be called up on their own, but they can be called up in conjunction with the print function.

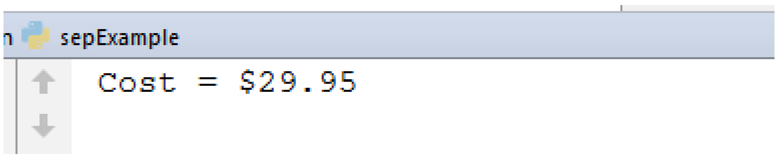
We can also read user input as lower-case with the use of the lower() function. You will see how handy this can be later on- in future topics.

Getting rid of the spaces generated by the comma.

If you have a float variable called cost, and you wish to print it out with a $ sign, you will not necessarily want a gap between the $ and the amount.

```
cost = 25.95
print(“Cost = $”, cost, sep=””)
```

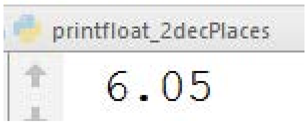
The result should show up without a gap.



```
Cost = $29.95
```

Printing a float so that it shows two(2) decimal places.

```
num = 6.04567
print( '%.2f' %num)
```



```
6.05
```