CHAPTER 6

EXERCISES

[1](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1616). What are escape characters?

* An escape character lets you use characters that are otherwise impossible to put into a string. An escape character consists of a backslash (\) followed by the character you want to add to the string.
* Eg: print(‘John’s father’) - this line of code throws an error because python thinks the total string as ‘John’.
* For avoiding this we use escape character. print(‘John\’s father’)

[2](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1617). What do the \n and \t escape characters represent?

* \n = New line
* \t = Tab space

[3](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1618). How can you put a \ backslash character in a string?

* \\ = backslash
* We can use **raw strings** that tells the python compiler to ignore all the special characters.

[4](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1619). The string value "Howl's Moving Castle" is a valid string. Why isn’t it a problem that the single quote character in the word Howl's isn’t escaped?

* Single quotes can be used in a string without an espace character by putting the entire string into double quotes and vice versa.

[5](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1620). If you don’t want to put \n in your string, how can you write a string with newlines in it?

* Multiline strings can be used.
* A Multiline string in Python begins and ends with either three single quotes or three double quotes

[6](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1621). What do the following expressions evaluate to?

* 'Hello, world!'[1]

Ans: ‘e’

* 'Hello, world!'[0:5]

Ans: ‘Hello’

* 'Hello, world!'[:5]

Ans: ‘Hello’

* 'Hello, world!'[3:]

Ans: 'lo, world!'

[7](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1622). What do the following expressions evaluate to?

* 'Hello'.upper()

Ans: ‘HELLO’

* 'Hello'.upper().isupper()

Ans: True

* 'Hello'.upper().lower()

Ans: False

[8](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1623). What do the following expressions evaluate to?

* 'Remember, remember, the fifth of November.'.split()

Ans: ['Remember,', 'remember,', 'the', 'fifth', 'of', 'November.']

* '-'.join('There can be only one.'.split())

Ans: There-can-be-only-one.

[9](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1624). What string methods can you use to right-justify, left-justify, and center a string?

* rjust(), ljust(), and center() Methods

[10](https://automatetheboringstuff.com/2e/chapter6/#calibre_link-1625). How can you trim whitespace characters from the beginning or end of a string?

* lstrip()
* rstrip()

#### Table Printer

Write a function named printTable() that takes a list of lists of strings and displays it in a well-organized table with each column right-justified. Assume that all the inner lists will contain the same number of strings. For example, the value could look like this:

tableData = [['apples', 'oranges', 'cherries', 'banana'],  
             ['Alice', 'Bob', 'Carol', 'David'],  
             ['dogs', 'cats', 'moose', 'goose']]

Your printTable() function would print the following:

   apples Alice  dogs  
  oranges   Bob  cats  
 cherries Carol moose  
   banana David goose

Hint: your code will first have to find the longest string in each of the inner lists so that the whole column can be wide enough to fit all the strings. You can store the maximum width of each column as a list of integers. The printTable() function can begin with colWidths = [0] \* len(tableData), which will create a list containing the same number of 0 values as the number of inner lists in tableData. That way, colWidths[0] can store the width of the longest string in tableData[0], colWidths[1] can store the width of the longest string in tableData[1], and so on. You can then find the largest value in the colWidths list to find out what integer width to pass to the rjust() string method.

Code:

tableData = [['apples', 'oranges', 'cherries', 'banana'],

['Alice', 'Bob', 'Carol', 'David'],

['dogs', 'cats', 'moose', 'goose']]

def printTable(table):

colWidths = [0] \* len(table)

# search for the longest string in each list of tableData

# and put the numbers of characters in the new list

for y in range(len(table)):

for x in table[y]:

if colWidths[y] < len(x):

colWidths[y] = len(x)

#print(colWidths)

#print table row by row

for x in range(len(table[0])) :

for y in range(len(table)) :

print(table[y][x].rjust(colWidths[y]), end = ' ')

print()

printTable(tableData)

Output:

Graphical user interface, text

Description automatically generated with medium confidence