IS1110 Tutorial 9 – Advanced Strings & String Formatting

Overview

This tutorial builds your understanding of strings & string formation, also introducing more advanced string exercises.

Strings

Exercise 1 – Movie Night Sentence (warm up exercise)

Goal:

Practise joining (concatenating) several strings into one message using user input.

Task:

Ask the user for their name, favourite movie, and favourite snack. Then print:

Expected output:

Hi <name>! Tonight you are watching <movie> with some <snack>.

Exercise 2 – Counting the Length of a String

Goal:

Practise using a for loop to count how many characters are in a string.

(Notice that spaces, punctuation, and numbers are all counted as characters.)

Task:

1. Define a function called string_length that takes one argument, str1.
2. Inside the function, use a loop to count how many characters are in str1.
3. The function should return the final count.
4. Call the function with the following strings and print the results:

"Hello"

"Hello World!"

"Python 4 everyone"

Expected Output:

5

12

17

Exercise 3 – Every Second Character from a String

Goal:

Practise using range() and len() to loop over a string by index and build a new string.

Task:

1. Define a function called odd_values_string that takes one argument, e.g. str3.
2. The function should create a new string made up of every second character, starting with the first character
3. Return this new string.
4. Test the function with:

"abcdef"

"python"

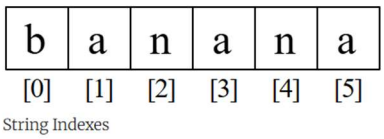
Expected Output:

ace

pto

Exercise 4 – Adding “ing” or “ly” to a Word

So `b` is the 0th letter ("zero-eth") of "banana", `a` is the 1th letter ("one-eth"), and `n` is the 2th ("two-eth") letter.



String Indexes

Goal:

Practise using string length and slicing to modify the end of a word.

Task:

1. Define a function called add_string that takes one argument, str2.
2. If the length of str2 is 2 characters or less, return the string unchanged.
3. If the length is greater than 2:
    a. If the string already ends with "ing", add "ly" to the end.
    b. Otherwise, add "ing" to the end.
4. Print the result of calling the function with:

"ab"

"abc"

"string"

Expected Output:

ab

abcing

stringly

Exercise 5 – Looping Through a String with for and while

Goal:

Show how to loop through each character in a string using both a for loop and a while loop.

Task:

1. Ask the user to enter a word and store it in a variable.
2. Use a for loop to print each character of the word on its own line, with a heading:
3. Using for loop:
4. Then, use a while loop to do the same thing. Start an index at 0.
5. Keep looping while the index is less than the length of the string.
6. Print the character at the current index and then increase the index by 1.
7. Expected Output:
8. If the user enters:

   cat

Then the output should look like:

Using for loop:	Using while loop:
c	c
a	a
t	t

String Formatting Exercises

Exercise 6 – Escape Sequences \t

Goal: Show \t (tab) inside strings.

Task:

Write a program that prints a tiny timetable like this (2 lines, with tabs):

Expected Output:

Day Activity

Mon Gym

Exercise 7 – Custom separator and line ending

Goal: Show how print() can change the separator and line ending.

Task:

Ask the user for three favourite foods.

Print them on one line, separated by " | " (pipe and spaces), and make sure the line ends with " THE END" instead of a normal newline.

Expected Output:

Pizza | Pasta | Sushi THE END

Exercise 8 – f-strings with a Calculation (fill in the blanks)

The following code asks the user for a product name and its price (as a number). Then a tax of 23% is added to the price. Print statement uses f-string and ensures that the value minus tax is returned to 2 decimal places.

```
product = _________("Enter the product name: ")

price = _______(input("Enter the price (before tax): "))


price_with_tax = ______  * 1.23


print(f"The price of {_____} with tax is €{_______:.2f}.")
```

Expected Output:

The price of <product> with tax is €<price_with_tax>.

Final submission for portfolio project on 23rd November

Submission 8 – optional advanced submission

Submission 9 – integration - putting it all together