MINI EXERCISES

Chapter 1 mini exercise

1. Write a small program that asks for four lines of an address. Afterwards print that address.
2. Write a small program that asks for 5 numbers. You must then print the sum and average of those numbers.
3. Print a hollow square shape using ‘\*’ in print statements.

Chapter 2 mini exercise

1. Write a small program that takes two numbers. Using a loop of your choice, print every number between those two numbers.
2. Write a small program that takes in a number. The program must print a triangle which has the height of the input number e.g input = 3 output=

\*

\*\*

\*\*\*

1. Write a shopping list calculator. The user can enter a price. If they wish to end the list they must type ‘stop’. Once the list has been complete, you should print the total.

Chapter 3 mini exercise

1. Write a small program that asks for 10 numbers and puts them into a list. After the list is complete you must print the highest number in the list.
2. Write a small program that takes in a list of 10 numbers. Produce two new lists for odd and even numbers and print them. (hint: look at a modulo operator)
3. Returning to the shopping list problem from the previous chapter, store all the item names in a list and then print the full list once the user has finished.

EXTENSIONS

Chapter 4 Extension

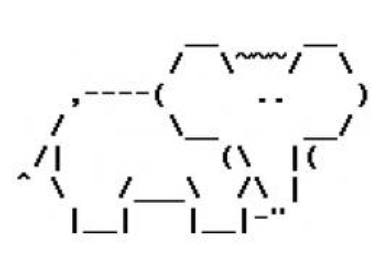
You may notice that checking for the user input is not extremely robust as it does not take into account if the user enters ‘’waRRior”. However there are ways to overcome this by having more rigorous checks. Can you think of a way of implementing this? (If you are struggling, a hint is provided in the code for this chapter on the Github link).

Chapter 5 Extension

Currently all items of the loot have an equal chance of being produced but as the game designer you may want to make it harder to get certain items. This is called adding weighting to items and adds more depth to the game. Can you think of how to implement it? (If you are struggling, a hint is provided in the code for this chapter on the Github link. Alternatively, you can return to this chapter later on as future chapters will teach you how to do this.)

Chapter 6 Extension

Remember at the beginning I mentioned ASCII art. This is artwork that is done using characters that can be typed on a computer but can be difficult to do. However, if done well, the game becomes more immersive and enjoyable for the player as you are giving them a visual cue. Here is an example of what ASCII art looks like:



This picture has been made using only characters that can be typed in the keyboard.

You can also make ASCII art for each role your game has and if you need some help, there are many ASCII art tutorials.

Chapter 7 Extension

Have you considered having spells in your game? These spells would use a separate resource such as mana or energy. Different classes can have different spells such as a Mage class that can heal themselves which is perhaps stronger than a potion. Another alternative is a fireball like spell which adds a damage over time debuff. The possibilities are endless but can be tricky to implement.

Also if you have the time, you can include ASCII art for these spells too!

Chapter 8 Extension

Expand your game

Remember to go back and have a go at some of the extension tasks. They really help develop the game further so that it looks more polished with further features. Don’t forget, this is your game so if you have an idea you should try to add it into the game. If you are not sure how to add extra features, you can try find other tutorials in Python which may help you accomplish your goal!