**HOMEWORK WEEK 3**

(handout for students)

**TASK 1 (Conditional flow)**

**Question 1**

Create a program that tells you whether or not you need an umbrella when you leave the house.

**The program should:**

1. Ask you if it is raining using input()

2. If the input is 'y', it should output 'Take an umbrella'

3. If the input is 'n', it should output 'You don't need an umbrella'

Answer:

weather = input("Is it raining outside? Y/N ")

if weather == "Y":

    print("You need an umbrella")

elif weather == "N":

    print("You can leave your umbrella at home")

else:

    print("Invalid input")

**Question 2**

I'm on holiday and want to hire a boat. The boat hire costs £20 + a refundable £5 deposit. I've written a program to check that I can afford the cost, but something doesn't seem right. Have a look at my program and work out what I've done wrong

my\_money = input('How much money do you have? ')

boat cost = 20 + 5

if my\_money < boat\_cost:

print('You can afford the boat hire')

else :

print('You cannot afford the board hire'

Answer:

my\_money = input('How much money do you have? ')

boat\_cost = 20 + 5

if int(my\_money) >= boat\_cost:

    print('You can afford the boat hire')

else :

    print('You cannot afford the board hire')

**Question 3**

Your friend works for an antique book shop that sells books between **1800** and **1950** and wants to quickly categorise books by the century and decade that they were written. Write a program that takes a year (e.g. 1872) and outputs the century and decade (e.g. "Eighteenth Century, Seventies")

Answer:

book\_year = int(input("What year was the book written? "))

if 1800 < book\_year < 1899:

    if 1800 < book\_year <= 1809:

        print("Noughties")

    elif 1810 < book\_year <= 1819:

        print("Tens")

    elif 1820 < book\_year <= 1829:

        print("Twenties")

    elif 1830 < book\_year <= 1839:

        print("Thirties")

    elif 1840 < book\_year <= 1849:

        print("Fourties")

    elif 1850 < book\_year <= 1859:

        print(Fifties)

    elif 1860 < book\_year <= 1869:

        print("Sixties")

    elif 1870 < book\_year <= 1879:

        print("Seventies")

    elif 1880 < book\_year <= 1889:

        print("Eighties")

    elif 1890 < book\_year <= 1899:

        print("Nineties")

    else:

        print("Invalid year")

    print("19th Century")

elif 1900 < book\_year <= 1950:

    if 1900 < book\_year <= 1909:

        print("Noughties")

    elif 1910 < book\_year <= 1919:

        print("Tens")

    elif 1920 < book\_year <= 1929:

        print("Twenties")

    elif 1930 < book\_year <= 1939:

        print("Thirties")

    elif 1940 < book\_year <= 1949:

        print("Fourties")

    else:

        print("Invalid year")

    print('20th Century')

else:

    print("Invalid year")

**TASK 2 (Lists and Dictionaries)**

**Question 1**

I have a list of things I need to buy from my supermarket of choice.

shopping\_list = [

"oranges",

"cat food",

"sponge cake",

"long-grain rice",

"cheese board",

]

print(shopping\_list[1])

I want to know what the first thing I need to buy is. However, when I run the program it shows me a different answer to what I was expecting? What is the mistake? How do I fix it.

Answer: To access the first item in the list, it would be a zero instead of a 1, as Python is zero indexed.

**Question 2**

I'm setting up my own market stall to sell chocolates. I need a basic till to check the prices of different chocolates that I sell. I've started the program and included the chocolates and their prices. Finish the program by **asking the user to input an item and then output its price**.

chocolates = {

'white': 1.50,

'milk': 1.20,

'dark': 1.80,

'vegan': 2.00,

}

Answer:

chocolates = {

    'white': 1.50,

    'milk': 1.20,

    'dark': 1.80,

    'vegan': 2.00,

}

item = input("Select a chocolate: white, milk, dark or vegan. ")

print(f'Price: {chocolates[item]}')

**Question 3**

Write a program that simulates a lottery. The program should have a list of seven numbers that represent a lottery ticket. It should then generate seven random numbers. After comparing the two sets of numbers, the program should output a prize based on the number of matches:

· £20 for three matching numbers

· £40 for four matching numbers

· £100 for five matching numbers

· £10000 for six matching numbers

· £1000000 for seven matching numbers

Answer:

import random

lottery\_numbers = [5, 9, 22, 25, 37, 40, 49]

your\_numbers = []

for i in range(7):

    numbers = random.randint(1, 49)

    your\_numbers.append(numbers)

print(your\_numbers)

matches = []

for num in your\_numbers:

    if num in lottery\_numbers:

        matches =+ 1

if matches == 3:

    print("You have won £20!")

elif matches == 4:

    print("You have won £40!")

elif matches == 5:

    print("You have won £100!")

elif matches == 6:

    print("You have won £10000!")

elif matches == 7:

    print("You have won the jackpot of £1000000!")

else:

    print("You didn't win anything, better luck next time!")

**TASK 3 (Read and Write files)**

**Question 1**

**You're having coffee/tea/beverage of your choice with a friend that is learning to program in Python. They're curious about why they would use pip. Explain what pip is and one benefit of using pip.**

PIP is one of many package managers that have the purpose of installing you’re your environment libraries of code that have already been published. Having PIP makes the management, installation and application of modules, libraries and packages much easier for the user.

**Question 2**

**This program should save my data to a file, but it doesn't work when I run it. What is the problem and how do I fix it?**

**poem = 'I like Python and I am not very good at poems'**

**with open('poem.txt', 'r') as poem\_file:**

**poem\_file.write(poem)**

Answer:

The file needs to be opened in write, append or update mode in order to add text into it. The current mode selected is ‘r’ for read, and will only allow it to be read.

**Question 3**

Here is a snippet of Elton John’s song “I’m Still Standing”

**You could never know what it's like**

**Your blood like winter freezes just like ice**

**And there's a cold lonely light that shines from you**

**You'll wind up like the wreck you hide behind that mask you use**

**And did you think this fool could never win?**

**Well look at me, I'm coming back again**

**I got a taste of love in a simple way**

**And if you need to know while I'm still standing, you just fade away**

**Don't you know I'm still standing better than I ever did**

**Looking like a true survivor, feeling like a little kid**

**I'm still standing after all this time**

**Picking up the pieces of my life without you on my mind**

**I'm still standing (Yeah, yeah, yeah)**

**I'm still standing (Yeah, yeah, yeah)**

**Tasks:**

1. Write the lyrics to a new file called song.txt

2. Check that a file has been created successfully.

3. The read lines from this file and print out ONLY those lines that have a word ‘still’ in them.

with open('song.txt', 'r+') as still\_standing:

    lyrics = still\_standing.readlines()

still\_in\_lyrics = []

for line in lyrics:

    if 'still' in line:

        still\_in\_lyrics.append(line)

print(still\_in\_lyrics)

**TASK 4 (API)**

**Question 1**

In this session you used the Pokémon API to retrieve a single Pokémon.

I want a program that can retrieve multiple Pokémon and save their names and moves to a file.

Use a list to store about 6 Pokémon IDs. Then in a for loop call the API to retrieve the data for each Pokémon. Save their names and moves into a file called 'pokemon.txt'

**Question 2 (optional)**

Here is a link to a really cool API:<https://opentdb.com/>

Answer the following questions:

· What is the name of this API?

· What does it do?

· Example URL to make a call to this API?

· Example output?