Learning Activity Week 4

Code Analysis

For each of the following given codes:

- Without executing the code try to read the code and write down what will be the output.
- Use the Python Code Visualizer and execute the code step-by-step. Observe how the variables and statements are executing in each iteration of the loops.

```
# Code 1
i = 7
for number in range(1, i + i):
 print(number)

# Code 2
i = 1
j = 10
for number in range(i, j):
   if number > 5:
       print(number)
   else:
       print('Hello')

# Code 3
sentence = "I just came to say hello!"
count = 0
for letter in sentence:
   if letter == " ":
       count = count + 1
   elif letter == "a":
       count = count - 1
print(count)

# Code 4
sentence = "I just came to say hello!"
for i in range(0, len(sentence)):
      print(sentence[i])

# Code 5
sentence = "I just came to say hello!"
for c in sentence:
      print(c)

Expectations:
- Code 1 will print the numbers 1 to 13
- Code 2 will print the numbers 1 to 5 and print “Hello” 5 other times
- Code 3 will calculate how many spaces there are and how many “a” there are and
basically do “space” minus “a”
- Code 4 will print the sentence vertically
- Code 5 will do the same as code 4
```

Supporting Topics

Data Formats

Task

1. Perform a free (re-)search and explore the answers for the following questions:

    - Digits in decimal numbers are 0-9. What are the digits in hexadecimal format? What are the digits in binary format?
      Answer:

Hexadecimal numbers (0, 1, 2, 3 ,4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
Binary: (0, 1)

- Convert (manually) the following decimal numbers to hexadecimal and binary: 8, 10, 15, 21, 32, 64, 256, 500, 512, 1000.
  (**to convert decimal to hexadecimal:** divide the number by 16, grab the number after the comma and multiply by 16, that's the remainder, when reached 0 with remainder one, stop. Convert the remainders to hexadecimal and write the number down from most significant number to least significant number)

  Decimal – Hexadecimal – Binary

  8 – 0x8 - 0b1000

  15 – 0xF - 0b1111

  21 – 0x15 - 0b10101

  32 – 0x20 - 0b100000

  64 – 0x40 - 0b1000000

  256 – 0x100 - 0b10000000

  500 – 0x1F4 - 0b111110100

  512 – 0x200 – 0b1000000000

  1000 – 0x3E8 - 0b1111101000

- How does Python represent these data formats? How can you use Python to convert these data formats to each other?

  Python uses decimal as "0-10", hexadecimal as "0x…" and binary as "0b…."
  Python uses functions such as int(), hex() and bin()

2. Use Python to:
    - Convert the decimal number 45 into its binary representation.
      ```
      bin(45)
      ```
    - Convert the binary number 1010101 into decimal form.
      ```
      int(0b1010101)
      ```
    - Add the binary numbers 10111 and 1101 and express the result in binary.
      ```
      bin(0b10111 + 0b1101)
      ```
    - Convert the decimal number 255 into its hexadecimal representation.
      ```
      hex(255)
      ```
    - Convert the hexadecimal number 2A into decimal form.
      ```
      int(0x2A)
      ```
    - Add the hexadecimal numbers C4 and 3A and express the result in hexadecimal.
      ```
      hex(0xC4 + 0x3A)
      ```
    - Convert the binary number 1101 into decimal form.
      ```
      int(0b1101)
      ```
    - Convert the hexadecimal number F0 into decimal form.
      ```
      int(0xF0)
      ```
    - Add the decimal numbers 123 and 456
      ```
      123 + 456
      ```
    - Convert the decimal number 157 into binary and then into hexadecimal.
      ```
      hex(bin(157))
      ```
    - Convert the binary number 11101101 into decimal and then into hexadecimal.
      ```
      hex(int(0b11101101))
      ```
    - Convert the hexadecimal number AB4 into decimal and then into binary.
      ```
      bin(int(0xAB4))
      ```
3. Real-life Applications:
    - Research and identify a real-world example where binary data is used extensively.
      All computers use binary as it’s main data format
    - Investigate how hexadecimal is used in computer memory addressing.
      Computer memory addressing uses hexadecimal being it a smaller way to write binary, therefore faster.
    - Explore how decimal data formats are used in financial calculations or accounting systems.
      Decimal data formats are very precise and it makes it easier to show fractional amounts and to change currencies. This is important for example for taxes.