Document name	:	HR.Skills.A1W4L3
Document description	:	This document contains the supporting topic: "Data formats"
Document version	:	V 1.0
Written by	:	Danny de Snoo
Date(s) written	:	07-06-2024

Supporting topic description:

Introduction:

Data formats are essential in the world of computing, providing a standardized way to represent and manipulate information. One of the most fundamental data formats is binary, which uses a base-2 numbering system composed of only two digits: 0 and 1. Binary is the foundation of all digital systems, with each digit representing the state of an electronic switch. Hexadecimal (hex) is another commonly used data format, employing a base-16 numbering system. It extends beyond the limited set of digits in decimal, introducing the additional symbols A-F to 0-9. Hexadecimal is frequently used to represent and display binary data in a more compact and human-readable format. On the other hand, decimal is a base-10 numbering system that most people are familiar with, utilizing ten digits 0-9. Decimal is widely used in everyday life and is the standard format for representing numerical values. Understanding these data formats is crucial for effectively working with data in various computing applications and systems.

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?
   - Convert (manually) the following decimal numbers to hexadecimal and binary: 8, 10, 15, 21, 32, 64, 256, 500, 512, 1000.
   - How does Python represent these data formats? How can you use Python to convert these data formats to each other?

2. Use Python to:
   - Convert the decimal number 45 into its binary representation.
   - Convert the binary number 1010101 into decimal form.
   - Add the binary numbers 10111 and 1101 and express the result in binary.
   - Convert the decimal number 255 into its hexadecimal representation.
   - Convert the hexadecimal number 2A into decimal form.
   - Add the hexadecimal numbers C4 and 3A and express the result in hexadecimal.
   - Convert the binary number 1101 into decimal form.
   - Convert the hexadecimal number F0 into decimal form.
   - Add the decimal numbers 123 and 456.
   - Convert the decimal number 157 into binary and then into hexadecimal.
   - Convert the binary number 11101101 into decimal and then into hexadecimal.
   - Convert the hexadecimal number AB4 into decimal and then into binary.

3. Real-life Applications:
   - Research and identify a real-world example where binary data is used extensively.
   - Investigate how hexadecimal is used in computer memory addressing.
   - Explore how decimal data formats are used in financial calculations or accounting systems.

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?

Hexadecimal:

Hexadecimal is numbering system with base16.
There are 16 symbols or digit value combinations possible

Digits: 1 to 9
Symbols: A, B, C, D, E and F

Example:

decimal		Hexadecimal
8	=	8
1000	=	3E8

Binary:

Binary is the most basic of numbering formats for a computer to understand.
This numbering system is represented by 1's and 0's

example:

decimal		binary
8	=	1000
1000	=	1111101000

   - Convert (manually) the following decimal numbers to hexadecimal and binary: 8, 10, 15, 21, 32, 64, 256, 500, 512, 1000.

Decimal	***Hexadecimal***	***Binary***
8	8	1000
10	A	1010
15	F	1111
21	15	10101
32	20	100000
64	40	1000000
256	100	100000000
500	1F4	111110100
512	200	1000000000
1000	3E8	1111101000

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

Python has a built-in function for turning decimal values into its binary form.
This function is bin(integer). You can turn a binary into an integer by using int(binary, 2) to convert it back into its decimal number. 2 signals to the function that it is converting a Binary into a integer

2. Use Python to:

- Convert the decimal number 45 into its binary representation.

  0101101

- Convert the binary number 1010101 into decimal form.

  85

- Add the binary numbers 10111 and 1101 and express the result in binary.

  381

- Convert the decimal number 255 into its hexadecimal representation.

  0xff

- Convert the hexadecimal number 2A into decimal form.

  42

- Add the hexadecimal numbers C4 and 3A and express the result in hexadecimal.

  FE

- Convert the binary number 1101 into decimal form.

  170

- Convert the hexadecimal number F0 into decimal form.

  240

- Add the decimal numbers 123 and 456.

  579

- Convert the decimal number 157 into binary and then into hexadecimal.

  010011101, 0x9d

- Convert the binary number 11101101 into decimal and then into hexadecimal.

  237, E0xed

- Convert the hexadecimal number AB4 into decimal and then into binary.

  2740, 0101010110100

3. Real-life Applications:
   - Research and identify a real-world example where binary data is used extensively.

     Binary is always used by a computer do to the architecture of our modern CPU's and GPU transistor based architecture.
     A transistor is basically ON or OFF and Binary shows this by its 1 and O.
     This means that for a modern computer, Binary is its primary language for communication.

   - Investigate how hexadecimal is used in computer memory addressing.

     Hexadecimal is used to represent computer memory addressing in simpler way than with a Binary based system.
     Hexadecimal helps with the readability of the memory for humans (because binary for very big number can get quite long very quickly)

     for instance:

Decimal	:	500
Binary	:	111110100
Hexadecimal	:	1F4

     In this way, its becomes much more readable for a normal human/developer to see which value is stored the memory and it's also easier for the computer to understand then a decimal number.

   - Explore how decimal data formats are used in financial calculations or accounting systems.

     Most financial calculations require the use of decimal numbers because decimal numbers are used to represent money in the real world.

     Because financial data concerning money need to always be precise the best readable format is always the decimal system.

     Doing calculation in for instance Binary or Hexadecimal would not work because this would not be readable for normal bookkeepers or accountants.

     (Have tried this with a fellow student (Hamdi Versloot (1037622)) who studies accountancy at Hogeschool Rotterdam and he couldn't read it)