Template Week 1 – Bits & Bytes

Student number: 569527

Assignment 1.1: Bits & Bytes intro

What are Bits & Bytes?

A **bit** is the smallest unit of data, either a **0** or **1**. A **byte** is a group of **8 bits**, used to represent things like letters, numbers, or small pieces of information. Computers use bits and bytes to store and process everything digitally.

What is a nibble?

A nibble is a fundamental unit of data in computing and digital technology. It represents four consecutive binary digits (bits), which is equal to half of an 8-bit byte. Since a byte contains 8 bits, a nibble can be thought of as a smaller, simplified division of data used in specific applications where small units are more practical.

What relationship does a nibble have with a hexadecimal value?

A **nibble** (4 bits) corresponds directly to a **single hexadecimal digit**. Each hexadecimal digit can represent one nibble, allowing a compact representation of binary numbers.

Why is it wise to display binary data as hexadecimal values?

Displaying binary data as hexadecimal is easier because **one hex digit represents exactly 4 bits**, making it straightforward to convert between binary and hexadecimal. This 4-bit grouping is also called a **nibble**, a playful term for half a byte. The structure allows for quick mental conversion and simplifies working with binary data

What kind of relationship does a byte have with a hexadecimal value?

A byte, consisting of 8 bits, directly corresponds to two hexadecimal digits because each hexadecimal digit represents 4 bits, making it an efficient and compact way to express binary data.

Explanation:

An IPv4 subnet is 32-bit, show with a calculation why this is the case.

An IPv4 address is composed of 4 octets.

Each octet is 8 bits by definition, as it can represent values from 0 to 255 numbers.

Total bits tere are 4 octets in an IPv4 adress so the total number of bits is :

4octets x 8bits per octet = 32 bits.

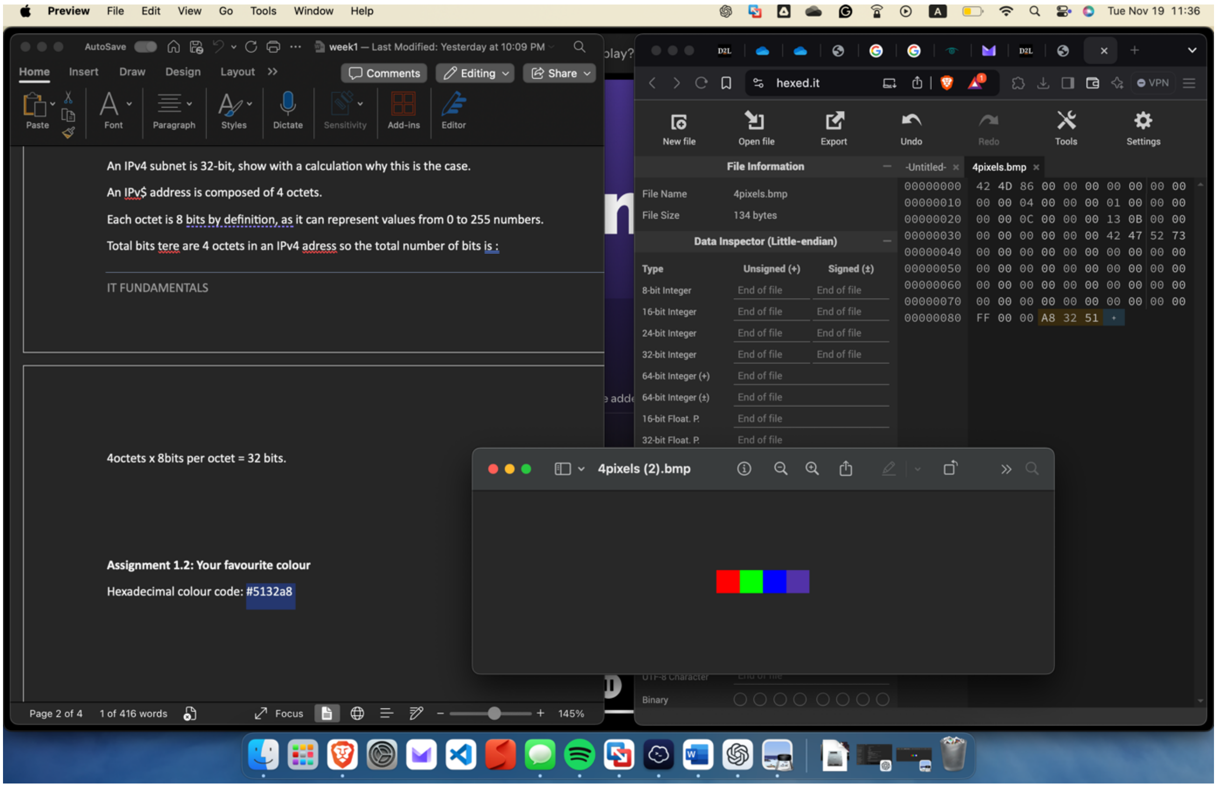
Assignment 1.2: Your favourite colour

Hexadecimal colour code: #5132a8

Assignment 1.3: Manipulating binary data

Colour	Colour code hexadecimaal (RGB)	Big Endian	Little Endian
RED	#FF0000	FF 00 00	00 00 FF
GREEN	#00FF00	00 FF 00	00 FF 00
BLUE	#0000FF	00 00 FF	FF 00 00
WHITE	#FFFFFF	FF FF FF	FF FF FF
Favourite **(previous assignment)**	#5132a8	51 32 A8	A8 32 51

Screenshot modified BMP file in hex editor:



Preview File Edit View Go Tools Window Help
Tue Nov 19 11:36
AutoSave
week1 — Last Modified: Yesterday at 10:09 PM
Home Insert Draw Design Layout
Comments Editing Share
Paste Font Paragraph Styles Dictate Sensitivity Add-ins Editor
Page 2 of 4 1 of 416 words Focus 145%
hexed.it
New file Open file Export Undo Redo Tools Settings
-Untitled- 4pixels.bmp
4pixels (2).bmp

An IPv4 subnet is 32-bit, show with a calculation why this is the case.

An IPv$ address is composed of 4 octets.

Each octet is 8 bits by definition, as it can represent values from 0 to 255 numbers.

Total bits tere are 4 octets in an IPv4 adress so the total number of bits is :

IT FUNDAMENTALS

4octets x 8bits per octet = 32 bits.

Assignment 1.2: Your favourite colour

Hexadecimal colour code: #5132a8

File Information

File Name	4pixels.bmp
File Size	134 bytes

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (±)
8-bit Integer	End of file	End of file
16-bit Integer	End of file	End of file
24-bit Integer	End of file	End of file
32-bit Integer	End of file	End of file
64-bit Integer (+)	End of file	
64-bit Integer (±)	End of file	
16-bit Float. P.	End of file	
32-bit Float. P.	End of file	

```
00000000  42 4D 86 00 00 00 00 00 00 00
00000010  00 00 04 00 00 00 01 00 00 00
00000020  00 00 0C 00 00 00 13 0B 00 00
00000030  00 00 00 00 00 00 42 47 52 73
00000040  00 00 00 00 00 00 00 00 00 00
00000050  00 00 00 00 00 00 00 00 00 00
00000060  00 00 00 00 00 00 00 00 00 00
00000070  00 00 00 00 00 00 00 00 00 00
00000080  FF 00 00 A8 32 51 +
```

Bonus point assignment – week 1

Convert your student number to a hexadecimal number and a binary number.

Explain in detail that the calculation is correct. Use the PowerPoint slides of week 1.

569527

Divide by 16 Repeatedly:

Start by dividing 569527 by 16, recording the quotient and remainder.

•First Division:

569527 \div 16 = 35595 remainder 7 .

(Write down 7 as the least significant digit in hexadecimal).

•Second Division:

35595 \div 16 = 2224 remainder 11 .

(Write down B as the next digit; 11 = B in hexadecimal).

•Third Division:

2224 \div 16 = 139 remainder 0 .

(Write down 0).

•Fourth Division:

139 \div 16 = 8 remainder 11 .

(Write down B).

•Fifth Division:

8 \div 16 = 0 remainder 8 .

Combine Remainders in Reverse Order:

569527 in hexadecimal = 8B0B7

BINARY:

1. 569527 \div 2 = 284763 , remainder = 1
2. 284763 \div 2 = 142381 , remainder = 1
3. 142381 \div 2 = 71190 , remainder = 1
4. 71190 \div 2 = 35595 , remainder = 0
5. 35595 \div 2 = 17797 , remainder = 1
6. 17797 \div 2 = 8898 , remainder = 1

7. 8898 \div 2 = 4449 , remainder = 0
8. 4449 \div 2 = 2224 , remainder = 1
9. 2224 \div 2 = 1112 , remainder = 0
10. 1112 \div 2 = 556 , remainder = 0
11. 556 \div 2 = 278 , remainder = 0
12. 278 \div 2 = 139 , remainder = 0
13. 139 \div 2 = 69 , remainder = 1
14. 69 \div 2 = 34 , remainder = 1
15. 34 \div 2 = 17 , remainder = 0
16. 17 \div 2 = 8 , remainder = 1
17. 8 \div 2 = 4 , remainder = 0
18. 4 \div 2 = 2 , remainder = 0
19. 2 \div 2 = 1 , remainder = 0
20. 1 \div 2 = 0 , remainder = 1

Combine remainders in reverse order:

10001011000010110111

Ready? Save this file and export it as a pdf file with the name: **week1.pdf**