# Activating Existing Knowledge

Decimal (aka "Base 10", meaning 10 symbols available: 0,1,2,3,4,5,6,7,8,9) is the most common number system we use for counting.

Example: Breakdown the decimal system number 103 (one hundred and three):

|  |  |  |  |
| --- | --- | --- | --- |
| *Units* | *100* | *10* | *1* |
| Number: | 1 | 0 | 3 |
| Proof: | 1 x *100* = 100 | 0 x *10* = 0 | 3 x *1* = 3 |
| 100 + 0 + 3 = 103 |

# Warm-up exercises

Exercise: Breakdown the decimal system number 496 (four hundred and ninety-six) as per the table above:

|  |  |  |  |
| --- | --- | --- | --- |
| *Units* | *100* | *10* | *1* |
| Number: | 4 | 9 | 6 |
| Proof: |  |  |  |
|  |

Exercise: Breakdown the decimal system number 5080 (five thousand and eighty) as per the table above:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Units* | *1000* | *100* | *10* | *1* |
| Number: | 5 | 0 | 8 | 0 |
| Proof: |  |  |  |  |
|  |

# Introducing New Knowledge

Computers fundamentally use patterns of electrical signals "off" and "on" to represent data. This is known as a "Base 2" or **Binary** system, meaning there are 2 symbols available: 0 & 1, where 0 represents "off" and 1 represents "on". This system can be used to encode into binary any number from the decimal number system:

Example #1: Express the decimal system number 5 as a binary number:

|  |  |  |  |
| --- | --- | --- | --- |
| *Units* | *4* | *2* | *1* |
| Binary Number: | 1 | 0 | 1 |
| Proof: | 1 x *4* = 4 | 0 x *2* = 0 | 1 x *1* = 1 |
| 4 + 0 + 1 = 5 |

*The binary representation for the decimal system number 5 is*: 101.

Example #2: Express the decimal system number 13 as a binary number:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Units* | *8* | *4* | *2* | *1* |
| Binary Number: | 1 | 1 | 0 | 1 |
| Proof: | 1 x *8* = 8 | 1 x *4* = 4 | 0 x *2* = 0 | 1 x *1* = 1 |
| 8 + 4 + 0 + 1 = 13 |

*The binary representation for the decimal system number 13 is*: 1101.

Example #3: Express the decimal system number 103 as a binary number:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Units* | *64* | *32* | *16* | *8* | *4* | *2* | *1* |
| Binary Number: | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| Proof: | 1 x *64* = 64 | 1 x *32* = 32 | 0 x *16* = 0 | 0 x *8* = 0 | 1 x *4* = 4 | 1 x *2* = 2 | 1 x *1* = 1 |
| 64 + 32 + 0 + 0 + 4 + 2 + 1 = 103 |

*The binary representation for the decimal system number 103 is*: 1100111.

# Binary exercises

Exercise: Express the decimal system number 3 as a binary number:

|  |  |  |
| --- | --- | --- |
| *Units* | *2* | *1* |
| Binary Number (remember you can only use 0 or 1): | \_ | \_ |
| Proof: | \_ x *2* = \_ | \_ x *1* = \_ |
|  \_+ \_ = \_ |

Exercise: Express the decimal system number 6 as a binary number:

|  |  |  |  |
| --- | --- | --- | --- |
| *Units* | *4* | *2* | *1* |
| Binary Number: | \_ | \_ | \_ |
| Proof: | \_ x *4* = \_ | \_ x *2* = \_ | \_ x *1* = \_ |
| \_+ \_+ \_ = \_ |

Exercise: Express the decimal system number 10 as a binary number:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Units* | *8* | *4* | *2* | *1* |
| Binary Number: | \_ | \_ | \_ | \_ |
| Proof: | \_ x 8 = \_ | \_ x *4* = \_ | \_ x *2* = \_ | \_ x *1* = \_ |
| \_+ \_+ \_+ \_ = \_\_ |

Exercises: Express these decimal system numbers as binary numbers. Work them out on a separate piece of paper. No need for proof this time:

|  |  |
| --- | --- |
| Decimal Number | Binary Number |
| 0 |  |
| 5 |  |
| 14 |  |
| 16 |  |
| 30 |  |
| 97 |  |

# Reversing Binary to Decimal

Example: Express the binary number 1001 as a decimal system number:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Units* | *8* | *4* | *2* | *1* |
| Binary Number: | 1 | 0 | 0 | 1 |
| Calculation: | 1 x *8* = 8 | 0 x *4* = 0 | 0 x *2* = 0 | 1 x *1* = 1 |
| 8 + 0 + 0 + 1 = 9 |

*The decimal number for the binary number* 1001 *is*: 9.

Exercises: Express these binary numbers as decimal numbers. Work them out on a separate piece of paper:

|  |  |
| --- | --- |
| Binary Number | Decimal Number |
| 1 |  |
| 10 |  |
| 100 |  |
| 1100 |  |
| 1111 |  |
| 11001 |  |
| 101011 |  |
| 1110101 |  |

|  |
| --- |
| *Interesting fact:*The 0 & 1 symbols available for representing binary numbers are known as **Binary Digits**.Computer scientists often shorten the word **Binary Digits** to '**Bits**'. Both words refer to the same thing – the zeros and ones used to represent digital data. For historical reasons, computer scientists refer to a grouping of eight binary digits together as a **Byte**. You’ll often see internet speeds measured in megabits or megabytes, hard drives measured in gigabits or gigabytes, and other measures involving bits and bytes. These are all a measure of zeros and ones, to measure either speed or size. |

# Extension: Other Base Systems

Base systems allow **Bn** combinations in each unit column, where **n** is the number of columns, and **B** is the number of different base symbols available:

Example #1: Calculate the ***Units*** column for the decimal number system:

|  |  |
| --- | --- |
| Different base symbols available: | 0,1,2,3,4,5,6,7,8,9: 10 different symbols, therefore:**B=10** |
| Unit calculation: | **102** | **101** | **100** |
| *Units:* | *100* | *10* | *1* |
| Proof: | 10 x 10 = 100 | Rule: Any number to power of 1 is itself | Rule: Any number to power of 0 is 1 |

Example #2: Calculate the ***Units*** column for the binary number system:

|  |  |
| --- | --- |
| Different base symbols available: | 0,1: 2 different symbols, therefore:**B=2** |
| Unit calculation: | **25** | **24** | **23** | **22** | **21** | **20** |
| *Units:* | *32* | *16* | *8* | *4* | *2* | *1* |
| Proof: | 2 x 2 x 2 x 2 x 2 = 32 | 2 x 2 x 2 x 2 = 16 | 2 x 2 x 2 = 8 | 2 x 2 = 4 | Rule: Any number to power of 1 is itself | Rule: Any number to power of 0 is 1 |

# Case study: Hexadecimal

The hexadecimal ("hex" for short) system is a **Base 16** system used in password hashing, colour encoding and Bitcoin mining. The 16 symbols and values are:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Values*: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Symbol: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |

Using this knowledge, convert the hex **b8** to a decimal number:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unit calculation: | **163** | **162** | **161** | **160** |
| *Units:* | *4096* | *256* | *16* | *1* |
| Hexadecimal number: |  |  | **b** | **8** |
| Calculation to convert to decimal: |  |  | b is 11, so:11x*16*=176 | 8x*1*=8 |
| 176 + 8 = 184 |

*The decimal number for hexadecimal representation* b8 *is*: 184.

# Challenge exercises: Hexadecimal

Fill in the missing table. The first row is completed for you:

|  |  |
| --- | --- |
| Hexadecimal | Decimal Number |
| b8 | 184 |
|  | 14 |
|  | 33 |
| 12c |  |

# Topics to research for next lesson:

* ASCII: A **7-bit encoding standard** for representing English characters as numbers, for example:
	+ A = 65
	+ B = 66
	+ Z = 90