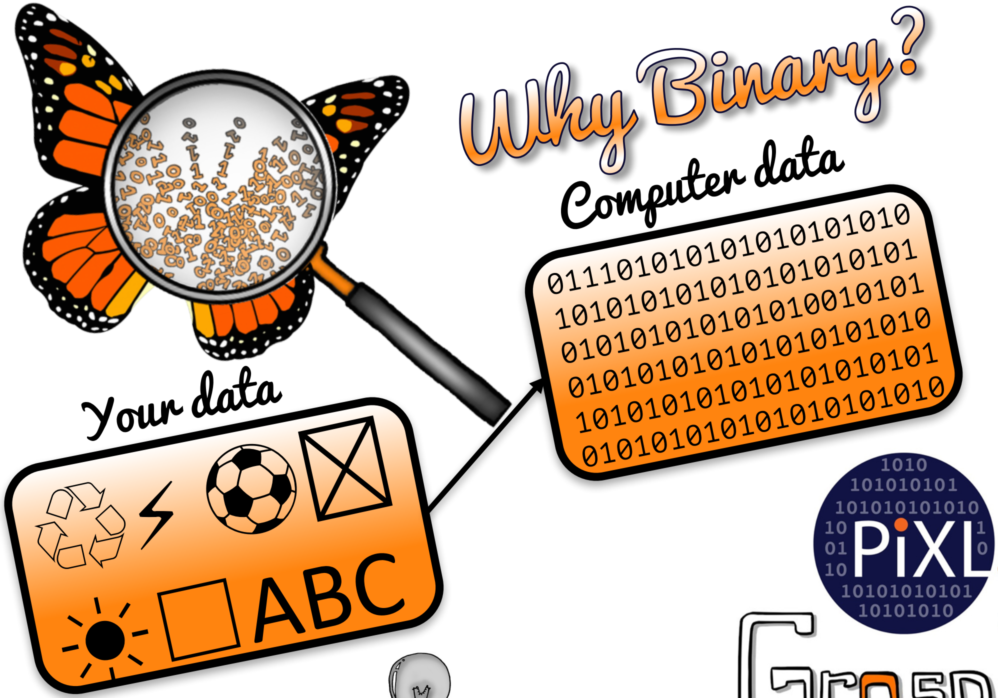
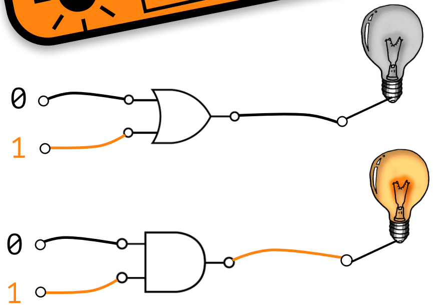
Why Binary?





BINARY DIGIT (BIT)	ELECTRONIC CHARGE	ELECTRONIC SYSTEM
1	●	ON
0	●	OFF

GraspIT

COMPUTATIONAL LOGIC

OPERATOR	DESCIPTION	EXAMPLE
+	ADDS 2 VALUES	2 + 34 = 36
-	SUBTRACTS ONE VALUE FROM ANOTHER	34 - 2 = 32
*	MULTIPLIES BOTH VALUES	2 * 34 = 68
/	DIVIDES ONE NUMBER BY ANOTHER	34 / 2 = 17
MOD	REMAINDER MODULUS – FINDS THE REMAINDER OF A DIVISION	39 MOD 2 = 1
DIV	INTEGER DIVISION – FINDS THE INTEGER PART OF A DIVISION	39 DIV 2 = 19
^	EXPONENTIAL – CALCULATES THE RESULT OF RAISING TO A POWER	2 ^ 4 =

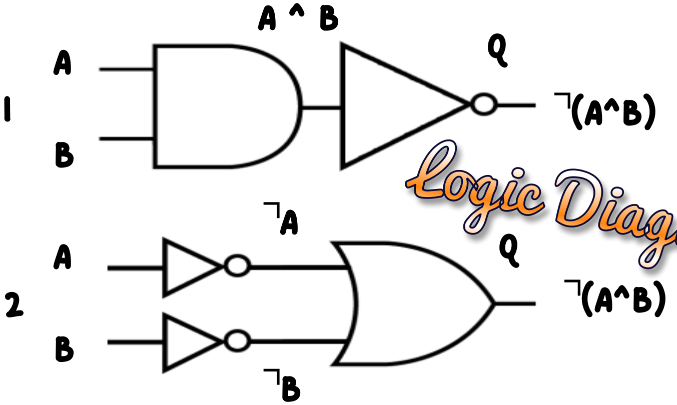
LOGICAL OPERATORS

OPERATOR	EXAMPLE

AND ^	x = 15
	y = 9
	x < 16 ^ y >8 RETURN TRUE

OR ^	x = 15 y – 9 x = = 8 V y = = 5 RETURN FALSE

NOT ¬	x = 16
	y = 9
	¬(x = = y) RETURN TRUE



Logic Diagrams & Truth Tables

GATE	A	B	Q
1	1	1	0
2	1	1	0

1 bit
1 nibble (4 bits)
1 byte (8 bits)
KB 1 kilobyte (1024 bytes)
MB 1 megabyte (1024 kilobytes)
GB 1 gigabyte (1024 megabytes)
TB 1 terabyte (1024 gigabytes)
PB 1 petabyte (1024 terabytes)

Why Binary?

- Know that an ‘instruction’ is a set of binary digits from 4 bits to several bytes in length
- Be able to discuss how a processor characterization eg 32 bit, 64 bit relates to the size of the instructions and memory
- Know that a binary file will also also contain metadata. Explain why meta data is important.
- Be able to discuss the link between logic gates and binary making it clear that the bit value output from a logic gate forms part of an instruction set

Computer Mathematics

Know how to work with variable values:
num1 = 39, num2 = 71, total = num1 ? num2
grossValue = 101, netValue = 71, deductions = grossValue ? netValue
radius = 10, pi = 3.142, circle = pi ?(r^r)
triangle = b * h ? 2, print(triangle)
Check if a number is even with MOD: IF numIn MOD 2 != 0 print(“Odd number”)
Check if a number is an integer with DIV: IF numIn DIV 1 == numIn print(“Integer”)
Calculate the value with base and exponent:
input base (5), input exponent (2), product = base ^ exponent (25)

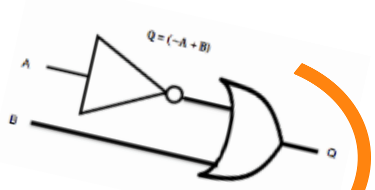
Mathematical Terms and Symbols
MOD : % (percentage sign)
DIV : // (double division sign)
Exponent : ** (double asterisk)

PiXL Computing

GraspIT

COMPUTATIONAL LOGIC

Logic Diagrams & Truth Tables



A	~A	B	Q
0	1	0	1
1	0	0	0
0	1	1	1
1	0	1	1

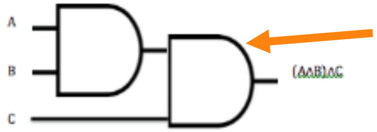
Be able to interpret logic statements and produce the circuit and truth table:
eg (A∧B) ∧C

Understand equivalence:
AND = ∧ **(A ∧ B) , OR = ∨, + (A ∨ B = A+B) ,**

NOT = ¬ , ‾ , ~ (¬A, A, ~A)

Be able to calculate the number of rows required in a truth table based on the inputs and/or logic gates: rows = 2^n where n = inputs (3 inputs = 2*2*2)

EG ¬ ((A ∧ B) ∧ C)
Inputs = 3 (A, B, C)



A	B	A∧B	C	(A∧B)∧C
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
0	0	0	1	0
1	1	1	0	0
0	1	0	1	0
1	0	0	1	0
1	1	1	1	1