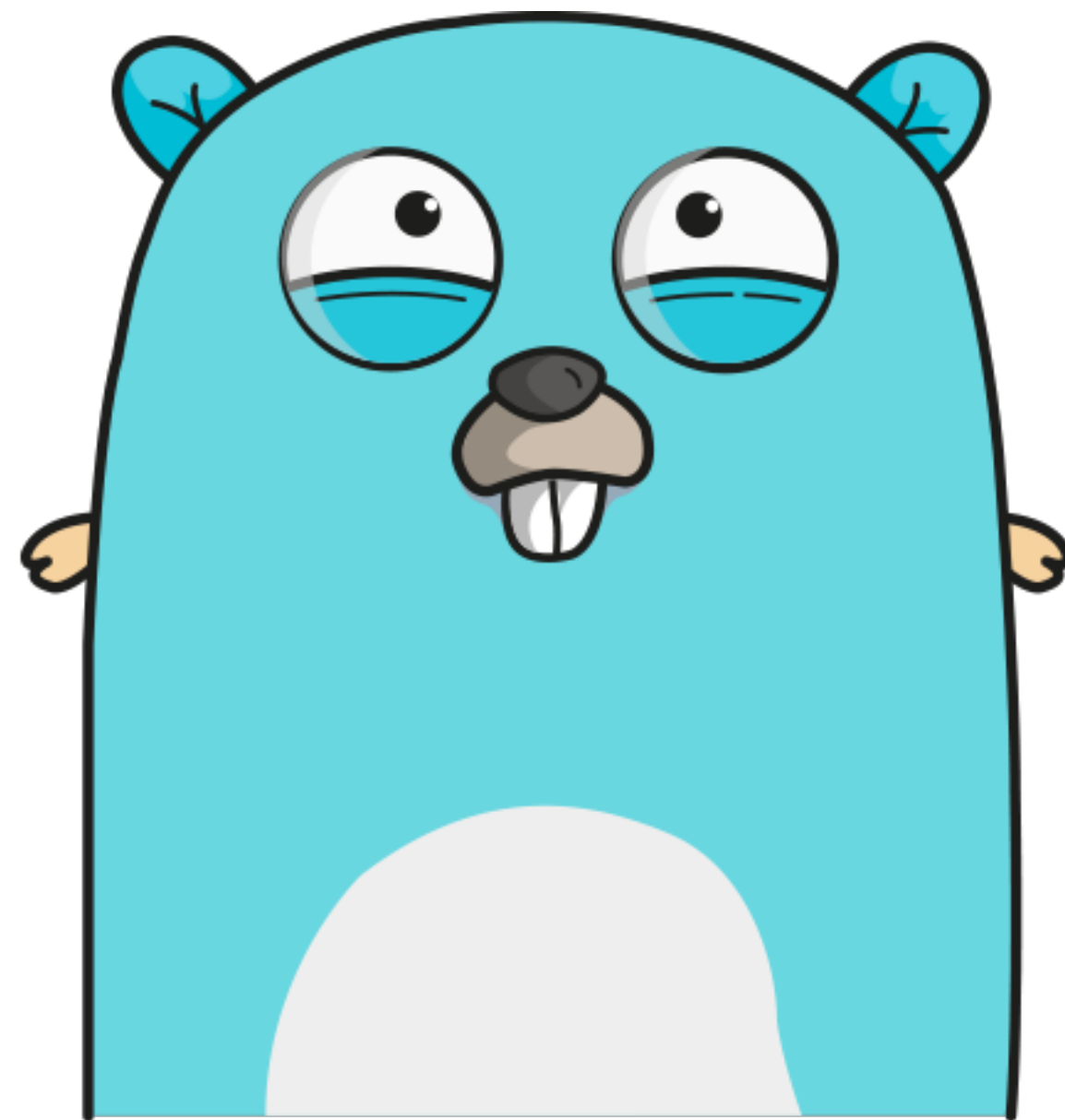
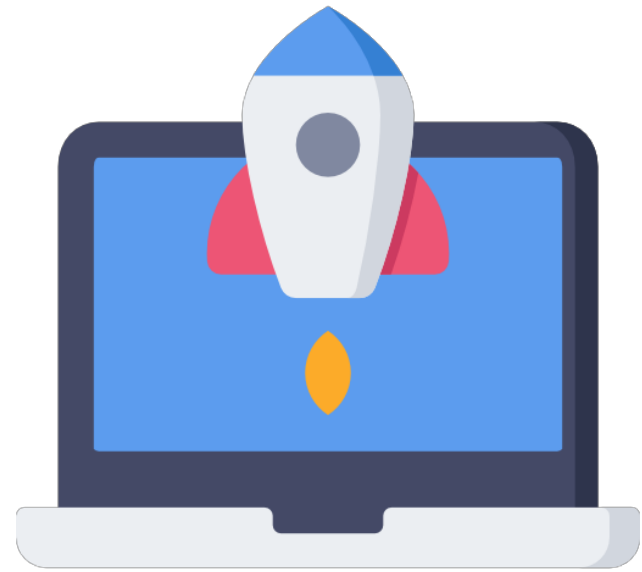


HOW COMPUTERS WORK? MEMORY MANAGEMENT

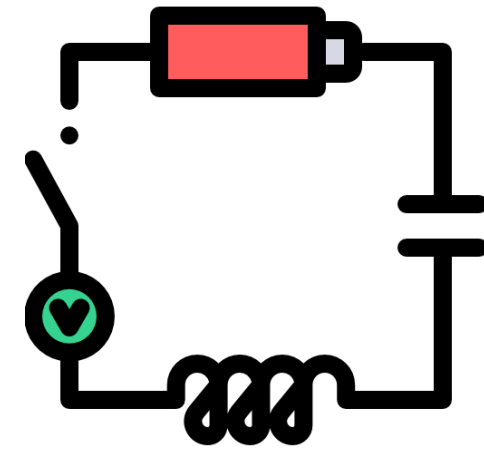


 GOPHERTUTS

HOW COMPUTERS WORK?

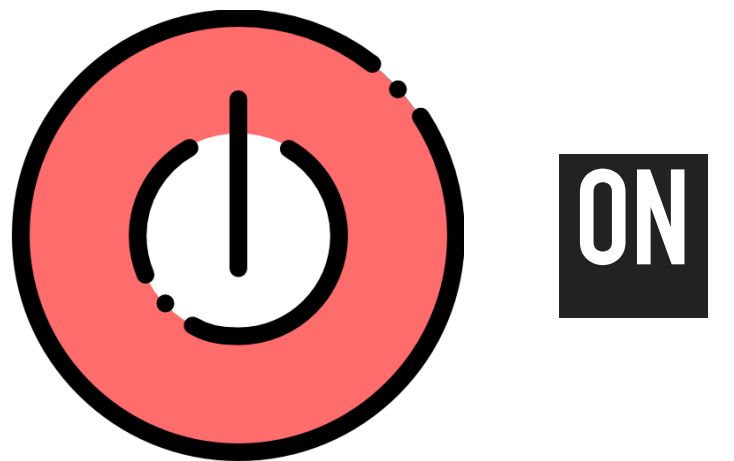
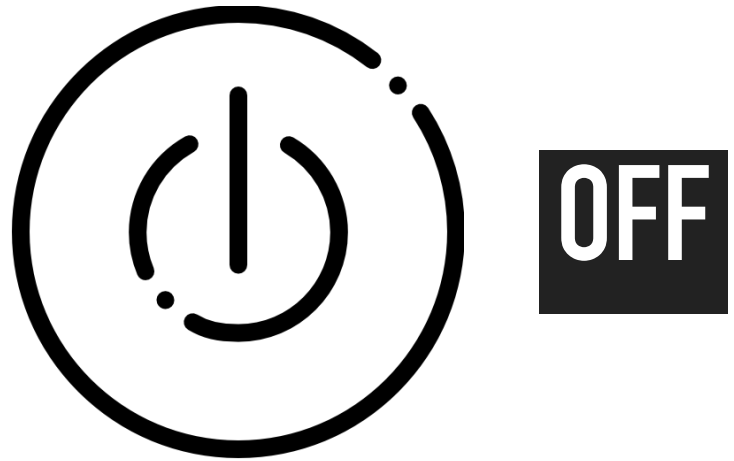


RUN ON ELECTRICITY



MADE OF CIRCUITS/SWITCHES

STATE REPRESENTATION



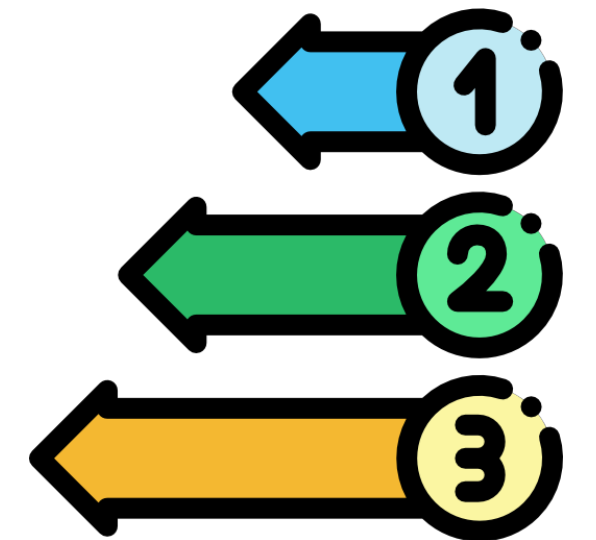
NUMBER SYSTEMS

BASE 2 = 2 DIGITS (0, 1) **BINARY**

BASE 8 = 8 DIGITS (0, 1, 2, 3, 4, 5, 6, 7) **OCTAL**

BASE 10 = 10 DIGITS (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) **DECIMAL**

BASE 16 = 10 DIGITS + 6 LETTERS (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) **HEXADECIMAL**



DECIMAL

HUMAN UNDERSTANDABLE VALUES

EASY TO USE AND REMEMBER

EASY TO PERFORM CERTAIN OPERATIONS ON

\$80K/YEAR FOR A PROGRAMMER SALARY

JOHN OWES ME \$50

21% TAXES FOR \$5000 SALARY = 5000*21/100 = \$1050



BINARY

COMPUTER LANGUAGE

BINARY = ON/OFF = 0/1 = TRUE/FALSE = STATE1/STATE2 = BIT

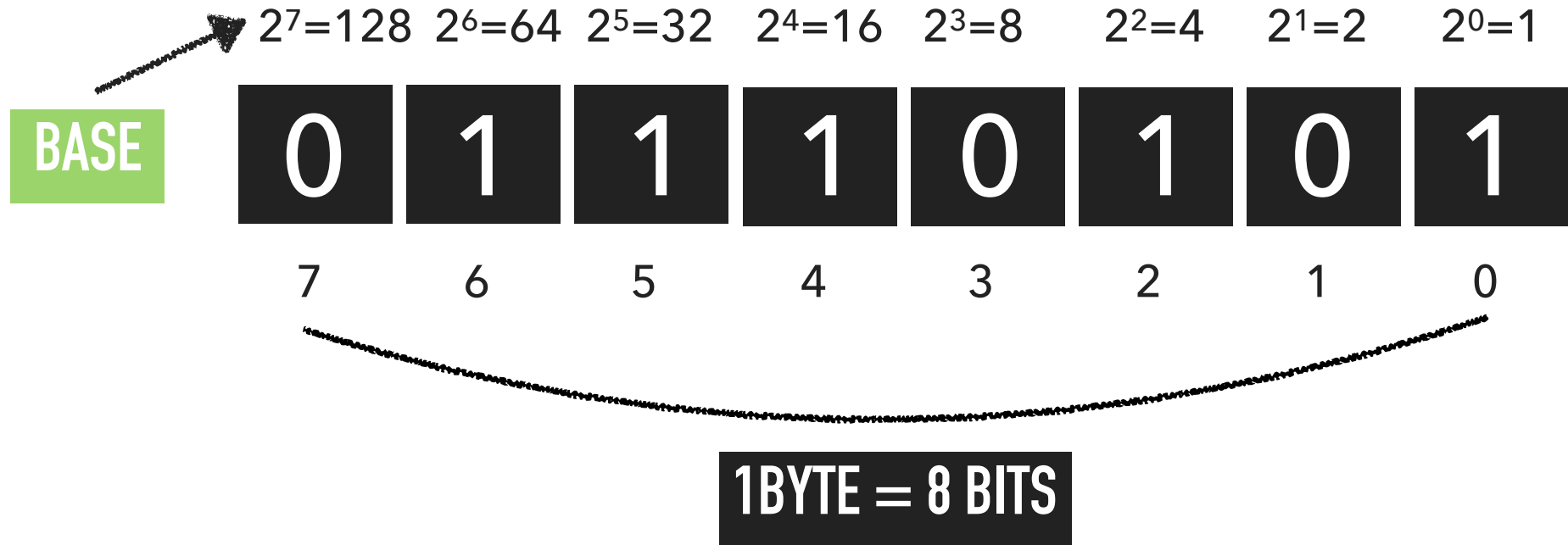
1 BYTE = 8 BITS

117₁₀ = 01110101₂ = WTF?

11001
01011
1000

0 + 1 + 1 + 1 + 0 + 1 + 0 + 1 = 117

$2^7=128$ $2^6=64$ $2^5=32$ $2^4=16$ $2^3=8$ $2^2=4$ $2^1=2$ $2^0=1$



HEXADECIMAL

USED TO REPRESENT BIG NUMBERS (MEMORY ADDRESS FOR EXAMPLE)

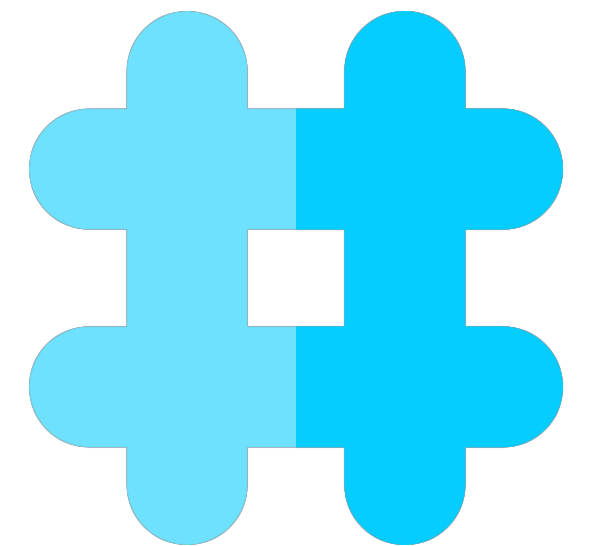
2FE
HEX

=

766
10

/ - DIVISION WITHOUT REMAINDER
 $25/2 = 12$

% - MODULO - REMAINDER OF DIVISION
 $12*2=24$; $25\%2 = 25-24 = 1$



$$2*16^2 + 15*16^1 + 14*16^0 = 766$$

2 F E

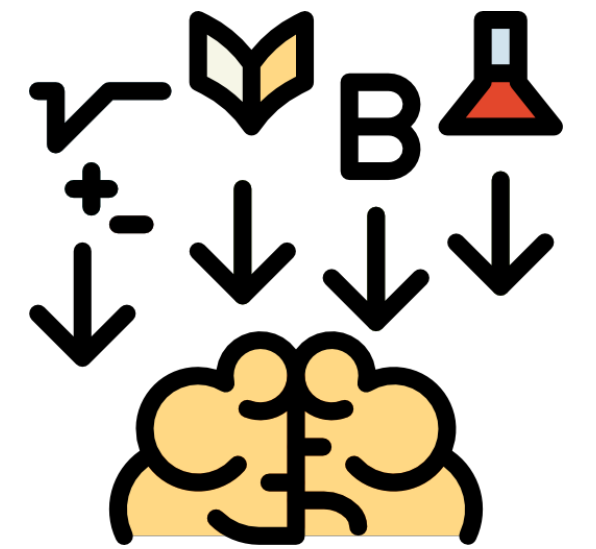
	$766 / 16$	=	47	
2	$766 \% 16$	=	14	=> E
	$47 / 16$	=	2	
1	$47 \% 16$	=	15	=> F
	$2 / 16$	=	0	
0	$2 \% 16$	=	2	=> 2
	=		2FE	

BASE 10 TO BASE N GENERALIZED

n / B \underline{n} - Decimal number
 $n \% B$ \underline{B} - Base

$\underline{n} = n / B$ on every iteration

Do this until $\underline{n} / B == 0$



BASE 10 TO BASE N EXAMPLE

N = 16

823

2

823 / 16

=

51

823 % 16

=

7

=>

7

1

51 / 16

=

3

51 % 16

=

3

=>

3

0

3 / 16

=

0

3 % 16

=

3

=>

3

=

337

N = 2

25

7

25 / 2

=

12

25 % 2

=

1

6

12 / 2

=

6

12 % 2

=

0

5

6 / 2

=

3

6 % 2

=

0

4

3 / 2

=

1

3 % 2

=

1

3

1 / 2

=

0

1 % 2

=

1

2

0

1

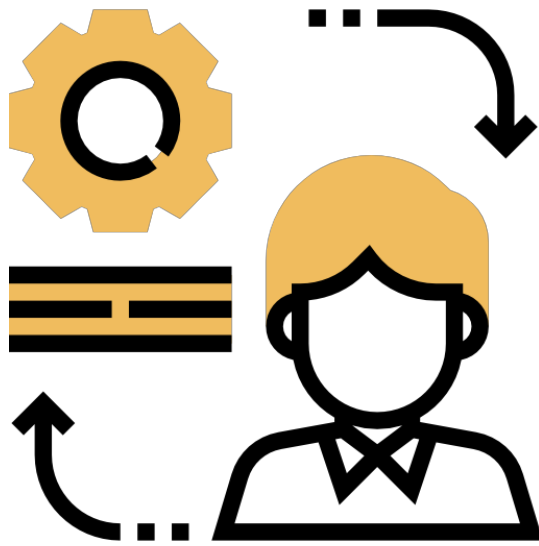
0

0

0

=

00011001



BASE N TO BASE 10 GENERALIZED

$$\sum_{i=n-1}^0$$

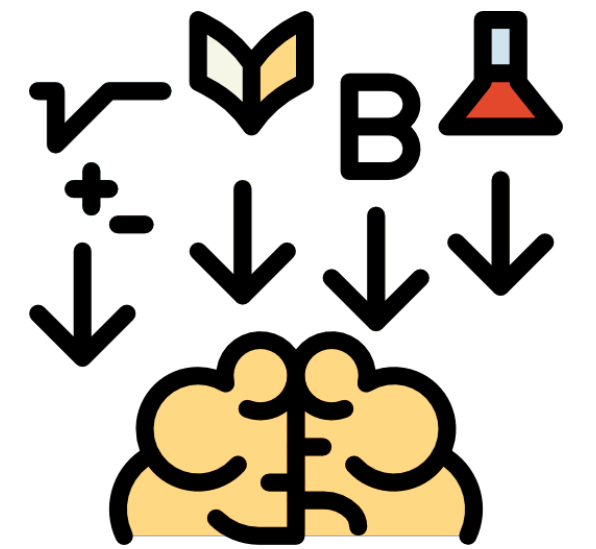
$$A_i * B^{|i - (n - 1)|}$$

,

n - number of symbols

A - list of symbols $\in B$

B - base



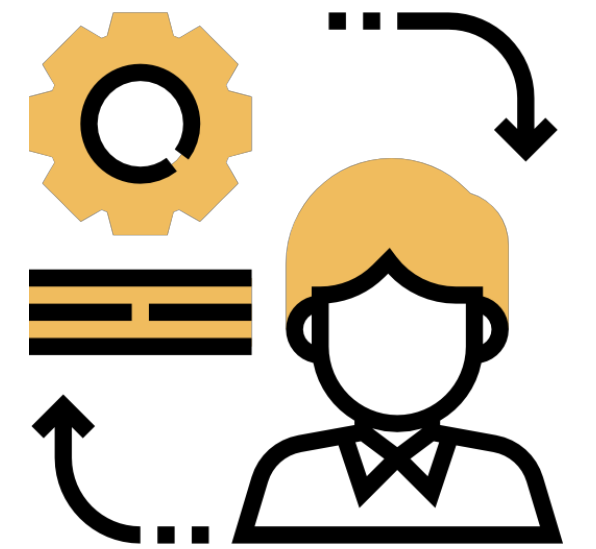
BASE N TO BASE 10 EXAMPLE

N = 16 ABE

$$14 \cdot 16^0 + 11 \cdot 16^1 + 10 \cdot 16^2 = 2750$$

N = 2 0110 1010 1010 0111

$$\begin{aligned} &1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7 + 0 \cdot 2^8 \\ &+ 1 \cdot 2^9 + 0 \cdot 2^{10} + 1 \cdot 2^{11} + 0 \cdot 2^{12} + 1 \cdot 2^{13} + 1 \cdot 2^{14} + 0 \cdot 2^{15} = 27303 \end{aligned}$$



EXAMPLES & CONVERSIONS

10: 115

2: 01110011 16: 73

16: 3AB

2: 0000001110101011 10: 939

2: 11110010

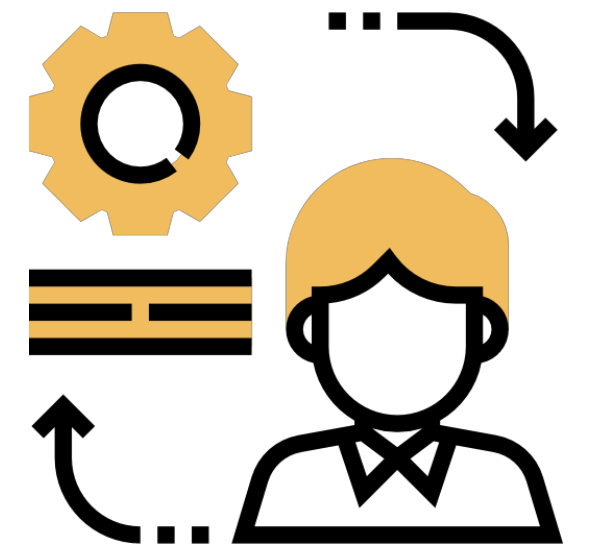
16: F2 10: 242

2: 11111111

16: FF 10: 255

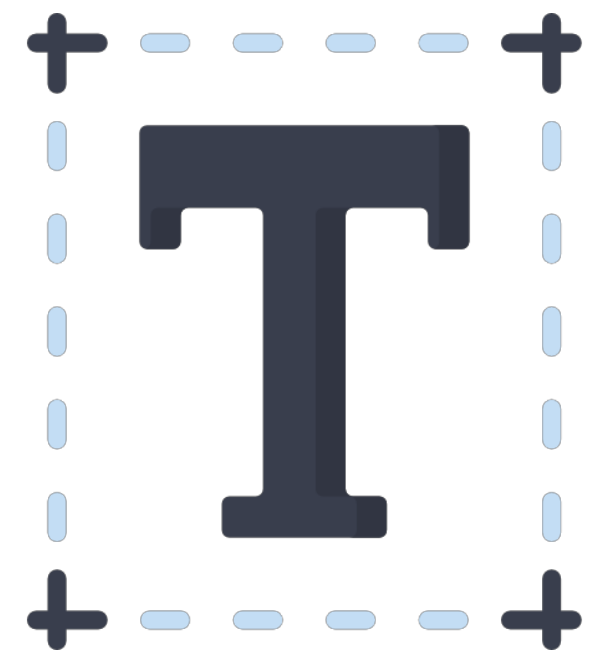
16: ABC

2: 0000101010111100 10: 2748



ASCII & UTF-8 (ENCODING SCHEMES)

84



1B ASCII / EXTENDED ASCII

$0-127 = 2^{(8*1) - 1} - 1 - 7 \text{ BITS}$

$0-255 = 2^{(8*1)} - 1 - 8 \text{ BITS}$

4B UTF-8

$0-4,294,967,295 = 2^{(8*4) - 1} - 8 \text{ BITS}$

ENCODING & DECODING EXAMPLE

HELLO

72 69 76 76 79

WORLD!

87 79 82 76 68 33

EXAMPLES & CONVERSIONS

72

H

01001000

228 184 173

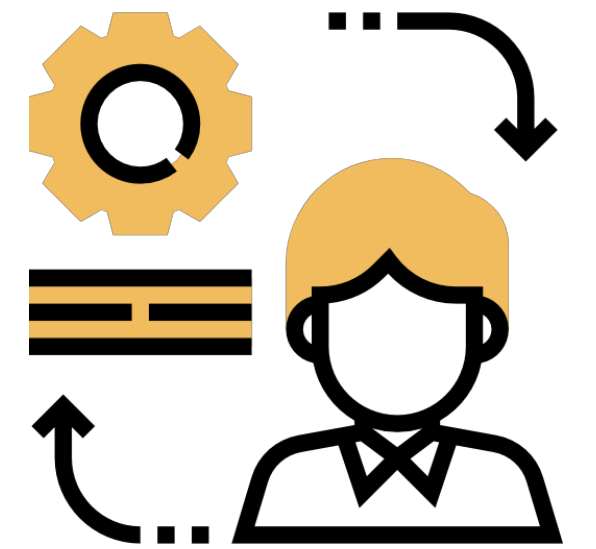
中

11100100 10111000 10101101

72 69 76 76 79

HELLO

01001000 01000101 01001100 01001100 01001111



PERSONAL ENCODING (STEVEN)

Steve 1	Steve 2	Meaning
Hand up	Hand down	Bring bear
Hand down	Hand up	Bring soda
Hand Up	Hand up	Bring bear & soda
Hand down	Hand down	Do not disturb

2^N – COMBINATIONS

SUMMARY



HOW COMPUTERS WORK?



NUMBER SYSTEMS



ENCODING SCHEMES