Las ondas eléctricas que según las frecuencia con la que se propagaban podían transmitir sonido, también podían ser expresadas como 1 y 0. A cada 1 y 0 lo llamaron Bit que posteriormente IBM al fabricar procesadores, decidieron llamar byte a la unidad mínima y útil de bits. Por eso 8 bits son 1 byte, era la unidad más pequeña y útil con la que se podía enviar información. Ya definidos los bits y bytes, con el sistema binario se decidió que una byte correspondería a una letra o símbolo. En la tabla ASCII esta la asignación byte-símbolo. Dentro de la tabla y según el procesador, hay ordenes para el procesador. Estas ordenes igualmente son asignación de bytes, pero son especiales, y recibieron el nombre de Assembler. Assembler a fin de cuentas es el código en el que se programan los procesadores. Por lo tanto, todo al final del día son bytes, por ejemplo las direcciones IP (Internet Protocol), que son 4 bytes, las imágenes…etc. La razón por la que un byte no es mayor a 255e porque dado que un byte son 8 bits, las combinaciones posibles son 256 es decir del 0 al 255.

Los computadores funcionan con Ceros y unos, pero no significa que existan en verdad (físicamente).

Hace mucho se descubrió que la electricidad funciona con ondas.

Las ondas controladas y pegadas a una membrana permitían guardar sonidos.

Ondas conectadas a rayos catódicos crearon la televisión, para transmitir vídeo.

Para enviar texto se empezó por usar el Código morse, que se basaba en la longitud de los tonos que se enviaban, pero era poco efectivo.

apareció la idea de transformar las ondas en tonos que suben y que bajan, para los tonos altos se designaría 1 y para los bajos 0.

Bit: es una subida de la Onda 1 o una bajada de la Onda 0.

IMB en los años 50 definió el concepto de Byte que serviría para almacenar un solo carácter, letra o número.

En esa época los CPU eran extremadamente costoso y la mínima unidad que encontraron fue de 8 espacios en las que solo se almacenarían ceros y unos.

le asignaron a cada posición binaría un valor:

|||||||||

128 64 32 16 8 4 2 1

por ejemplo:

1010 = 10

Porque:

0 = 0

1 = 2

0 = 0

1 = 8

2 + 8 = 10

Notar que el número siguiente (de derecha a izquierda) duplica al anterior.

Ejercicio:

10011011

1 = 1

1 = 2

0 = 0

1 = 8

1 = 16

0 = 0

0 = 0

1 = 128

128 + 16 + 8 + 2 + 1 = 155

Te reto a hacer estos:

10011010

11011111

10010111

Cada carácter o letra tiene un byte asignado incluidos los caracteres especiales

@ = alt + 64

por ende @ tiene su representación en bytes que es:

01000000

De aquí nace la tabla ASCII, que determina que byte corresponde a que carácter.

Los computadores saben que cada 8 bits deben hacer una pausa porque los bytes siempre equivalen a 8 bits.

Existen bytes que son especiales que pueden significar ordenes específicas para el ordenador como pedir que arranque, o que ejecute una operación matemática.

Juntando todos estos bytes especiales nace Assembler. Del cual se derivaron lo que hoy conocemos como lenguajes de programación que no son más que abstracciones de Assembler.

TODO EN NUESTRA HUMANIDAD ACTUAL SON BYTES!

Por ejemplo una ip:

127.0.0.1 (nada como el Hogar, guiño guiño)

es un conjunto de 4 números y ninguno es mayor a 255 porque solo existen 256 números que se pueden expresar con un byte (de 0 a 255).

Una imagen realmente es una grilla de pixeles y cada pixel representa a un byte cuyo número es igual a su color.

y que hay de los Emojis?

Se añadieron a la tabla ASCII y cada emoji equivale a 2 bytes.

Cuando se democratizo el acceso a la computación muchos países por obvias razones tenían idiomas diferentes con distintos caracteres y para solventar eso se creó el protocolo utf-8 que permitió que todo el mundo use la misma lista de caracteres.

Pero existían miles de caracteres que no cabían en 8 bits (utf-8) por lo que se diseñó utf-16 (2 bytes) y como sobro tanto espacio, pues a crear emojis.

Nota: Unicode es la Asociación que se encarga de mantener todo el asunto de ASCII y tal.