How computer work

Computer work on electricity

First generation computer is **eniac**

Second generation computer is **transistors**

Third generation computer is **integrated circuits**

Generation of Computers

1. Vacuum tubes
2. Transistors
3. Integrated circuits (chips)
4. Microprocessors (cpu’s)
5. AI (?)

*Circuits*, *switches*, *transistors*, and even “*gates*” are all words used to refer to this thing within a computer that can either be ON or OFF. It’s a *circuit*, it’s a *switch*. It’s a gate that can either be OPENED or CLOSED, it’s a *transistor*-you will learn that people use all of those words to talk about this same thing, this ability of computers to store ON/OFF states.

1 bit

8 bits 1 byte

10(00)24 bytes 1 KiloByte

1000 KiloBytes 1 MegaByte

1000 MegaBytes 1GigaByte

1000 GigaBytes 1 TeraByte

*ON & OFF, 1 & 0, Binary Digits, Bits, and Machine Language are all words used to refer to this idea that, within a computer, it’s all nothing but a bunch of ZERO’s and ONE’s, or switches that are ON or OFF, it’s all just a bunch of Binary Digits, or Bits, that’s the language which computers speak, it’s machine language.*

Coding scheme

Coding scheme is a set of codes, defined by the words and phrase that researchers assign to categorize a segment of the data by topic. To develop a preliminary coding scheme, researchers consider what questions they are trying to answer and related topics to those questions.

Example of coding schemes are EBCDIC, UTF-8, ADCII & Unicode.

Why is coding scheme used?

Coding Schemes are necessary to perform the Encoding process, involving the conversion of data from Human readable status to Machine understandable format, such as a Binary Language which consists of 0 and 1. Without having a Coding Scheme to convert data, computers wouldn’t be able to store, process, ot execute data.

Printing

General

%v the value in a default format

%#v a Go-syntax representation of the value

%T a Go-syntax representation of the type of the value

%% a literal percent sign; consumes no value

Boolean:

%t the word true or false

Integer:

%b base 2

%c the character represented by the corresponding Unicode

%d base 10

%o base 8

%q a single-quoted character literal safely escaped with Go

%x base 16, with lower-case letter for a-f

%X base 16, with upper-case letter for A-F

%U Unicode format: U+1234; same as “U+%04X”