Information Technology in Business and Society
- Technical Foundations

Pearl Yu

Outline

- Information Representation
- Information and hardware
- Computer Architecture
- OS, Softwares: Algorithm and Program

NYU
STERN SCHOOL OF BUSINESS

Information Representation

Objective:

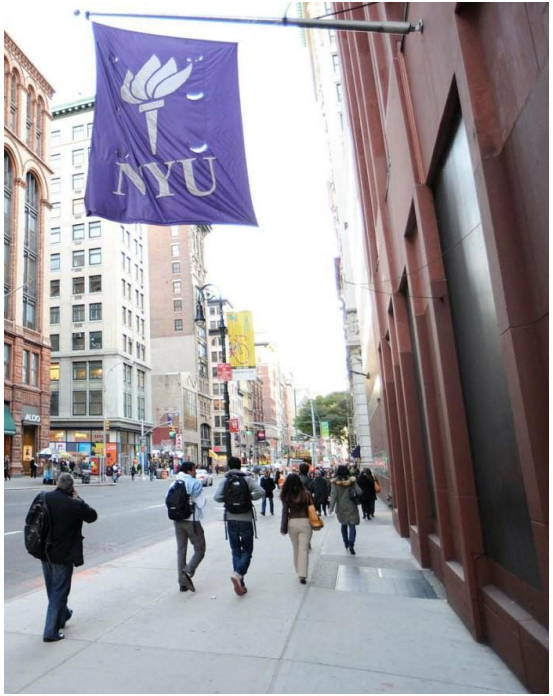
Understand how information is represented and processed in digital computer.

Information

- Humans can infer meanings from ambiguous symbols and signs. Natural languages, fuzzy patterns.

Information

- Humans can infer meanings from ambiguous symbols and signs. Natural languages, fuzzy patterns.



Information

- Humans can infer meanings from ambiguous symbols and signs. Natural languages, fuzzy patterns.
- Data we want computers to deal with has to be represented as bits, or, 0’s and 1’s.

Information

Information Representation: Number

(10 based) Decimal — Binary (2 based)

42 ⇔ 00101010

0 ⇔ 0

1 ⇔ 1

2 ⇔ 10

3 ⇔ 11

4 ⇔ 100

5 ⇔ 101

6 ⇔ 110

7 ⇔ 111

8 ⇔ 1000

We only have 2 symbols.

No symbol for 2. {0, 1}.

We add 1.

decimal.

9+1 --> 10

In binary

0 + 1 --> 1

1 + 1 ---> 10

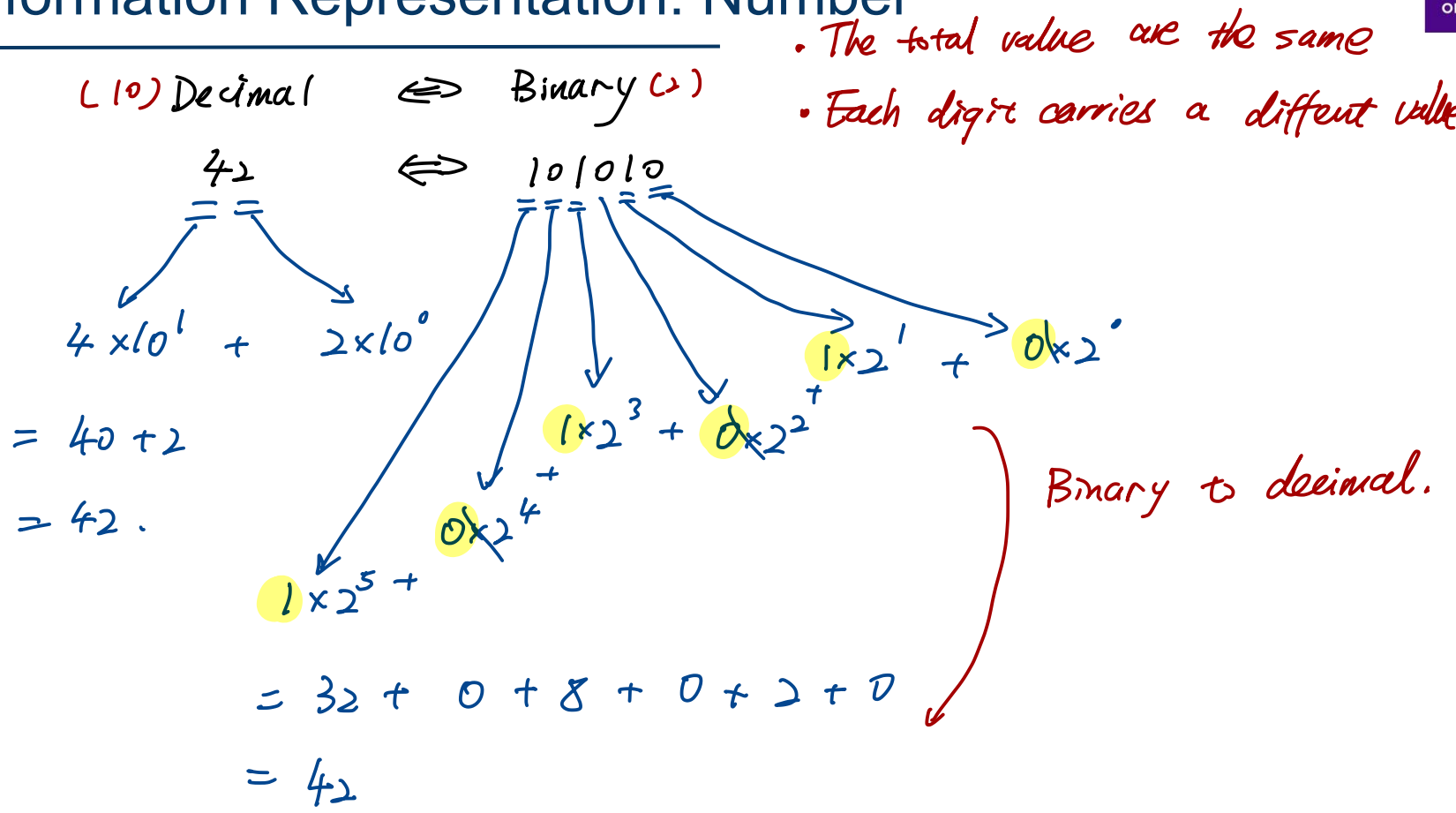
carry over

10 + 1 --> 11

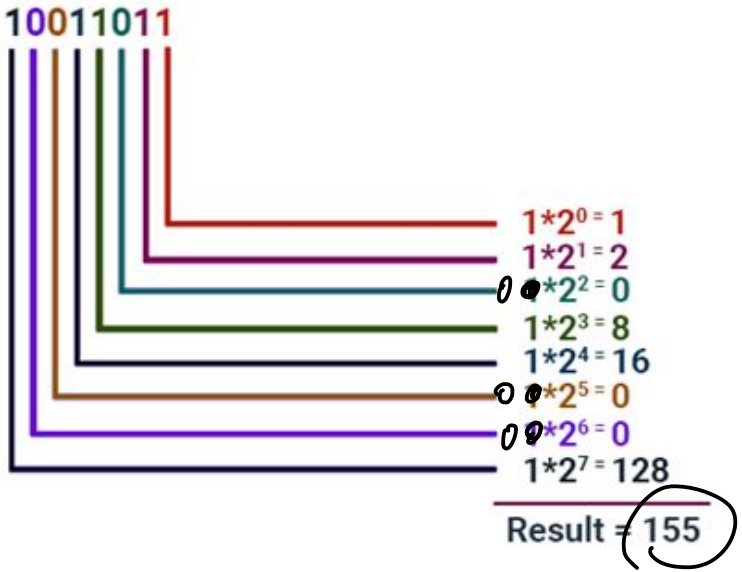
11 + 1 --> 100

carry twice

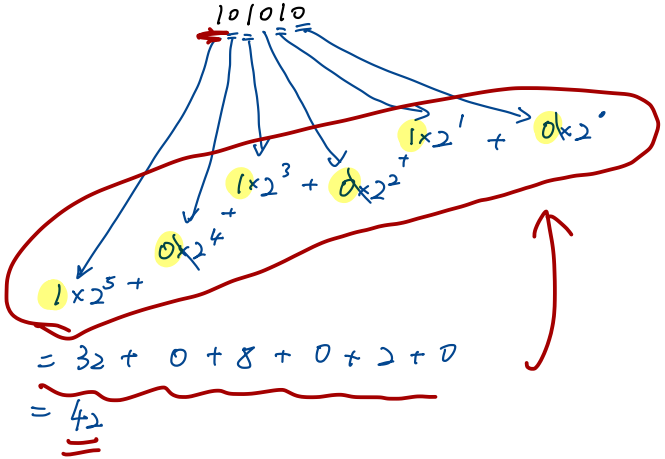
Information Representation: Number



Binary => Decimal conversion



Decimal => Binary conversion



Deconstruct the decimal into the sum of powers of two

Successive decision

	quotient	remainder
2 | 42		
2 | 21		0 LSB
2 | 10		1
2 | 5		0
2 | 2		1
2 | 1		0
	0	1 MSB

↑ read backwards

(most significant bit)

⇒ 101010

Decimal => Binary conversion

Proof

$$\underbrace{n}_{\text{Decimal}} = a_k \cdot 2^k + a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 \qquad a_i \in \{0, 1\}$$

$$2 \mid n$$

$$2 \mid a_k \cdot 2^{k-1} + a_{k-1} \cdot 2^{k-2} + \cdots + a_1$$

$\vdots$

remainder

a_0

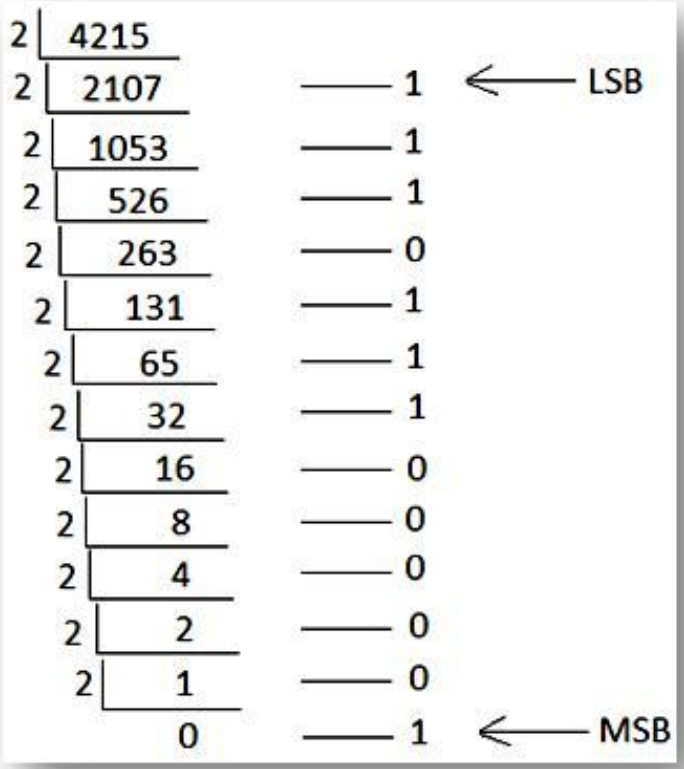
a_1

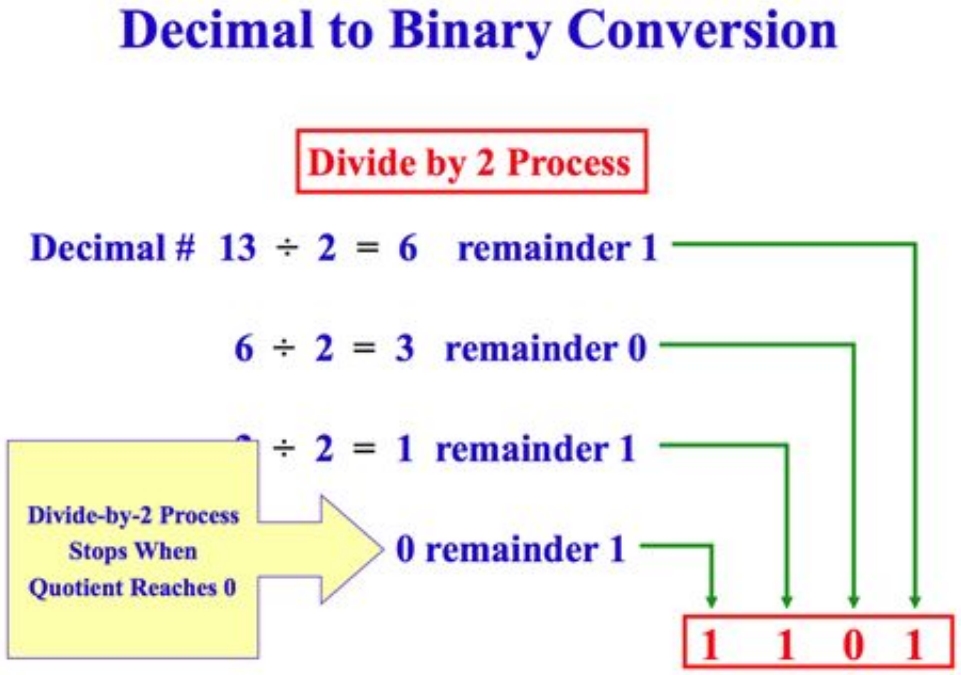
$\vdots$

a_{k-1}

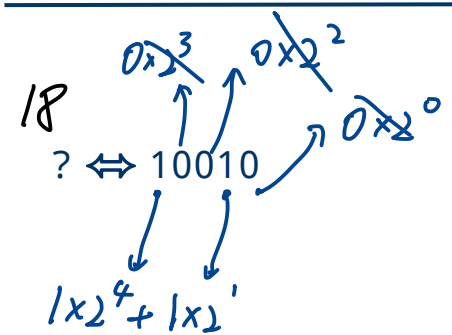
a_k.

Decimal => Binary conversion





Decimal => Binary conversion



321 ⇔ ?

101000001

2 | 321	
2 | 160	1
2 | 80	0
2 | 40	0
2 | 20	0
2 | 10	0
2 | 5	0
2 | 2	1
2 | 1	0
0	1

Information representation - Text

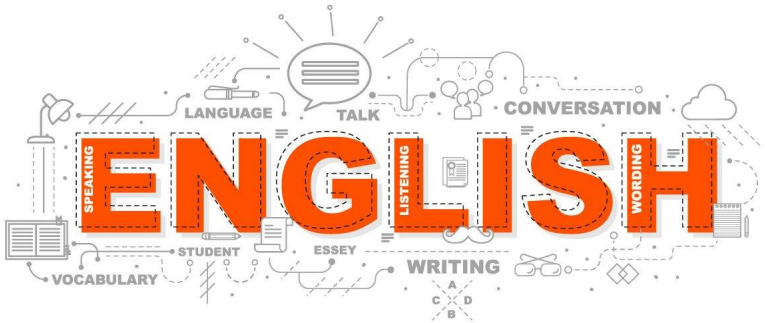
IT ⇔ 01001010 01010100

- ASCII: American Standard Code for Information Interchange.
- Created by American Standards Association (ASA) (today the American National Standards Institute or ANSI) in 1960s
- ASCII defines the standard for encoding characters for electronic communication.

ASCII - Binary Character Table

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

ASCII - Binary Character Table



26 letters □ Combinations

1 byte = 8 bits

漢 汉

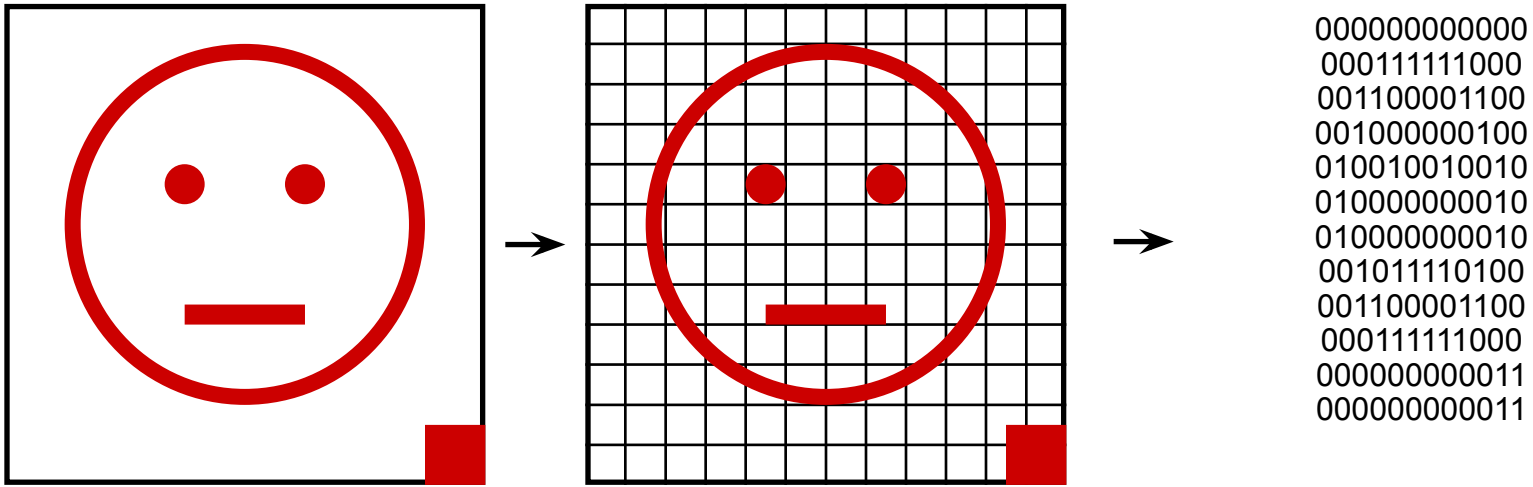
字 字

Symbols/ characters

2 bytes = more than 60,000 combinations

Information representation - Image

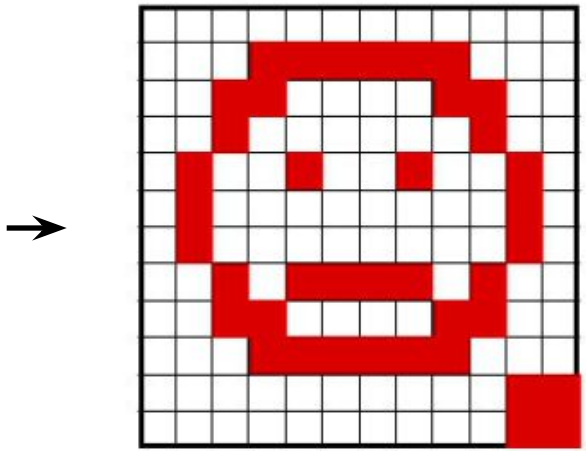
- How to digitize an image?

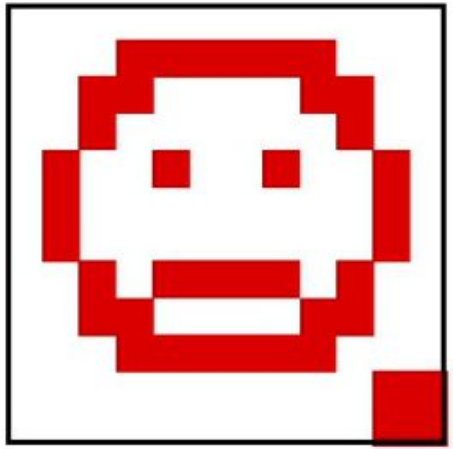


Information representation - Image

- How to digitize an image?

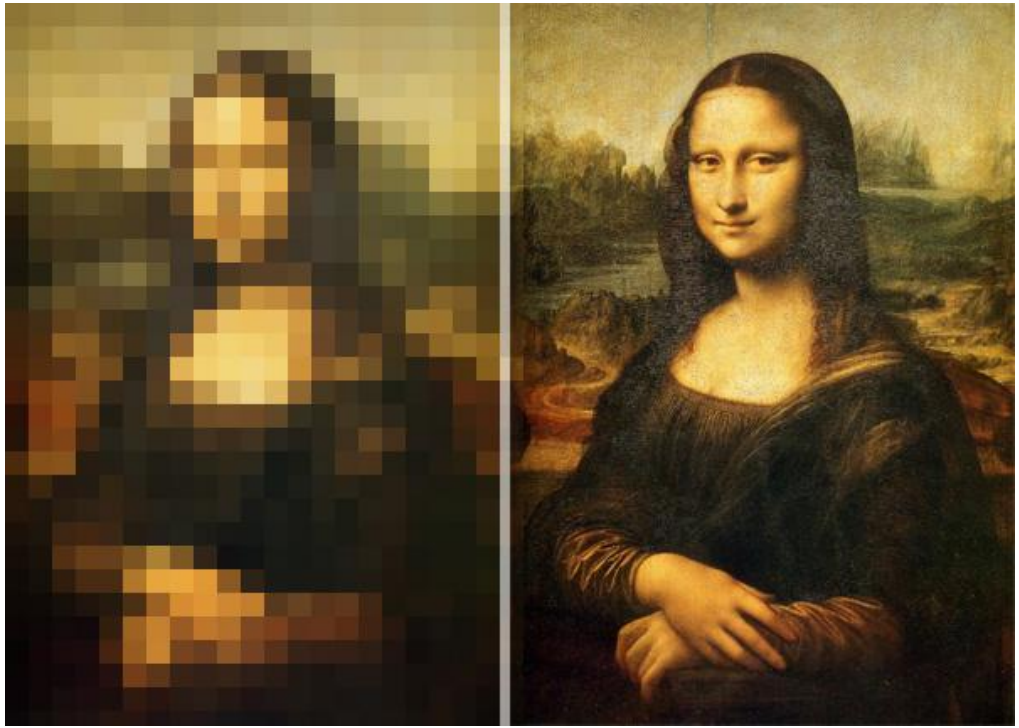
000000000000
000111111000
001100001100
001000000100
010010010010
010000000010
010000000010
001011110100
001100001100
000111111000
000000000011
000000000011





Information representation - Image

- How to digitize an image?

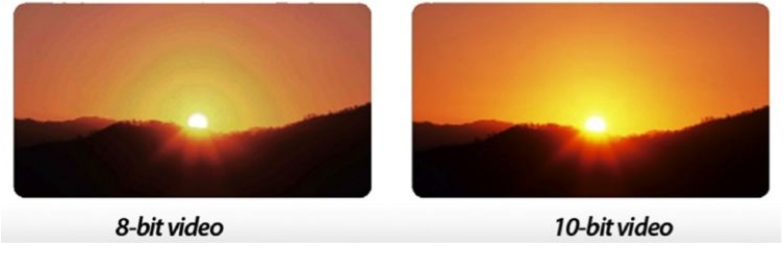


.gif, .jpeg, .bmp…

Information representation - Image

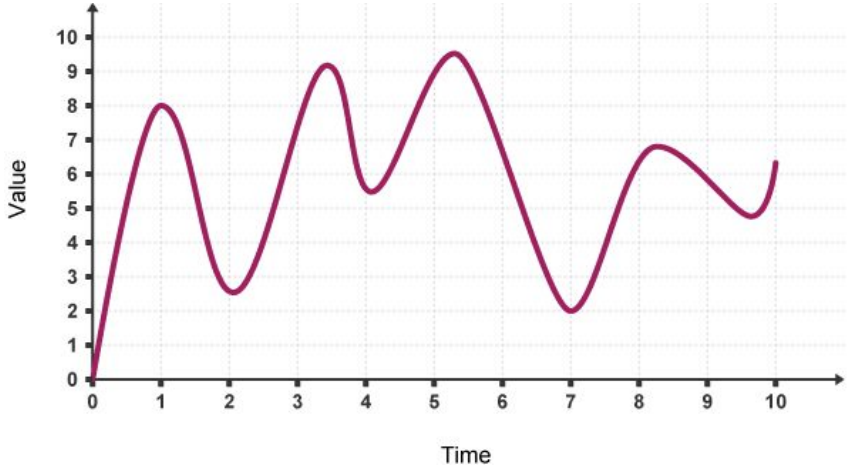
- Color – a binary string
  - Color depth: the number of **bits per pixel** on a computer monitor to represent a specific color
    - 1-bit (2^1 or 2 colors)
    - 2-bit (2^2 or 4 colors)
    - 4-bit (2^4 or 16 colors)
    - 8-bit (2^8 or 256 colors)
    - ……
    - 32-bit (2^{32} or 4,294,967,296 colors) Today, most computers support at least 32-bit.
    - 48-bit (2^{48} or 281,474,976,710,656 colors)

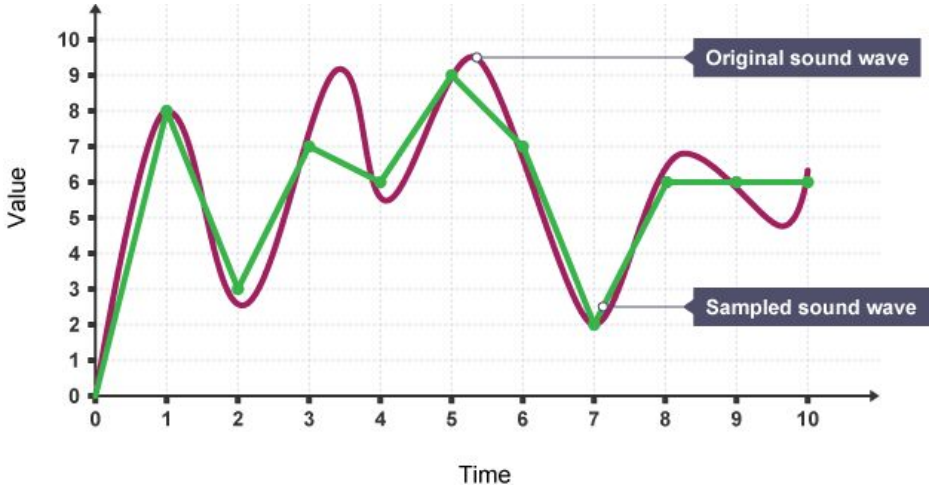
10	10	10	10	10	10
10	00	10	10	00	10
10	11	10	10	11	10
10	10	10	10	10	10
10	01	01	01	01	10
10	10	10	10	10	10



Information representation - Sound

- Sound
  - The converter samples a sound wave at regular time **intervals**, and ascribes a numerical value to each **amplitude**. The sample is converted into a binary code.





Information representation - Sound

- Sound

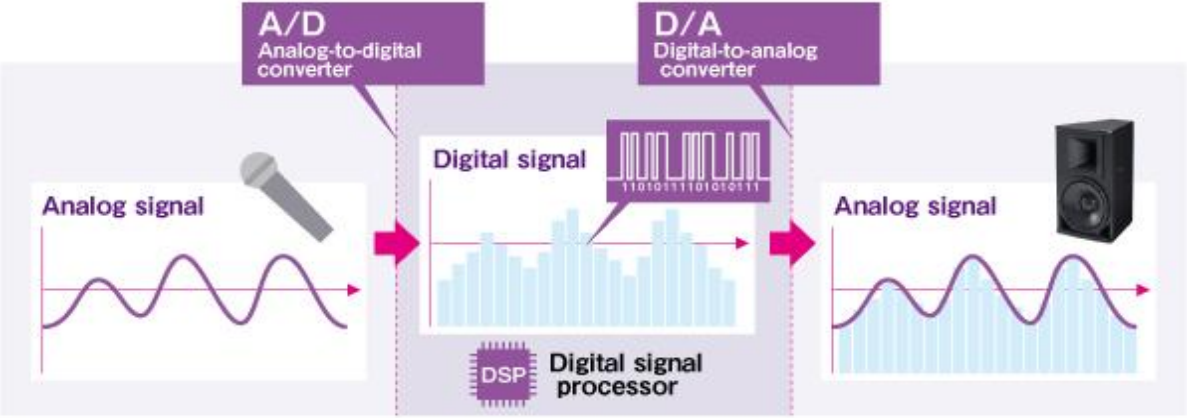
Time sample	1	2	3	4	5	6	7	8	9	10
Denary	8	3	7	6	9	7	2	6	6	6
Binary	1000	0011	0111	0110	1001	0111	0010	0100	0110	0110

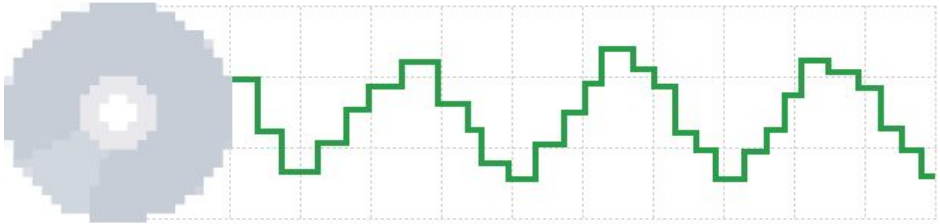
  - The frequency at which samples are collected is called the **sample rate**, and is measured in Hertz (Hz).

  - 1 Hz is one sample per second. Most CD-quality audio is sampled at 44,100 Hz.

Information representation - Sound

- Sound





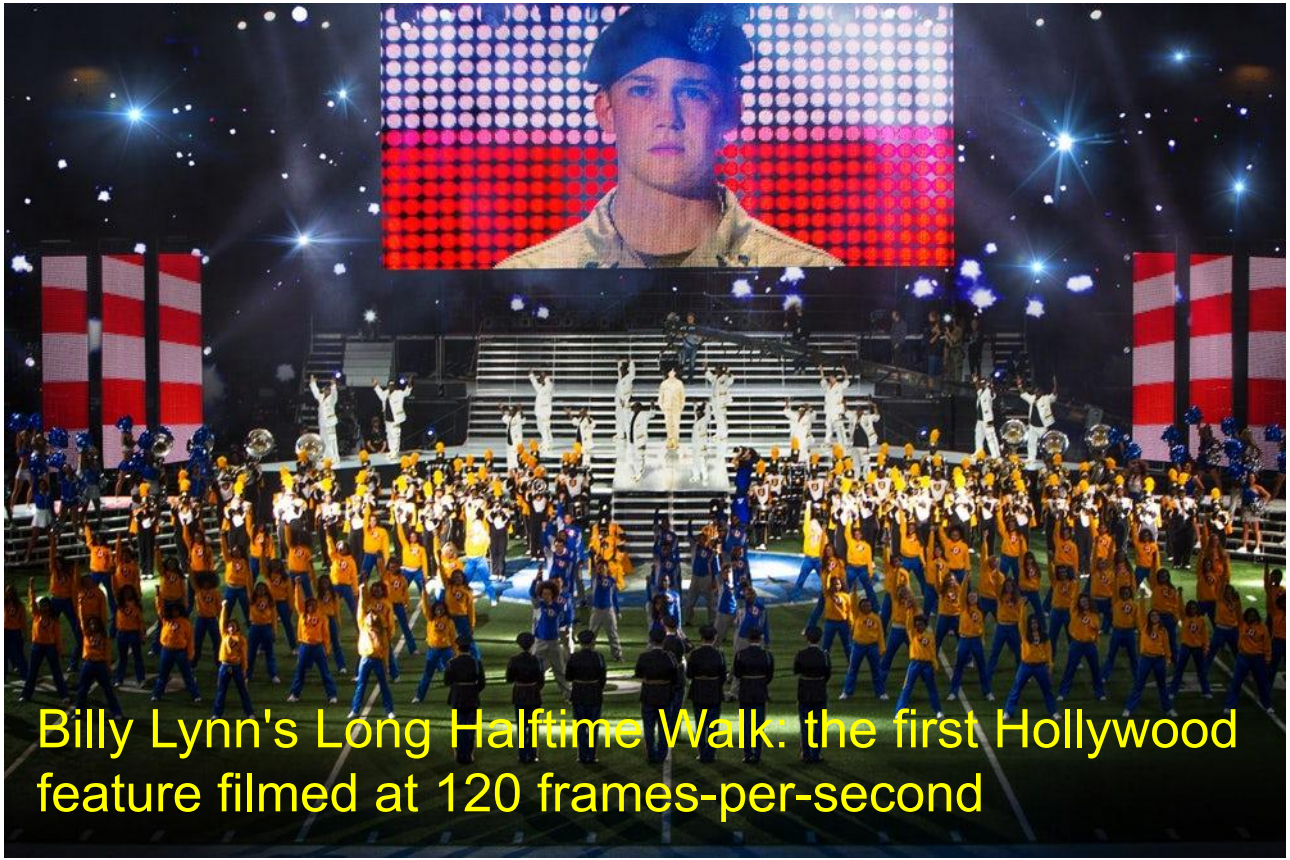
Information representation - Video

Information representation - Video

- Video
  - A digital video consists of a series of static images played consecutively at a high speed.
  - Usually around 24 frames per second but can be up to around 100 frames per second or more.

Information representation - Video

Information representation - Video



Information representation

- How much hard drive memory (in Byte) is required to store a movie?
- Standard definition
  - 720 x 480 pixels
  - Frames Per Second = 24
  - 2 hours long
  - Color Depth – assume 30 bit color (2^30 possible colors)
- 1 Byte = 8 bits
- 720 x 480 x 24 x 2 x 60 x 60 x 30 / 8 = $2.2*10^{11}$ Bytes

NYU
STERN SCHOOL OF BUSINESS

Information & Hardware

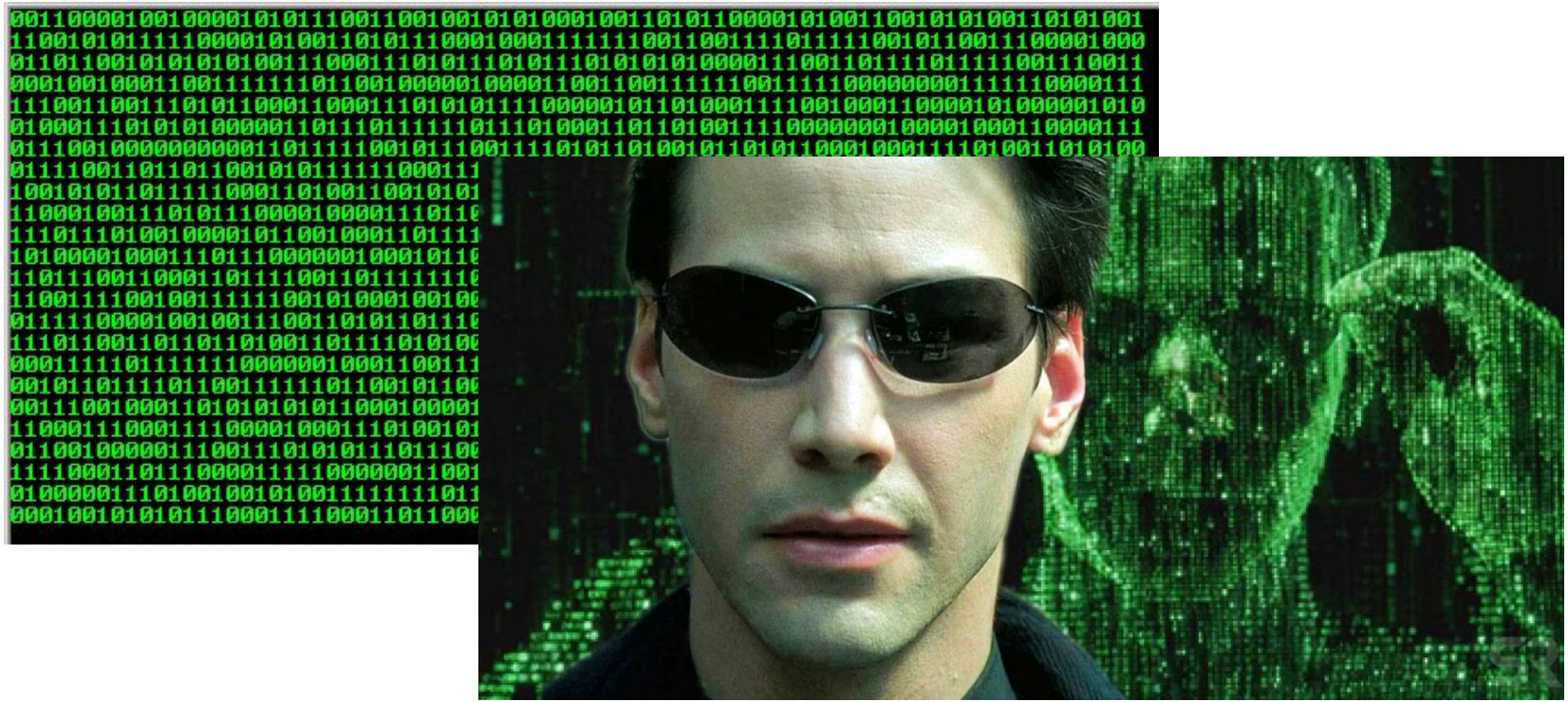
Objective:

Understand why computers use the binary system.

Understand why Moore’s law is accurate.

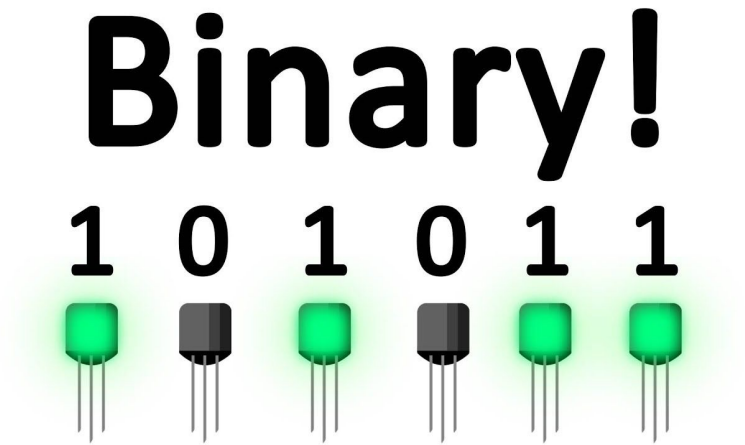
Information representation

- Everything needs to be converted into binary. Binary everything



Information ⇔ Hardware

- Why does computer represent information in binary codes?
  - Computers process and store data using tiny **on-off** switches. (transistors)
  - Sequences of on/off ⇔ strings of bits (**1's and 0's**).



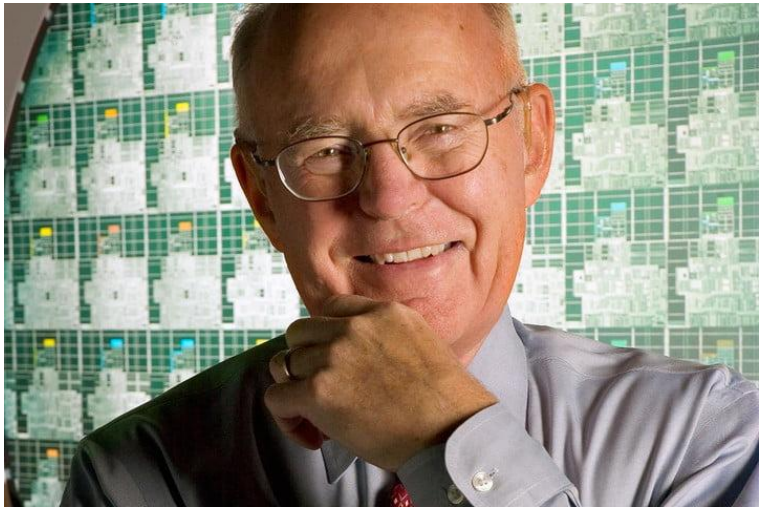
Information ⇔ Hardware

- Transistor
  - A small semiconductor electronic device, with at least two input contacts and one output, operating essentially like a switch (On and Off).
  - The **On and Off states** of a transistor reflect **1 and 0** used in binary data.

- Integrated circuit
  - A collection of **thousands or millions** of transistors placed on a small silicon chip.

- Foundation of computers
  - They are the fundamental building block of modern electronic computers.
  - **More transistors on a chip** means more powerful chips and computers.

Moore’s Law



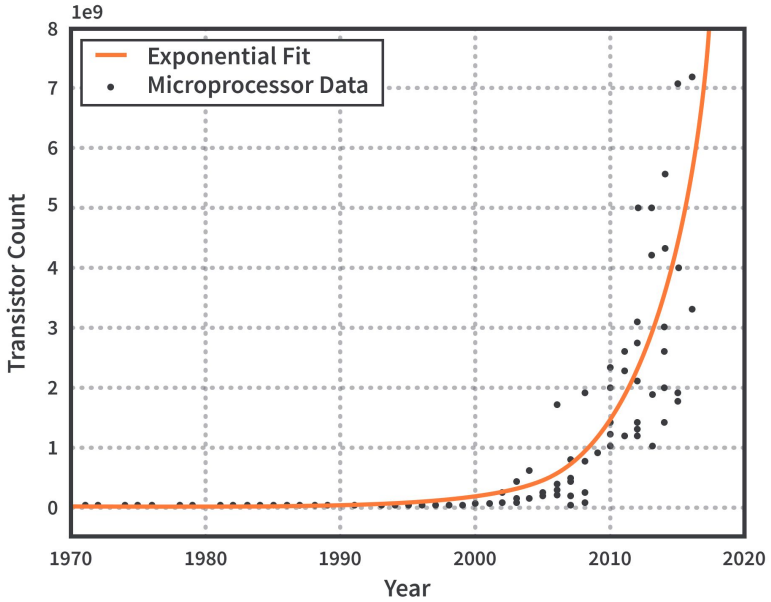
Gordon Moore, one of the founders of Intel, in 1965:

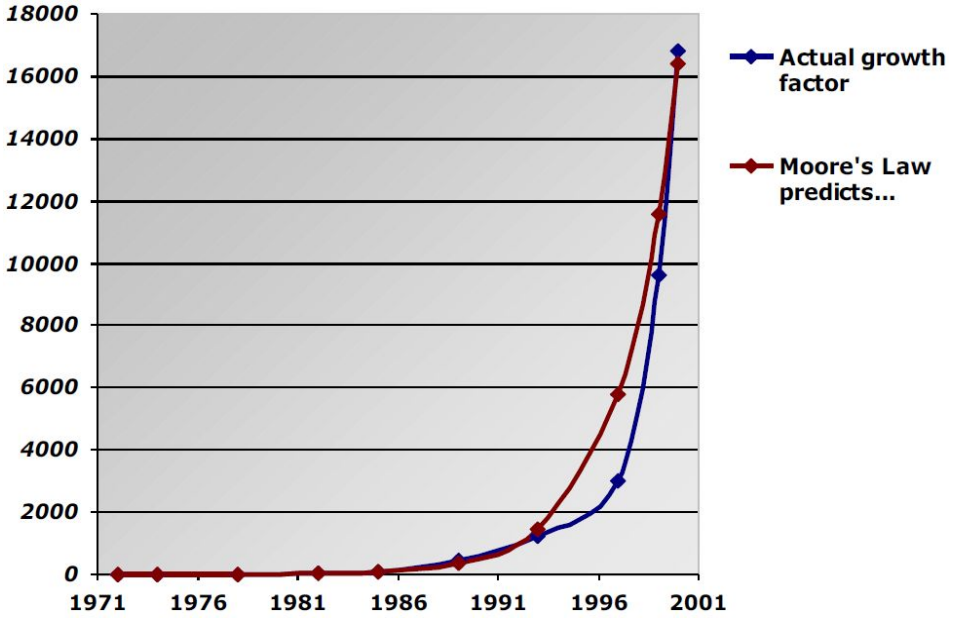
Observed that the number of transistors per square inch on integrated circuits had **doubled** every year.

Predicted that the number of transistors per square inch on integrated circuits **double** approximately every 18-24 months.

Current version of Moore’s law:
Doubles every 18 months.

Moore's Law

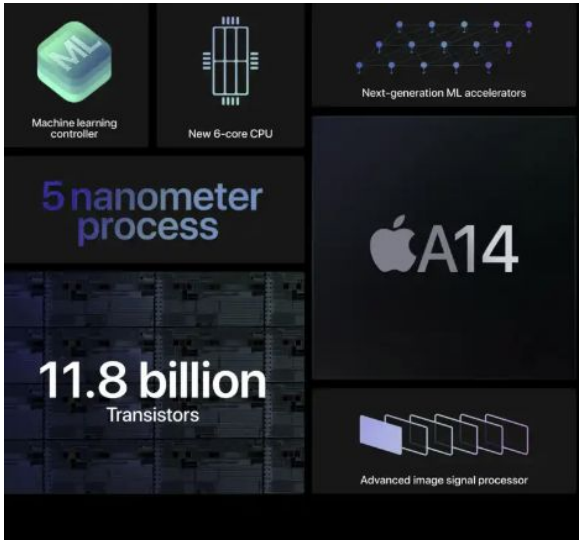

1e9
8
7
6
5
4
3
2
1
0
Transistor Count
Exponential Fit
Microprocessor Data
1970
1980
1990
2000
2010
2020
Year


18000
16000
14000
12000
10000
8000
6000
4000
2000
0
1971
1976
1981
1986
1991
1996
2001
Actual growth factor
Moore's Law predicts...

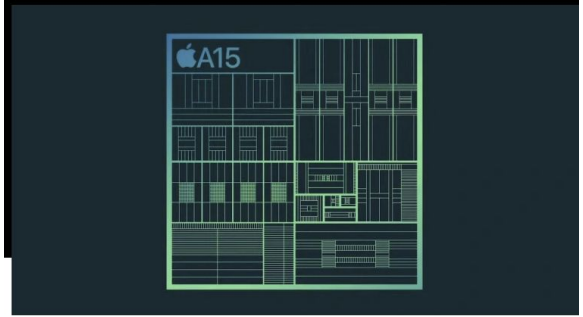
Moore's Law

Apple A13 For iPhone 11 Has 8.5 Billion Transistors, Quad-Core GPU

Ramish Zafar · 3y



Apple's iPhone 13 features A15 Bionic processor with 15B transistors



Moore’s Law

- Why has Moore’s Law been so accurate?

Moore's Law

- Why has Moore's Law been so accurate?

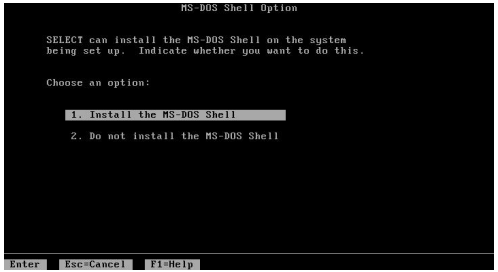
Moore's Law

- Why has Moore's Law been so accurate?

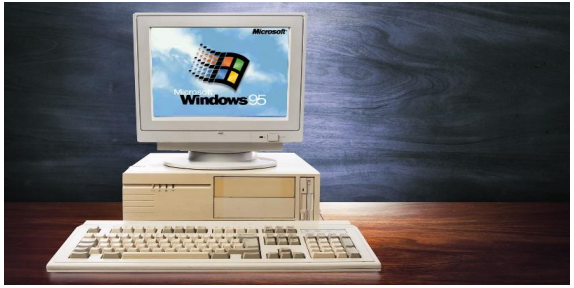
The future chose to follow the prediction!

- A key benefit of Moore's Law: **a coordination device** for the technology industry.
- "Common music to dance to" – MIT Technology Review
- (Developers of related technology, such as operating systems, software, or video games:) I know that in two years we can count on this amount of power and that I can develop this functionality—and if you're Intel you know that people are developing for that and that there's going to be a market for a new chip.

Industry Ecosystem



Industry Ecosystem



Industry Ecosystem

Limits of Moore's law

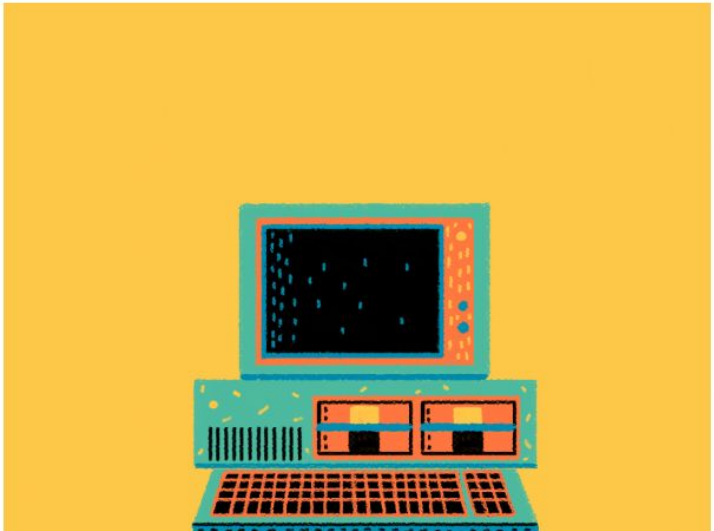
NYU STERN SCHOOL OF BUSINESS

Intelligent Machines

Moore's Law Is Dead. Now What?

Shrinking transistors have powered 50 years of advances in computing—but now other ways must be found to make computers more capable.

by Tom Simonite May 13, 2016



CES 2019: Moore's Law is dead, says Nvidia's CEO

The long-held notion that the processing power of computers increases exponentially every couple of years has hit its limit, according to Jensen Huang.

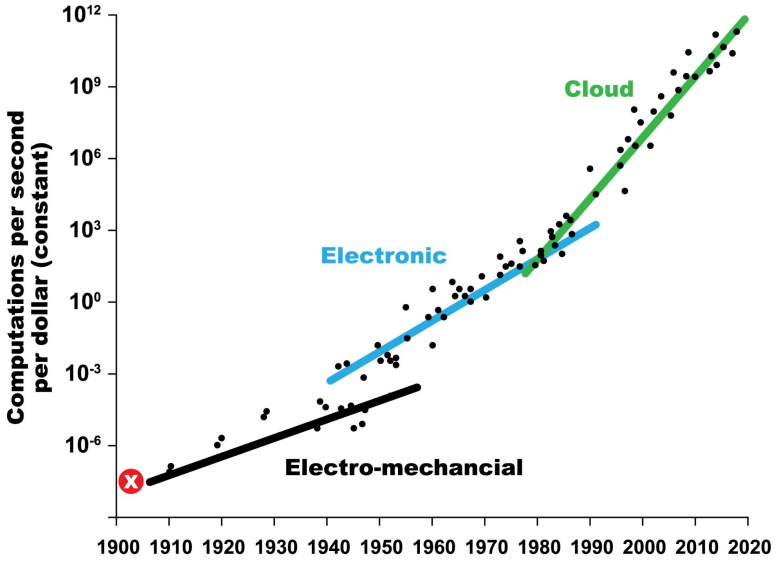
BY SHARA TIBKEN | JANUARY 9, 2019 11:46 AM PST

Moore's Law - Future

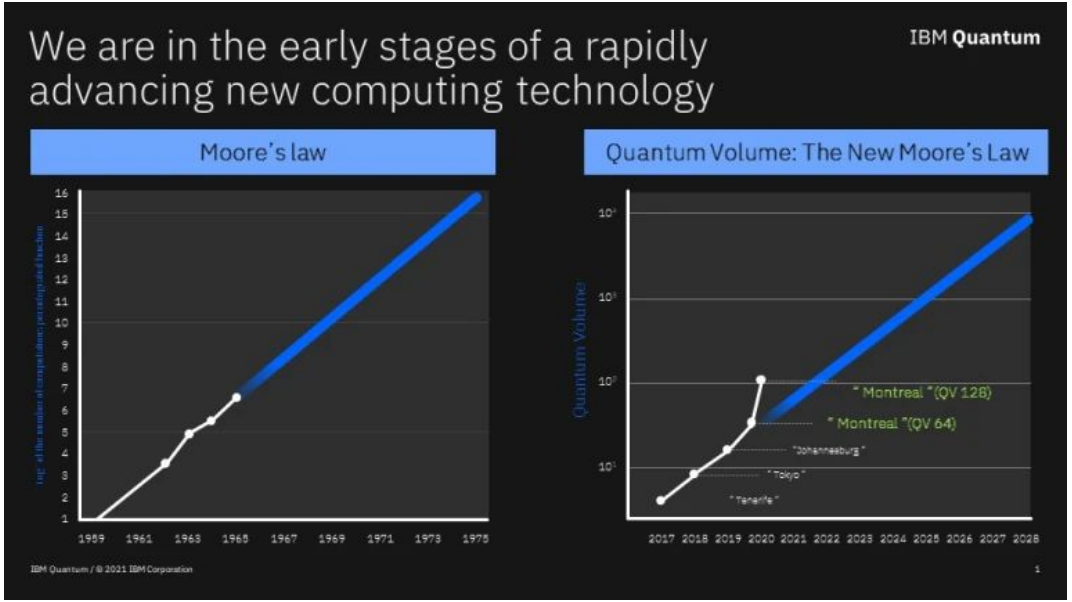
Stretch break

NYU STERN SCHOOL OF BUSINESS

Is it dead? Open question.

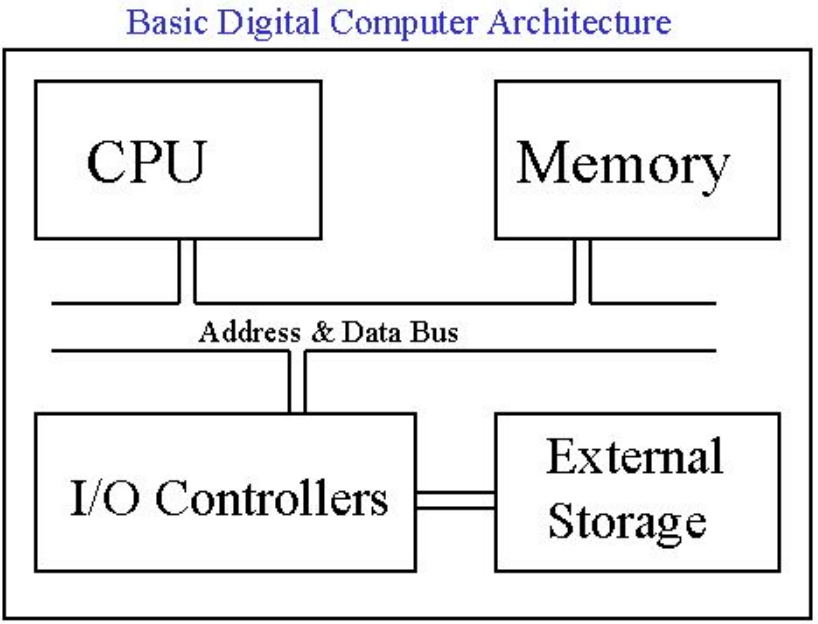


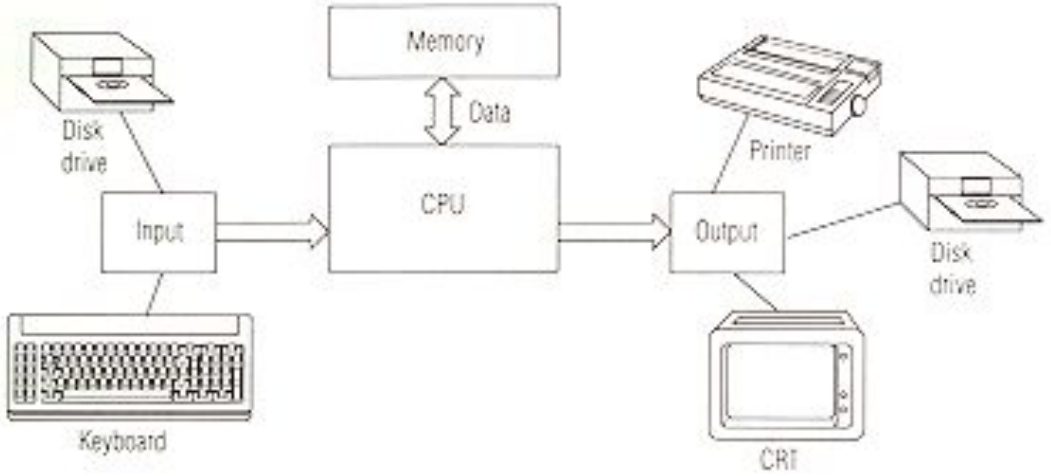
Computational power



Quantum computing?

Computer Architecture





NYU
STERN SCHOOL OF BUSINESS

Computer Architecture

Objective:

Get familiar with the components and their functions.

Computer Architecture - CPU

- Central processing unit (CPU)
  - The actual hardware that interprets and executes software instructions and **coordinates** how all the other hardware devices work together.
    - Intel: 286 → 386 → 486 → Pentium I,II,III,IV…
    - AMD Athlon
    - Sun SPARC…

Brain!!

Computer Architecture - RAM

- Random Access Memory (RAM)
  - The place for keeping the data and applications while the computer is running.

    Fast!

- Storage
  - A tool for storing information for use at a later time (floppy disk, CD, DVD, hard disk, tape).

    Slow!

I/O Devices

Input device

A tool used to capture information and commands from the environment.

Output device

A tool we use to see, hear, or otherwise receive the results of information-processing requests.

NYU
STERN SCHOOL OF BUSINESS

OS, Softwares

Objective:

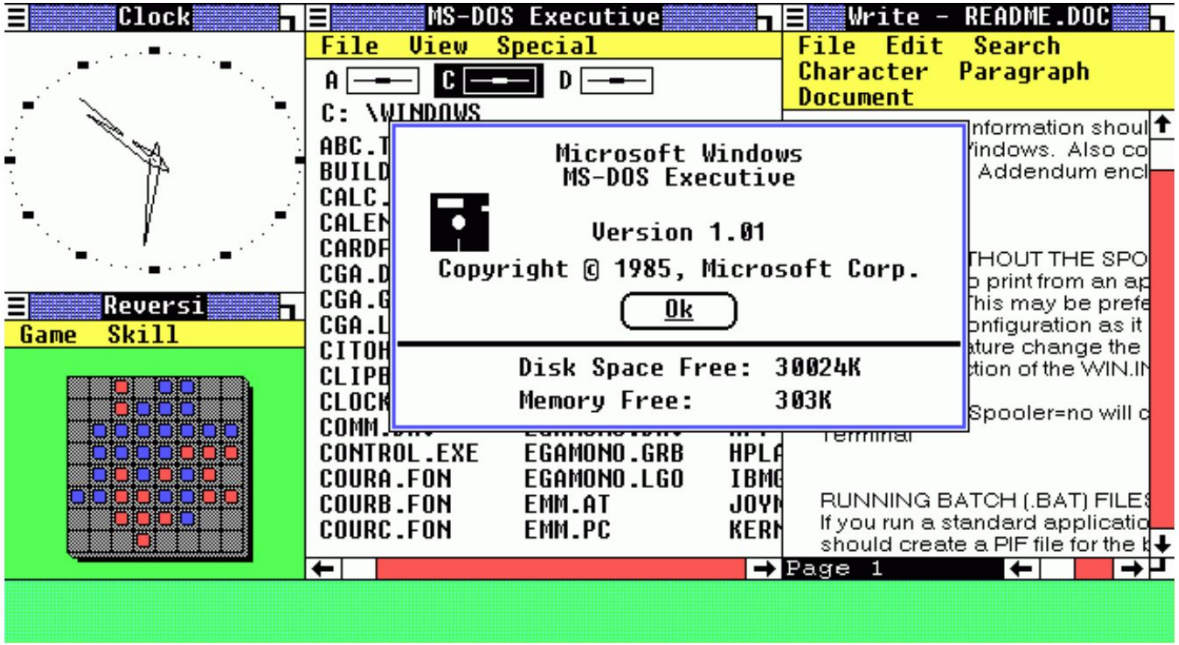
Get familiar with the components.

Operating Systems

- A set of programs that ‘manage and run your computer’
  - Manages memory, coordinates the running of application software, controls input/output hardware.
- Examples
  - MS-DOS, Windows 3.11, Win95, Win98, WinNT, Win2000, WinXP, Vista, Windows 7, Windows 8, Windows 10, …
  - Mac OS 8, Mac OS 9, Mac OS X, …
  - Linux, UNIX, Sun Solaris, IBM AIX, …
  - Pocket PC, Palm, iOS, Android, …

Operating Systems

Operating Systems



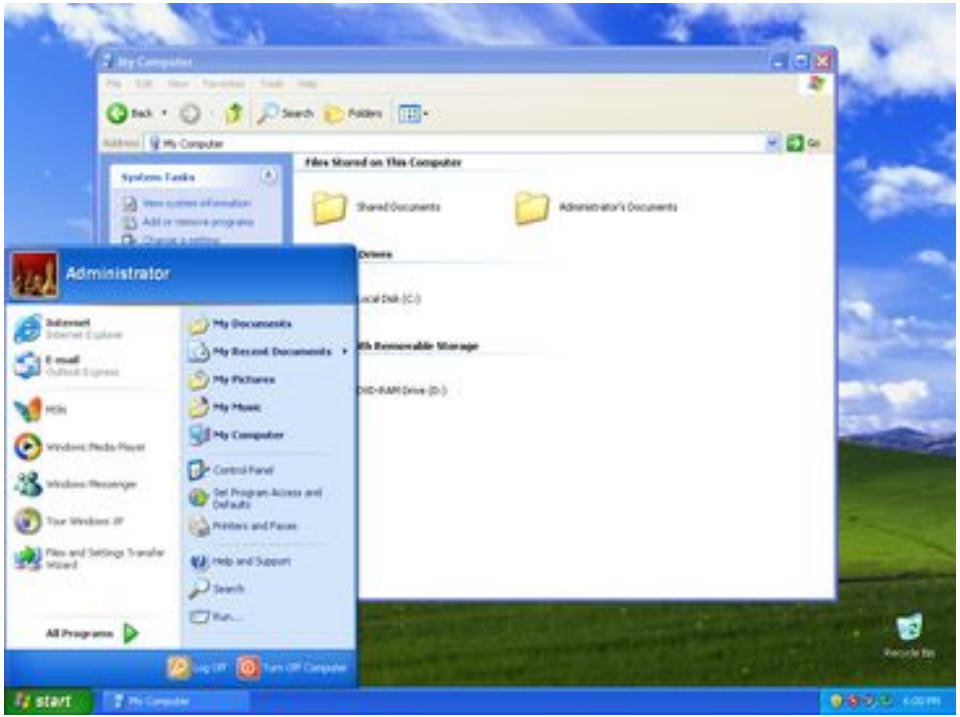
WIndows 1.0

Modern OS use a graphical user interface (GUI) (“gooey”) that lets you use a mouse to click icons, buttons, and menus displayed using graphics and text.

Operating Systems

WIndows 95 (1995)

Operating Systems

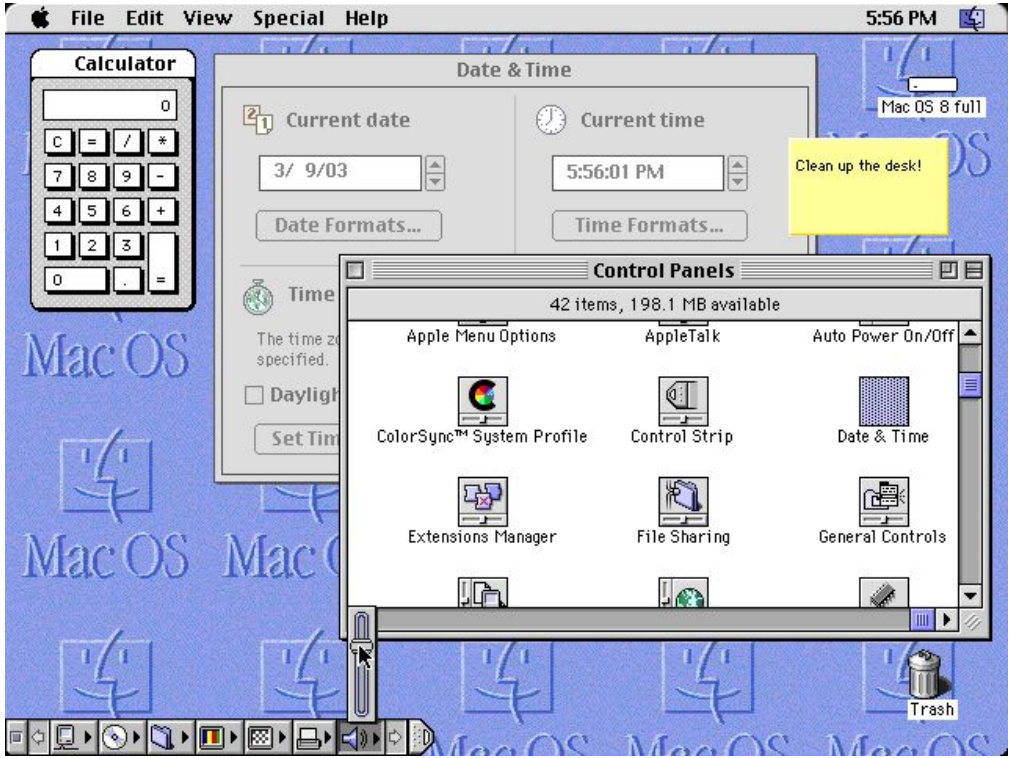


WIndows XP (2010)

Operating Systems

WIndows 10 (2015)

Operating Systems



MAC OS 8 (1997)

Operating Systems

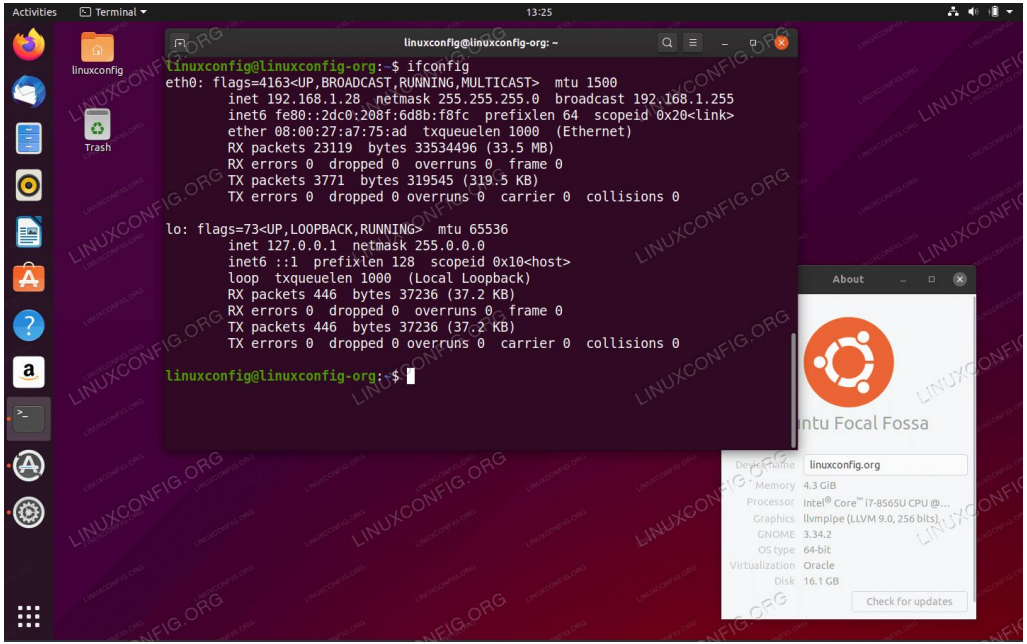
MAC OS 9 (1999)

Operating Systems

MAC OS 2007

Usually a new version of the O/S is compatible with older versions.

Operating Systems



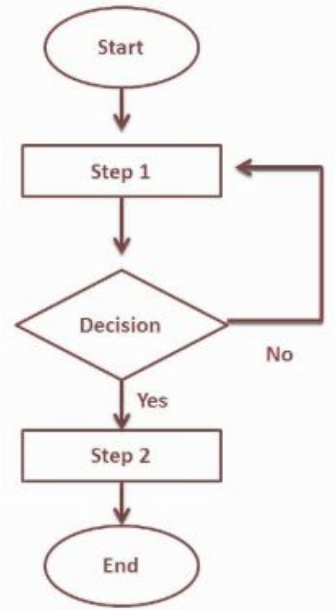
Linux

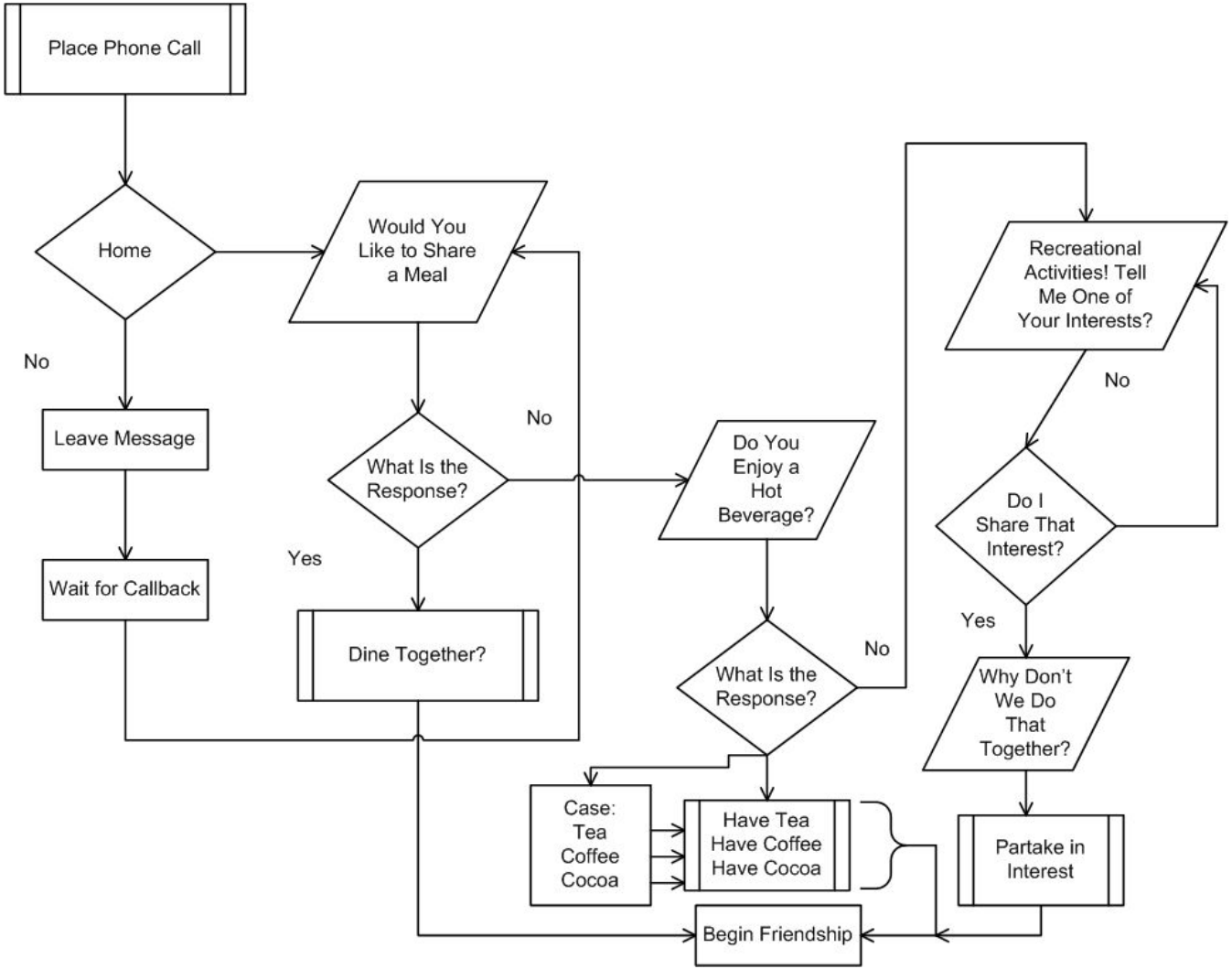
Open-source.

Contrast to the previous ‘proprietary softwares’.

Softwares - Algorithm

- What is an algorithm?
  - A **step-by-step blueprint** (**readable by humans**) for instructing a computer to complete a task.





[the algo]

Software - Program

- What is an algorithm?
  - A step-by-step blueprint (**readable by humans**) for instructing a computer to complete a task.
- What is a program?
  - All computers work by performing structured sequences of instructions (**that computers understand**) called programs.

Programming Languages

- Programming language
  - Special **languages used to create programs**: programming languages (BASIC; C; C++, Java, JavaScript, Ada, Python…).
  - Some more intuitive to learn (closer to the human language), but less efficient (more resource-consuming) for computer to run.
  - Some more efficient for computers, but more difficult to memorize and interpret for human programmers.

Programming Languages

NYU STERN SCHOOL OF BUSINESS

TECH • TECHNOLOGIZER

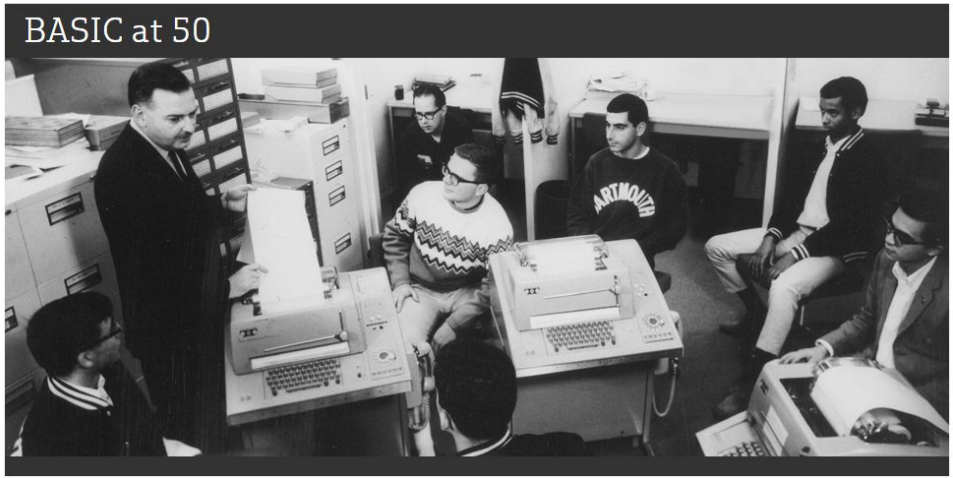
TIME

Fifty Years of BASIC, the Programming Language That Made Computers Personal

DARTMOUTH

Dartmouth Home | Search | Index

BASIC at 50



John Kemeny shows off his vanity license plate in 1967 Adrian N. Bouchard / Dartmouth College

Programming Languages

- Examples : BASIC

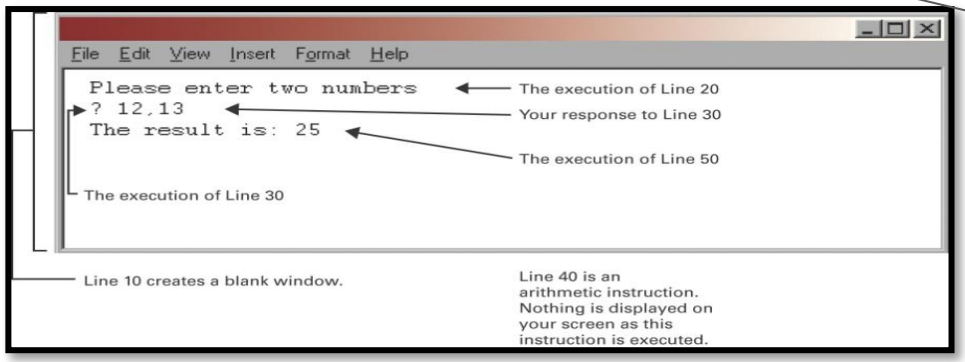
```
10    CLS
20    PRINT "Please enter two numbers"
30    INPUT A, B
40    C = A + B
50    PRINT "The result is:"; C
60    STOP
```

1. Ask for two numbers

2. Add the two numbers

** The equal sign refers to assignment of values.

3. Display the results



Programming Languages

- Examples : C

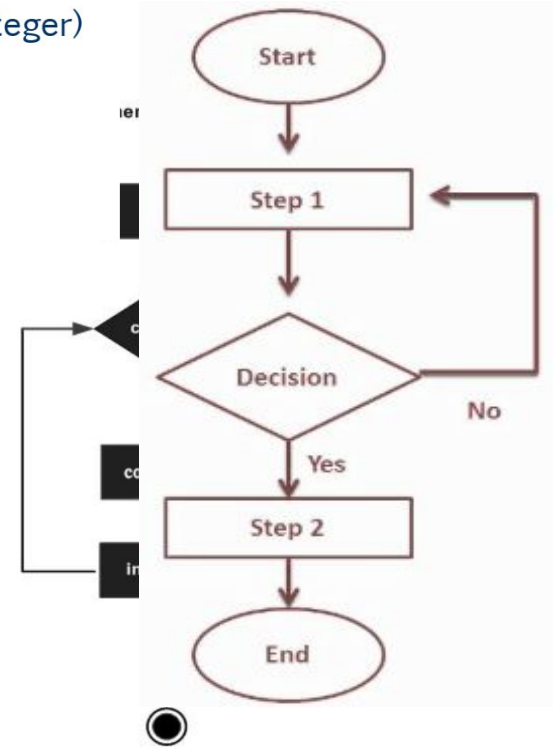
```c
#include<stdio.h>

main()
{
   int n, c = 2;  Declaring variables n, c; Type of the variable: int (integer)

   printf("Enter a number to check if it is prime\n");
   scanf("%d",&n);

   for ( c = 2 ; c <= n - 1 ; c++ ) Declaring loop (for)
   {Initial state    Condition    Increment
      if ( n%c == 0 )
      {     Break condition: n can be evenly divided by c
         printf("%d is not prime.\n", n);
         break;
      }
   }
   if ( c == n ) If we have tried n times.
      printf("%d is prime.\n", n);

   return 0;
}
```



Programming Languages

- Examples : Python

script.py IPython Shell

```
1   # Python program to check if the input number is prime or not
2
3   num = 407
4
5   # take input from the user
6   # num = int(input("Enter a number: "))
7
8   # prime numbers are greater than 1
9   if num > 1:
10      # check for factors
11      for i in range(2,num):
12          if (num % i) == 0:
13              print(num,"is not a prime number")
14              print(i,"times",num//i,"is",num)
15              break
16      else:
17          print(num,"is a prime number")
18
19  # if input number is less than
20  # or equal to 1, it is not prime
21  else:
22      print(num,"is not a prime number")
```

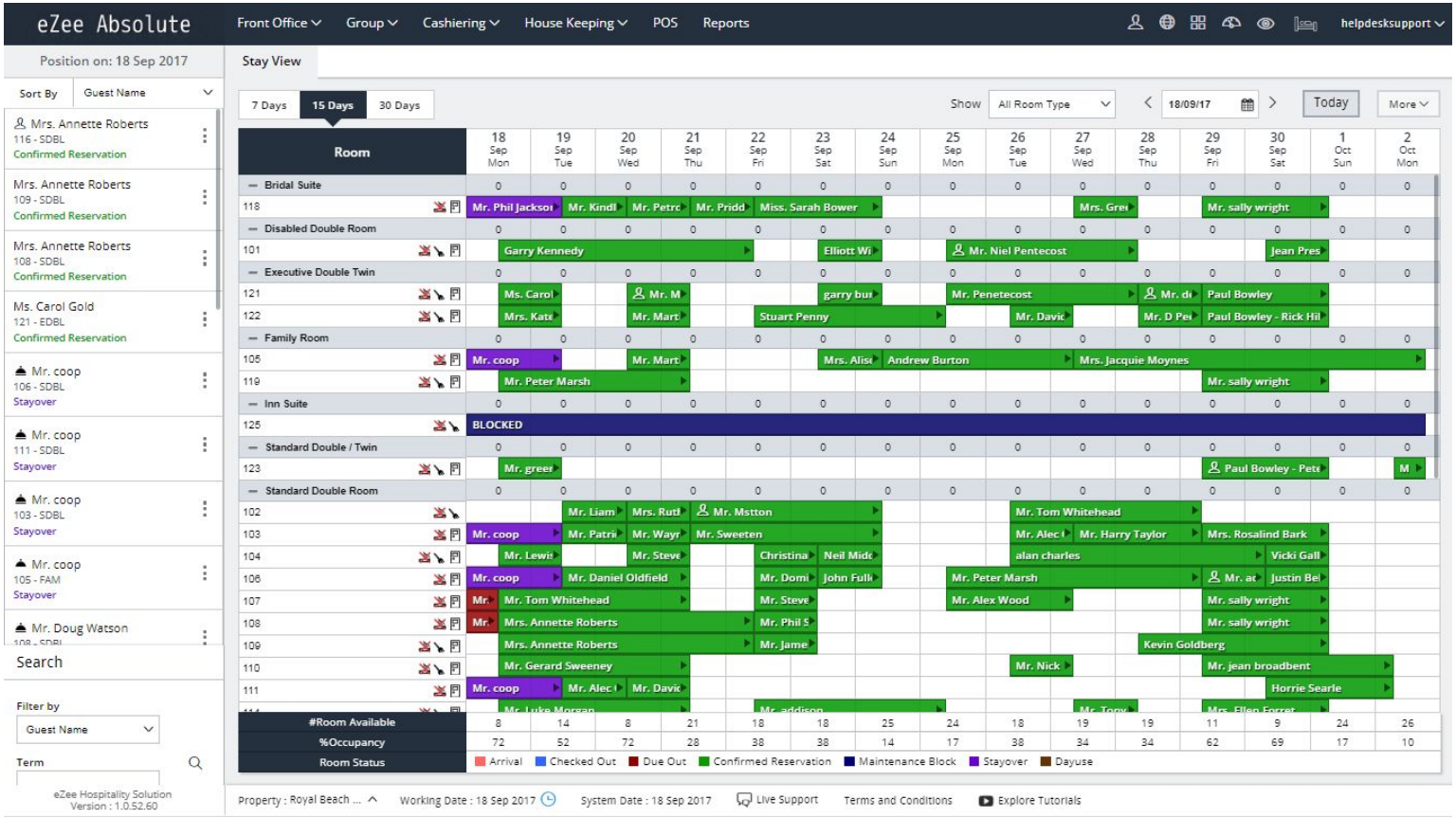
Softwares

- Software
  - A set of **related programs** that fulfill specific information processing and/or business needs.
  - Application software: software for specific personal and commercial activities.

Application Softwares

- Categories
  - Vertical:

    can be used for a specific industry
    - Studio (TV/Film) project management software
    - Hospitality reservation management software
    - Patient scheduling software…



Hotel reservation software

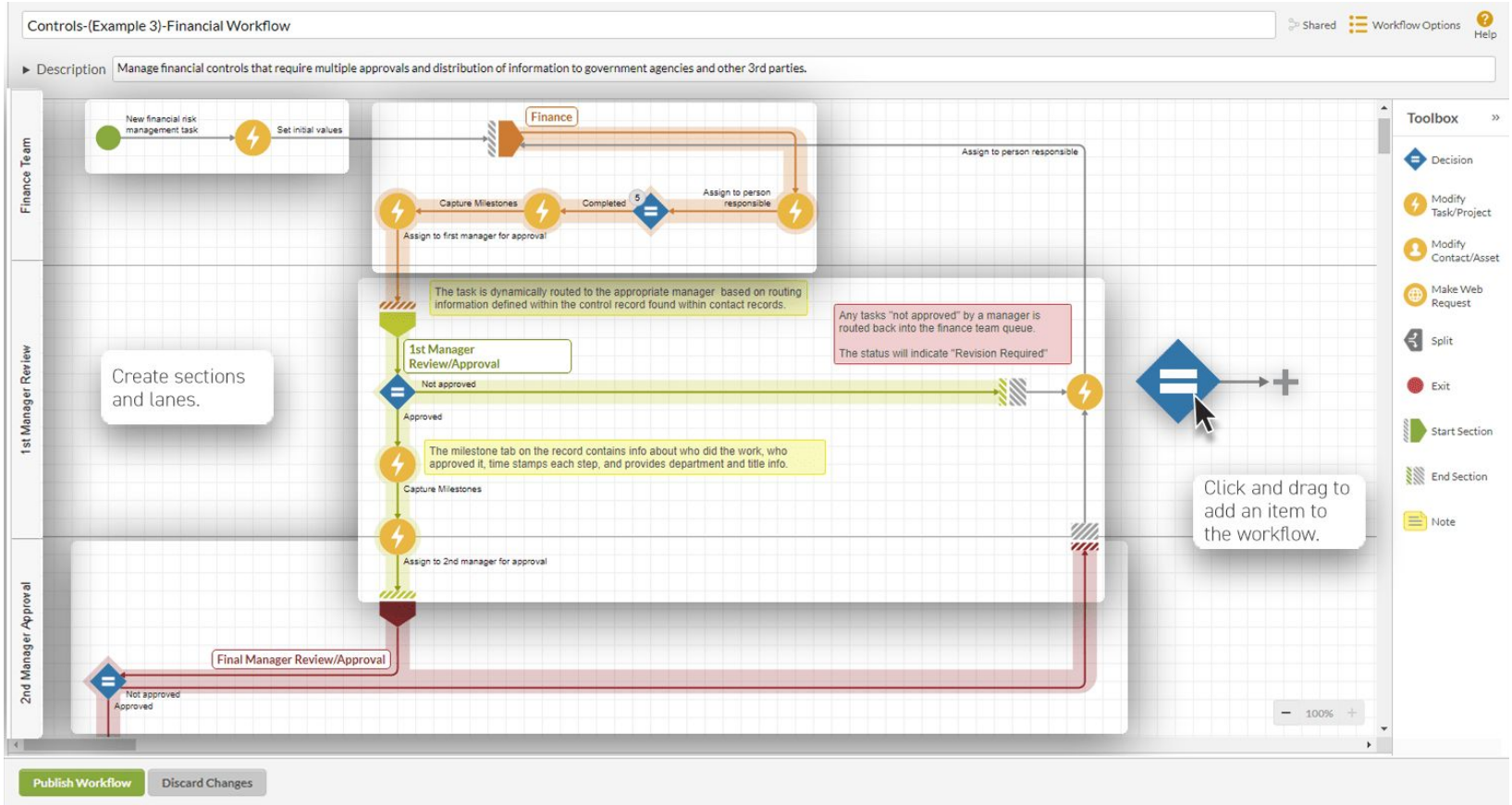
Application Softwares

- Categories
  - Vertical:

    can be used for a specific industry
    - Studio (TV/Film) project management software
    - Hospitality reservation management software
    - Patient scheduling software…
  - Horizontal:

    for specific business functions, can be used in multiple verticals
    - Business process management software
    - Customer relationship management software
    - Inventory management software…

NYU STERN SCHOOL OF BUSINESS



Business Process Management software [demo]