華南理工大學

South China University of Technology

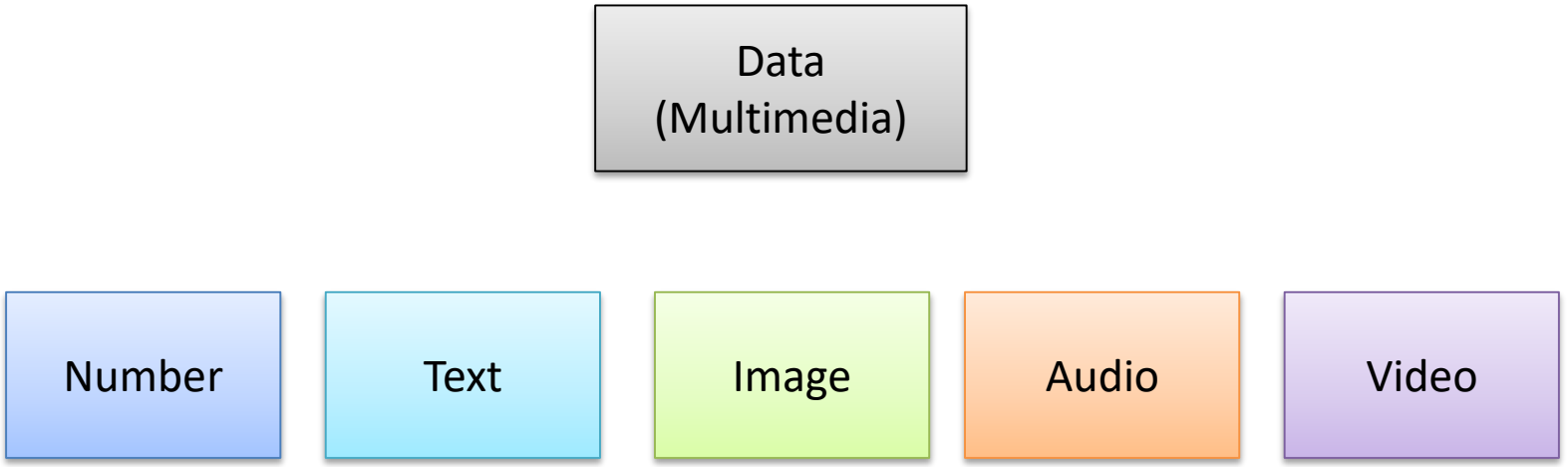
Intro to Computer Science and Software Engineering

Data Representation

Dr Yubei Lin
yupilin@scut.edu.cn
School of Software Engineering

Data types

- Computer needs to process different types of data
- **Multimedia**: referred to information contains numbers, text, images, audio and video.



Data inside the computer

- For sake of efficiency, a **uniform** representation of data of all types.
- **Bit Pattern**, is the widely used solution in electronic computers.
- That is, all data needs to encoding into bit pattern before storing inside computers.

Bit

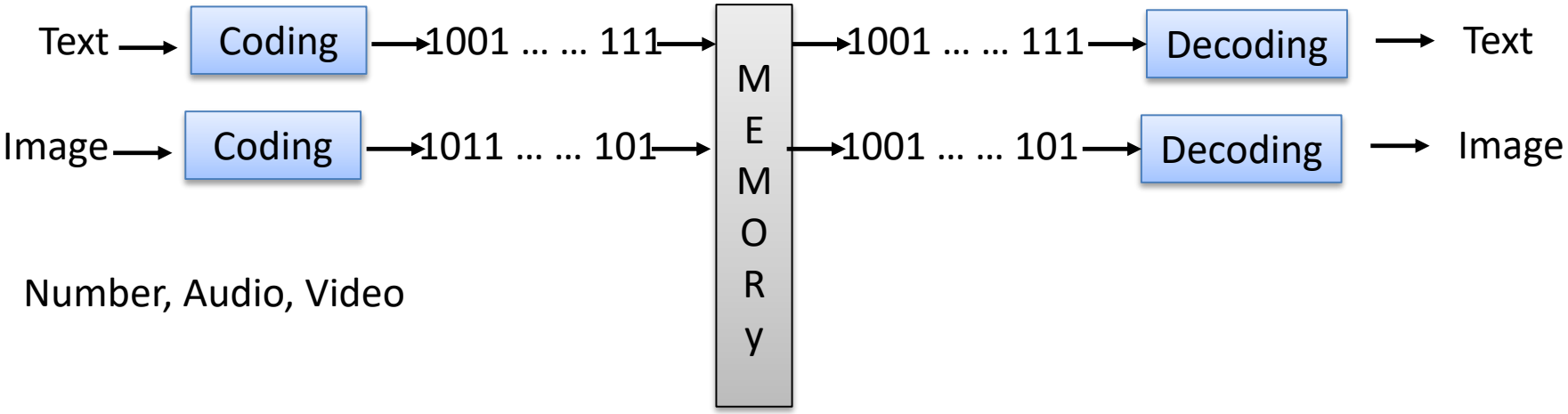
- A bit (binary digit) is **the smallest unit** of data can be stored in a computer; it's either **0 or 1**.
- Bits can be implemented in many forms, depending on the underlying two-state devices.
- In most modern computing devices, a bit is usually represented by
  - the electrical state of a **flip-flop circuit (触发器)**.
- Interest in Bit's Physical representation @ https://en.wikipedia.org/wiki/Bit

Bit Pattern

- A bit pattern is a sequence (or string) of bits.

0	1	0	1	1	0	0	1	0	1

- The above bit pattern is a number, text, or image?
  - The computer can only process a bit pattern correctly if it knows the type of the data.
  - How computer know the type? By programs!



Measuring the size

- Bit: the smallest unit, value of either 0 or 1
- Byte: a bit pattern of 8 bits
- Kilobyte: 2^{10} (1024) bytes
- Megabyte: $2^{10} * 2^{10}$ (1024^2) bytes
- **Gigabyte**: $2^{10} * 2^{10} * 2^{10}$ (1024^3) bytes
- **Terabyte**: $2^{10} * 2^{10} * 2^{10} * 2^{10}$ (1024^4) bytes
- Petabyte, Exabyte, Zettabyte, Yottabyte

Representing Text

- Text: a sequence of symbols (in a language)
  - “BYTE” in English consists of ‘B’, ‘Y’, ‘T’ and ‘E’ symbols.
- Each symbol should be represented by an unique bit pattern.
- How about the length of each bit pattern?
  - Depends on the number of symbols of the set.
  - **N : $\log_2 N$**, the relationship is logarithmic!

Codes, or character-encoding schemes

- Code: defines rules for mapping a set of bit patterns to text symbols.
- There exists several codes, we’re going to look at ASCII, and Unicode briefly.
- GB2312是中华人民共和国国家标准简体中文字符集，全称《信息交换用汉字编码字符集·基本集》，由中国国家标准总局发布，1981年5月1日实施。

ASCII Code

- ASCII: American Standard Code for Information Interchange, by ANSI
  - ASCII uses a **7-bit pattern** ranging from 0000000 to 1111111.
  - represent text in computers, communications equipment, and other devices that use text.
  - Most modern codes are based on ASCII, though they support many additional characters.
  - ASCII was the most common character encoding on the World Wide Web until December 2007, when it was surpassed by **UTF-8**, which includes ASCII as a subset.

ASCII Code

USASCII code chart

$b_7 b_6 b_5$ →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits ↓ b_4 b_3 b_2 b_1	Column → 0	1	2	3	4	5	6	7
0 0 0 0	0 NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1 SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2 STX	DC2	"	2	B	R	b	r
0 0 1 1	3 ETX	DC3	#	3	C	S	c	s
0 1 0 0	4 EOT	DC4	$	4	D	T	d	t
0 1 0 1	5 ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6 ACK	SYN	&	6	F	V	f	v
0 1 1 1	7 BEL	ETB	'	7	G	W	g	w
1 0 0 0	8 BS	CAN	(	8	H	X	h	x
1 0 0 1	9 HT	EM	)	9	I	Y	i	y
1 0 1 0	10 LF	SUB	*	:	J	Z	j	z
1 0 1 1	11 VT	ESC	+	;	K	[	k	{
1 1 0 0	12 FF	FS	,	<	L	\	l	|
1 1 0 1	13 CR	GS	-	=	M	]	m	}
1 1 1 0	14 SO	RS	.	>	N	^	n	~
1 1 1 1	15 SI	US	/	?	O	_	o	DEL

1: 011 0001 (49); A: 100 0001 (65); a: 110 0001 (97)

ASCII Code

- 7位编码，128种符号：包括所有的大写和小写英文字母、数字0到9、各种标点符号等可显示符号（95个），以及常用的特殊控制符号（33个）。

1. 0～9的代码小于A～Z的代码，A～Z的代码小于a～z的代码
2. 数字0到9的ASCII代码依次递增1，数字0的代码为30H
3. 字母A到Z的ASCII代码依次递增1，字母A的代码为41H
4. 字母a到z的ASCII代码依次递增1，字母a的代码为61H
5. 同一个字母的大写字母ASCII要比小写字母ASCII小32或20H
6. 空格符、回车符、换行符的ASCII分别为20H，0DH，0AH

Unicode

- **Unicode（UCS）编码**，也称为**统一码**、万国码或单一码，是一种在计算机上广泛使用的**多字节字符编码**。
- 它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。
- Unicode有**UCS-2**和**UCS-4**两种编码标准。

UCS-4

- UCS-4字符集采用四维编码空间，整个空间有**128个组**，每个组再分为**256个平面**，每个平面有**256行**，每行有**256个列**。
- 每个符号的UCS-4编码有4个字节，分别表示代码这个符号的代码点在该四维空间所在的**组、平面、行和列**。
- 书写Unicode代码点时使用十六进制数表示，并且在数字前加上前缀“U+”

BMP与UCS-2

- 第0组的第0个平面被称作**BMP**。该平面的**65536**个代码点编码了常用的各国文字字母、标点符号、图形符号等，基本满足各种语言的使用。
- 将UCS-4的BMP平面代码点去掉前面的两个零字节就得到了**UCS-2**。
- UCS-2的两个字节分别表示了代码点在BMP平面的行和列

Unicode的具体实现

- UTF规范包括UTF-8、UTF-16和UTF-32三种实现方式。
- **UTF-8**以字节为单位对Unicode进行编码，对不同范围的字符使用不同长度的编码。
- **UTF-16**编码以16位无符号整数为单位。
- **UTF-32**编码以32位无符号整数为单位。
- 目前UTF-8和UTF-16被广泛使用，而由于UTF-32太浪费存储空间而很少被使用。

UTF-8

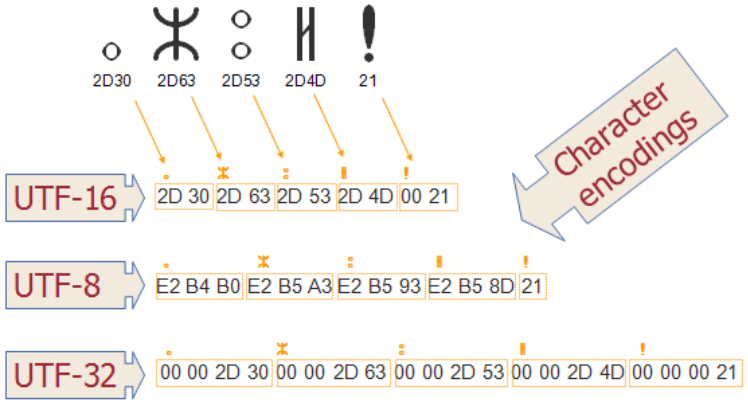
Unicode (16)	UTF-8 (2)
00000H~00007FH	0xxxxxxx
000080H~0007FFH	110xxxxx 10xxxxxx
000800H~00FFFFH	1110xxxx 10xxxxxx 10xxxxxx
01000H~10FFFFH	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

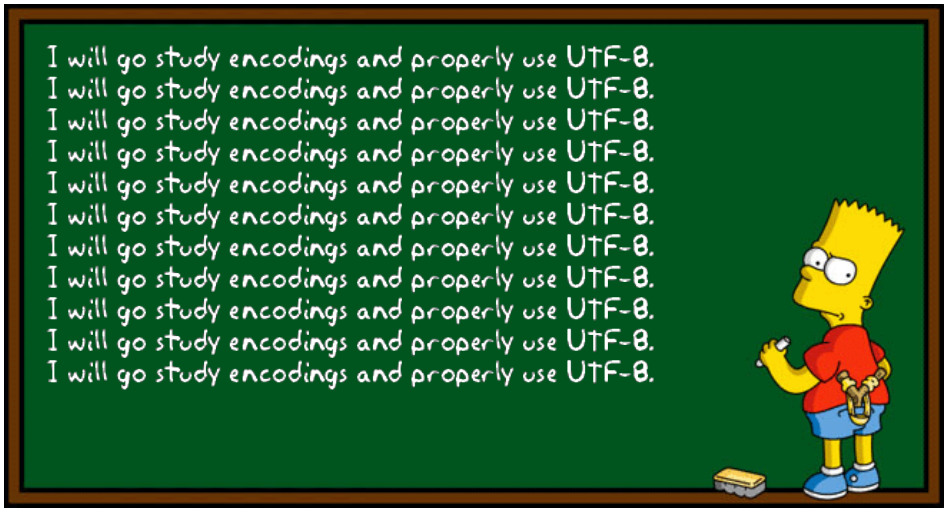
- 例“汉”的Unicode编码为U+6C49。
  - 在000800H~00FFFFH之间
  - 6C49的二进制0110 1100 0100 1001
  - 11100110 10110001 10001001
  - 即“汉”的UTF-8编码为E6B189

所有图片均来自网络

Unicode and UTF-8

- Unicode uses 16 bits and can represent up to 65536 (2^{16}) symbols.
- Different sections of the code are allocated to symbols from different languages in the world.
- UTF-8: 8-bit Unicode Transformation Format

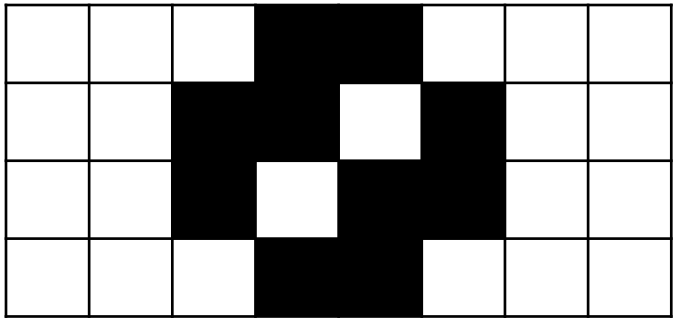




Representing Image

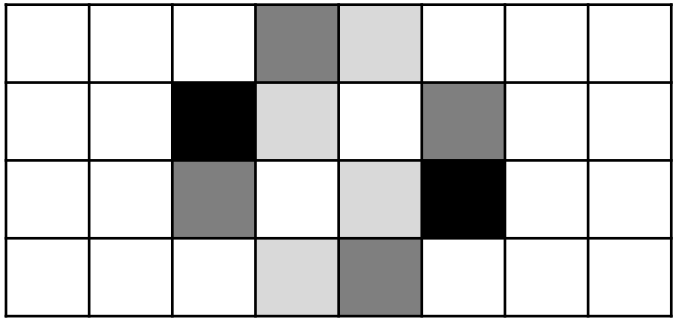
- Two methods: Bitmap and Vector graphic
- Bitmap Graphic (位图): a matrix of pixels

For black-white bitmap: 0 represents black pixels, 1 represents white pixels.



1	1	1	0	0	1	1	1
1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	1

For gray-scale bitmap: 00 - black, 01 – dark gray, 10 – light gray, 11 – white.



11	11	11	01	10	11	11	11
11	11	00	10	11	01	11	11
11	11	01	11	10	0	11	11
11	11	11	10	01	11	11	11

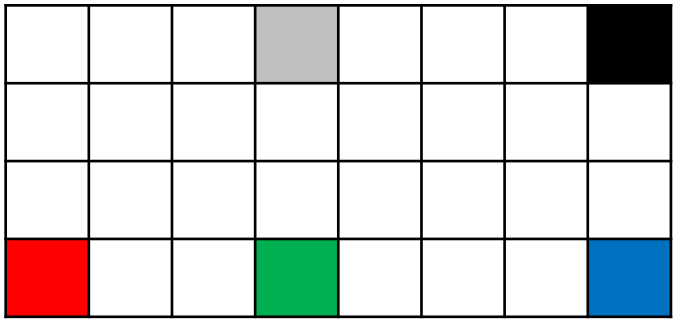
Representing Image

For color bitmap:
each colored pixel is decomposed into three primary colors:
Red, Green and Blue (RGB);
A 8-bit pattern is used to represent the intensity of each color.

R: 11111111
G: 11111111
B: 11111111

R: 01100000
G: 01100000
B: 01100000

R: 00000000
G: 00000000
B: 00000000



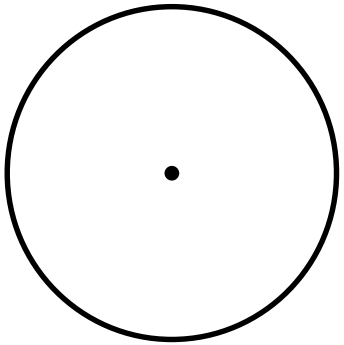
R: 11111111
G: 00000000
B: 00000000

R: 00000000
G: 11111111
B: 00000000

R: 00000000
G: 00000000
B: 11111111

Representing Image

- Vector graphic (矢量图)
  - The image is decomposed into a combination of curves and lines.
  - Each curve or line is represented as a mathematical formula.
  - The formula is encoded and stored in computers.



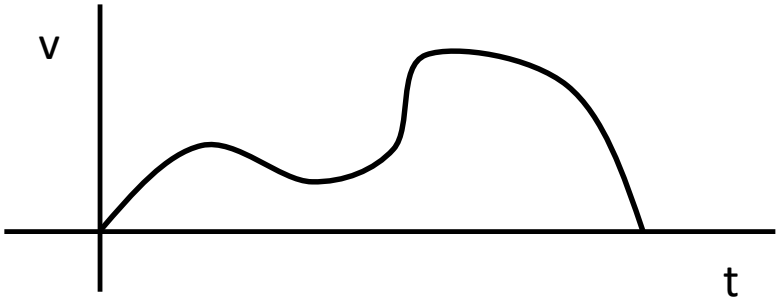
A circle can be described by:
- The coordinates of its center
- The length of its radius

Representing Video

- Video is a representation of images (called frames) in time.
- Video is normally encoded and then compressed before storing.

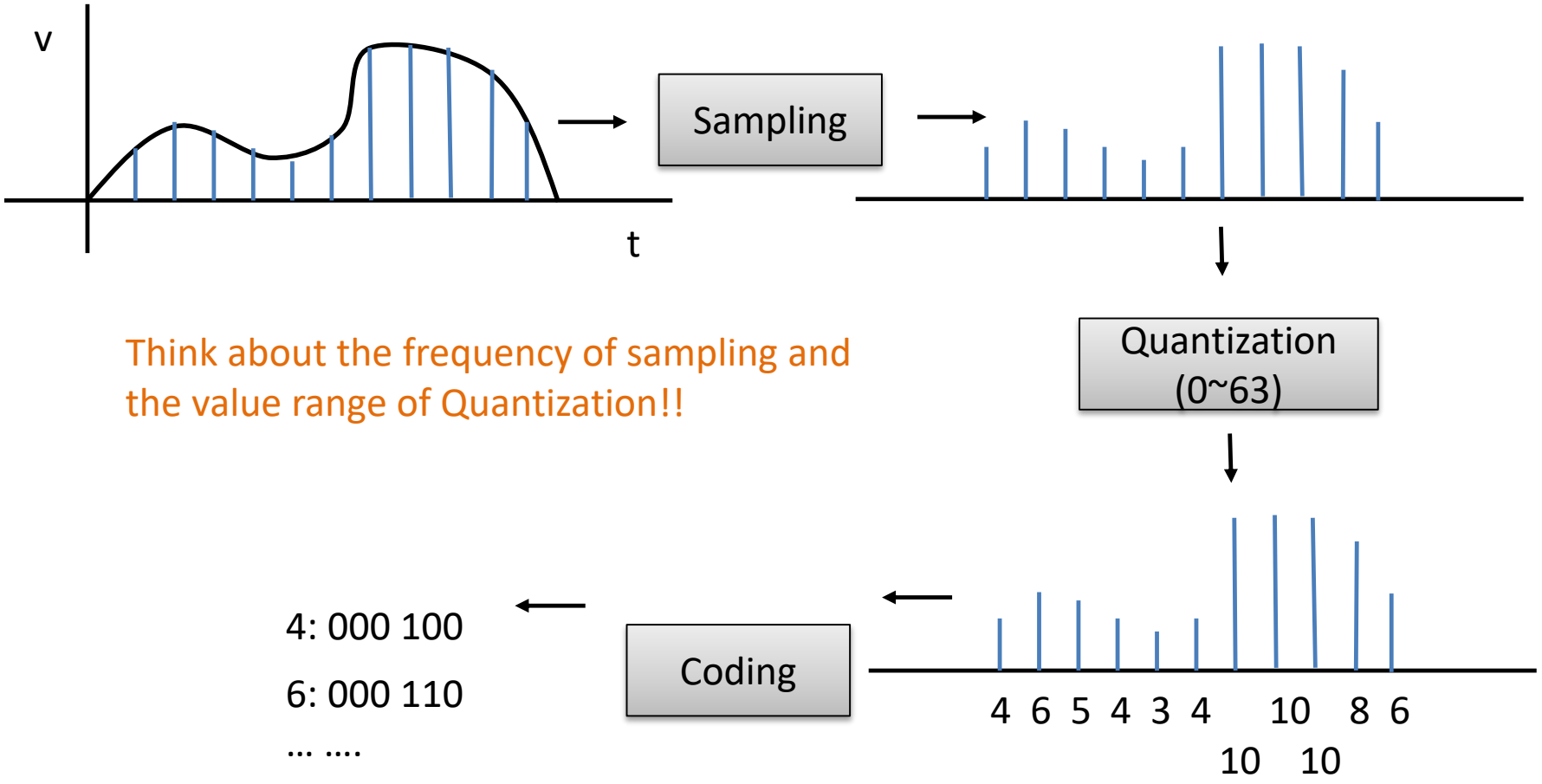
Representing Audio

- Audio is by nature analog (continuous);



- The idea is to convert audio to digital (discrete) data, and use bit patterns to store them.
- The process involves **Sampling**, **Quantization** and **Coding** steps.

Representing Audio



Think about the frequency of sampling and the value range of Quantization!!

Hexadecimal Notation

- Hexadecimal notation is based on 16.
- A 4-bit pattern can be represented by a hexadecimal digit, and vice versa.

0	1	2	3	4	5	6	7
0000	0001	0010	0011	0100	0101	0110	0111
8	9	A	B	C	D	E	F
1000	1001	1010	1011	1100	1101	1110	1111

- Commonly, add lowercase ‘x’ before the digits to show that the representation is in hexadecimal.
  - E.g. xCE2 (1100 1110 0010)

Octal Notation

- Octal notation is based on 8.
- A 3-bit pattern can be represented by a octal digit, and vice versa.

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

- Commonly, add lowercase ‘o’ before the digits to show that the representation is in hexadecimal.
  - E.g. o562 (101 110 010)

Homework

- 深入学习：UTF-8，WWW中广泛使用
- 深入学习汉字编码，如GB2312编码
- Edsger Dijkstra：就像望远镜之于天文学，计算机科学不仅仅是计算机。
- 用UTF-8 和GB2312进行编码上述名言。
- 如“就”： \xE5\xB0\xB1 (UTF-8)

- 认真学习，不要直接网上查询！

所有图片均来自网络