编码 | 进制

01

计算机的底层硬件实现就是用电路的开和闭两种状态来表示0和1两个数字的

在计算机上也能表示、存储和处理像文字、符号等等之类的字符，就必须将这些字符转换成二进制数字

EBCDIC码

英文字母不是连续排列的，中间出现多次断续，这带来了一些困扰和麻烦

ASCII码

一个字符占用一个字节也就是 8 个比特位

一共可以表示256个字符, 但是128个字符已经足够表示英语中常见的字符符号

因此最高位通常是0, 不表示任意意义的内容, 剩余的7位用来表示常见的字符

1 、0~31：控制字符或通讯专用字符(不可显示不可打印字符)，如0x07(BEL响铃)会让计算机发出哔的一声、0x00(NUL空，注意不是空格)通常用于指示字符串的结束、0x0D(CR回车)和0x0A(LF换行)用于指示打印机的打印针头退到行首(即回车)并移到下一行(即换行)等。
2 、32~126：可显示可打印字符
3 、48~57为0-9的阿拉伯数字
4 、65~90为26个大写英文字母
5 、97~122为26个小写英文字母
其余的是一些标点符号、运算符号等。
6 、127：控制字符DELETE(删除符号)

无法显示更多语言

GBXXXX编码

一个小于127的字符的意义与原来相同，但两个大于127的字符连在一起时，就表示一个汉字，前面的一个字节（他称之为高字节）从0xA1用到 0xF7，后面一个字节（低字节）从0xA1到0xFE，这样我们就可以组合出大约7000多个简体汉字了

GB2312

GBK编码　支持繁体字等

Unicode

一种字符集（Character Set），它的目标是为世界上所有的字符分配唯一的编号（码点，Code Point），以便在不同的系统、平台、语言之间实现字符的统一编码

Unicode（统一码、万国码、单一码、标准万国码）是业界的一种标准，它可以使电脑得以体现世界上数十种文字的系统

Unicode是字符集，UTF-32/ UTF-16/ UTF-8是三种字符编码方案

码点（Code Point）：每个字符对应一个唯一的编号，如汉字"你"的码点是 U+4F60。
范围：Unicode 目前定义的码点范围是 0x0000 ~ 0x10FFFF，可以表示超过一百万个字符。
本质：Unicode 只是一个"编号表"，并没有规定具体如何存储这些编号。

UTF-32

每个字符都使用4字节。就空间而言，是效率非常低的。因此UTF-32使用并不广泛

UTF-16

最明显的优点是它在空间效率上比UTF-32高两倍，因为每个字符只需要2-4个字节来存储

UTF-16 也支持可变

UTF-8

一种针对[Unicode]的可变长度字符编码

1 、128个US-ASCII字符只需一个字节编码（Unicode范围由U+0000至U+007F）。
2 、带有附加符号的拉丁文、希腊文、西里尔字母、亚美尼亚语、希伯来文、阿拉伯文、叙利亚文及它拿字母则需要二个字节编码（Unicode范围由U+0080至U+07FF）。
3 、类似汉字通常使用三个字节编码。
4 、其他极少使用的Unicode辅助平面的字符使用四字节编码。

UTF-8

Unicode编码范围（十六进制）	UTF-8编码方式（二进制）
000000 - 00007F	0xxxxxxx（ASCII编码）
000080 - 0007FF	110xxxxx 10xxxxxx
000800 - 00FFFF	1110xxxx 10xxxxxx 10xxxxxx
010000 - 10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

汉字中
1 、Unicode 码点是：20013
2 、转成 二进制 -> 010011100101101
4 、 utf-8 表示：11100100 10111000 10101101

base64

ASCII使用 7 个二进制位表示字符

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

base64 使用 6 个二进制位，对应能表示 64 个字符，所以叫 base64

1 、对应的 ascii 编码是 72 101 108 108
2 、对应二进制 01001000 01100101 01101100 01101101 01101111
3 、按照 6 位分组 010010 000110 010101 101100 011011 000110 111100 不足 6 位的用 0 补齐
4 、转位十进制 18 6 21 44 27 6 60
5 、对应的 base64 SGVsbG8 （编码后的长度要是 4 的倍数，不足用=补齐）SGVsbG8=　比如 Hello　如何转换

每5个二进制位编码成1个字符　对应 A-Z 2-7　base32

长度是 8 的倍数，不足补=