计算机中为什么会出现乱码？

—— 从乱码谈计算机编码的核心原理

本报告将通过以下问题解释编码的本质

“0和1是怎么变成文字的？”

“为什么需要编码？”

“不同的编码是如何工作？”

“中文为什么容易出现乱码？”

“如何确定一段文本所用的编码”

……

报　告 人：郝伟

报告时间：3月9日(周三) 18:00-18:30

腾讯会议：459-169-756

面向听众：公司所有技术人员

北京华云安信息技术有限公司

目录

CONTENTS

Part 1
背景：乱码现象

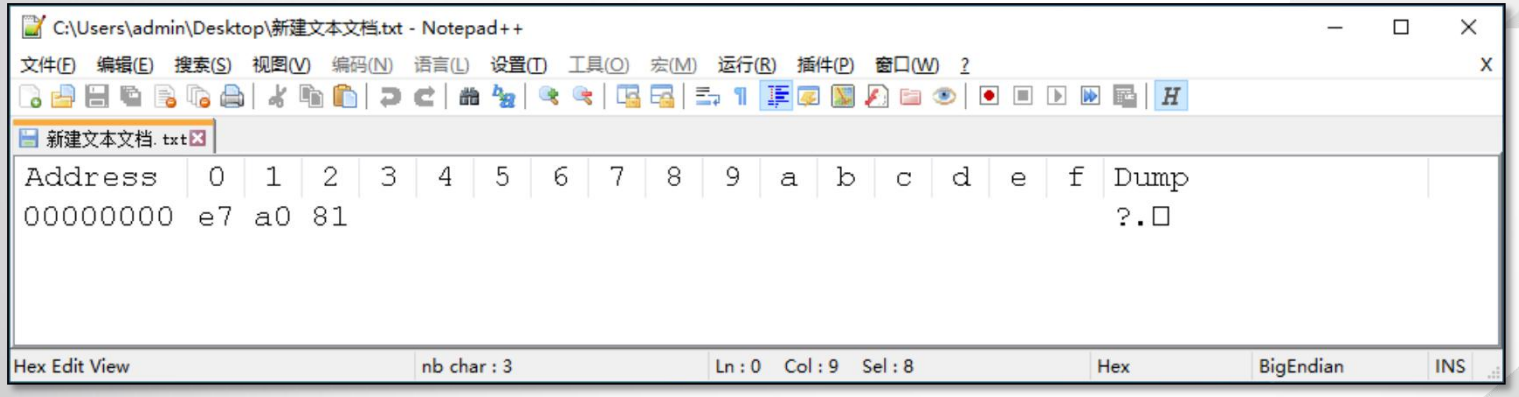
乱码示例

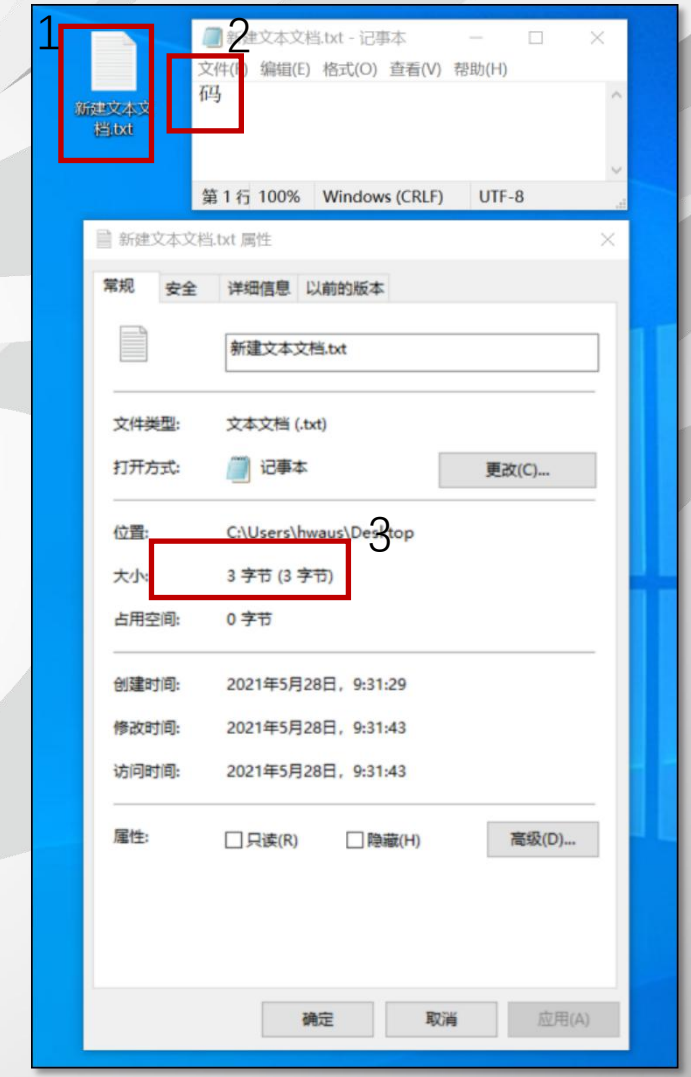
```
#############################################################################
# 1. 灏嗗敓鎴愭儴 HTML 鍐呭  鍐欏叆鎿嶄綔涓欢
#############################################################################
# 蹇呴渶鏈変嚑灏戜釜鍙傛暟锛1涓  矾寰勫悕鏂囦欢锛
if len(sys.argv) == 1:
    print(f'path "{sys.argv[1]}"does not exit.')
    exit(0)

# 璺  緞蹇呴渶瀛樺湪
if not os.path.exists(sys.argv[1]):
    print(f'path "{sys.argv[1]}"does not exit.')
    exit(0)

# content.txt 蹇呴渶瀛樺湪
dir = sys.argv[1]
contentfile=os.path.join(dir, "zcontent.txt")
if not os.path.exists(contentfile):
    print(f'The file {contentfile} does not exist.')
    exit(0)
```

代码来源 http://121.199.10.158:8107/pyupdate.py

以文本进行存储

- 1、新建一个记事本文件输入“码”并保存
- 2、查看属性发现文件大小为3个字节
- 3、如果直接查看二进制，会看到三个字节为 **E7 A0 81**

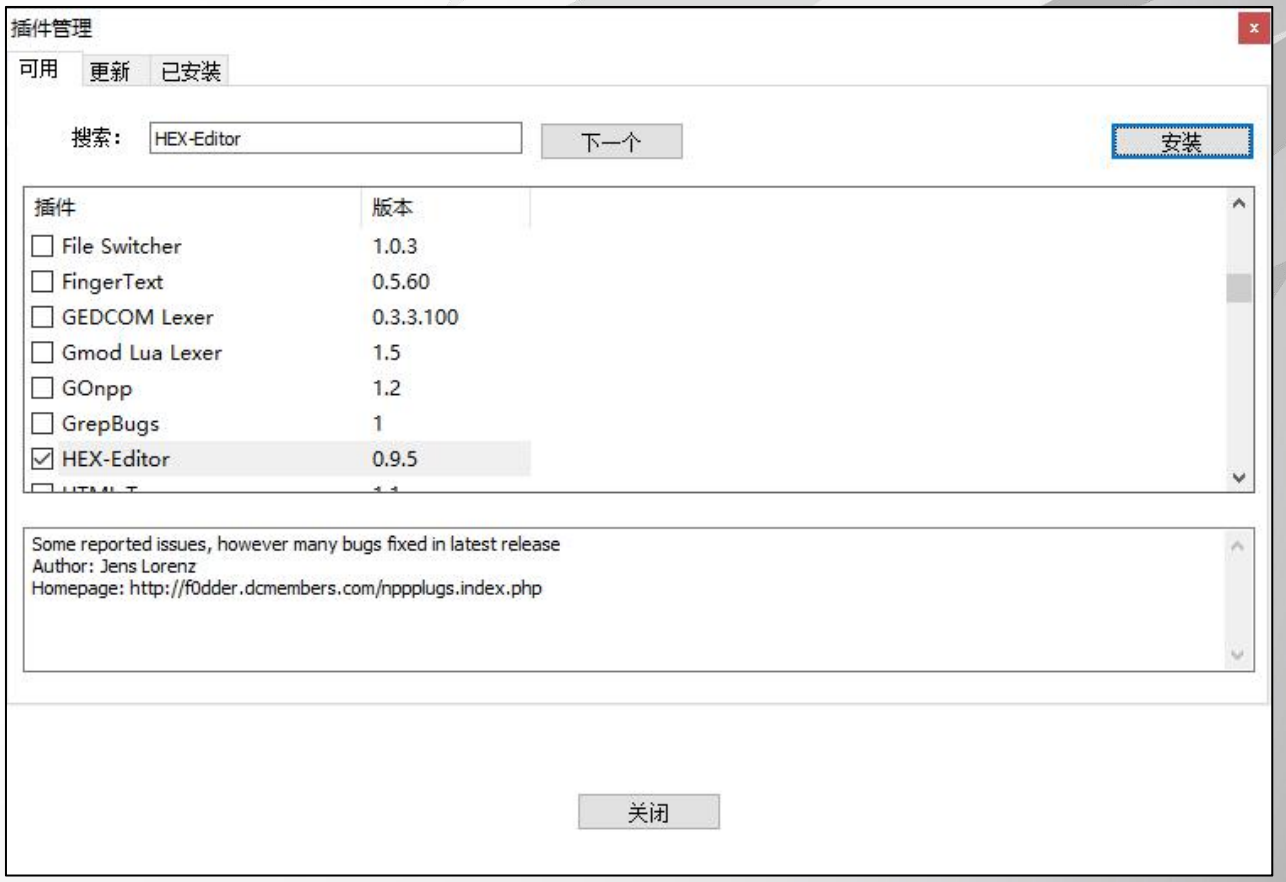




- **问题：**
  - **1、长度为什么是3?**
  - **2、内容为什么出现e7 a0 81?**

查看二进制编码
使用Notepad++配合HEX-Editor插件

- 1.打开notepad++软件
- 2.点击工具栏中的插件,插件管理,搜索HEX-Editor,选中点击安装
- 3.等待下载安装完成,重新打开软件
- 4.打开一个2进制文件,点击工具栏中的H,就可以显示16进制了



字符的两种显示方式

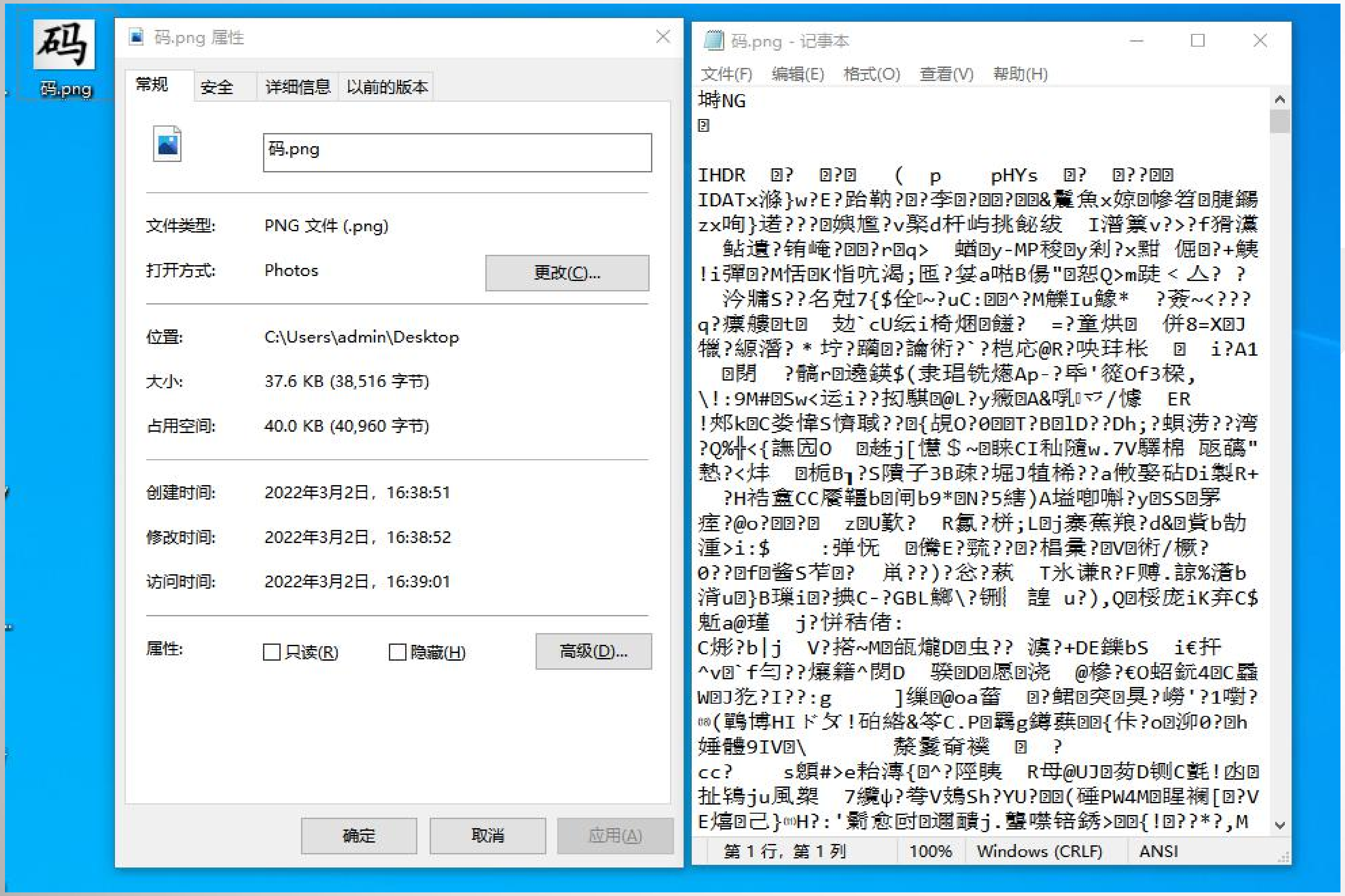
方式1：图片

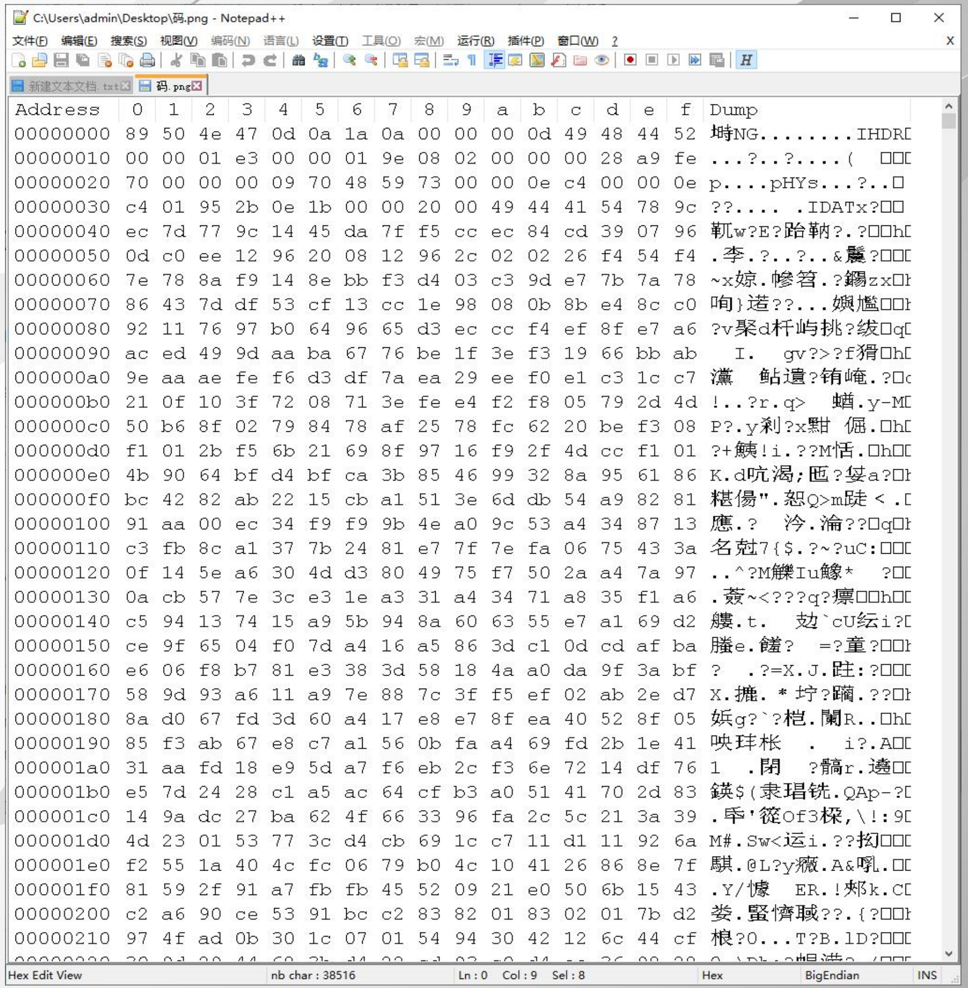
码

方式2：文本编码

方式1：使用图片进行文字表示

以图片进行存储


码.png
码.png 属性
常规
安全
详细信息
以前的版本
码.png
文件类型: PNG 文件 (.png)
打开方式: Photos
更改(C)...
位置: C:\Users\admin\Desktop
大小: 37.6 KB (38,516 字节)
占用空间: 40.0 KB (40,960 字节)
创建时间: 2022年3月2日, 16:38:51
修改时间: 2022年3月2日, 16:38:52
访问时间: 2022年3月2日, 16:39:01
属性: 只读(R) 隐藏(H) 高级(D)...
确定
取消
应用(A)
码.png - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
第 1 行, 第 1 列
100%
Windows (CRLF)
ANSI

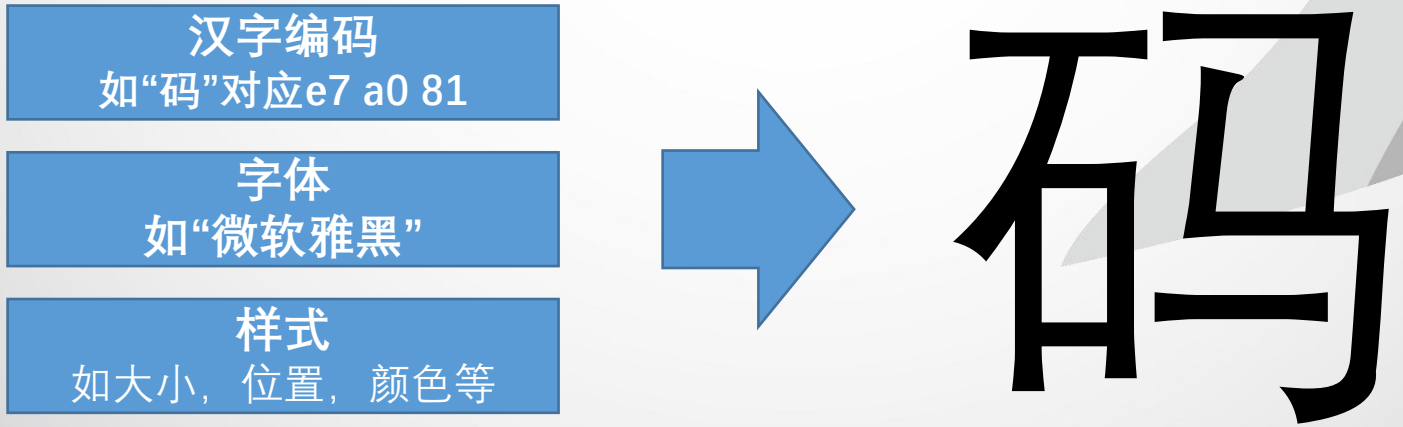

C:\Users\admin\Desktop\码.png - Notepad++
Address 0 1 2 3 4 5 6 7 8 9 a b c d e f Dump
00000000 89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52
Hex Edit View
nb char : 38516
Ln : 0 Col : 9 Sel : 8
Hex
BigEndian
INS

方式2：使用编码进行文字显示

- **编码**：文本的**本质就是数字索引**，即使用特定数字作为编号表示指定的文字，如e7 a0 81表示“码”。
- **字体**：每个字符的外形都是有一定复杂性，需要使用一定的方式进行描述的，比如汉字“码”，有很多笔划，肯定无法使用3个字节表示，所以通过字体文件对其外观进行描述。所以字体文件通常比较大，比如微软雅黑的字体文件(共3个文件）大小为46.3M

  注：关于字体的知识也很丰富，如矢量，不在本讲座范围内。

- 样式：字体在实际显示时的大小、斜体、颜色、粗细等，多在在富文本应用场合时定义。



Part 2
第1阶段：ASCII 编码

ASCII码简介

由于计算机早期是在美国发明的，所以只当时只考虑了英文的情况：

- 1、只包括字母+数字 + 标点 + 特殊符号；
- 2、数量加一起总共不到128个。

所以 ASCII使用1个字节表示。

如：

- **10** 表示换行符 \n，
- **48** 表示数字 **0**，
- **65** 表示字母 **A**，
- **97** 表示小写字母 **a**。

ASCII表

（ American Standard Code for Information Interchange 美国标准信息交换代码 ）

高四位 / 低四位		ASCII控制字符 0000						ASCII控制字符 0001						ASCII打印字符 0010		ASCII打印字符 0011		ASCII打印字符 0100		ASCII打印字符 0101		ASCII打印字符 0100		ASCII打印字符 0111		
		0						1						2		3		4		5		6		7		
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0	空字符	16	►	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH		标题开始	17	◄	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	^Backspace 代码：DEL

ASCII的短板

最多只能表示最多256个字符

问题：非英文国家怎么办?

我们也要使用计算机!

私たちもコンピュータを使います!

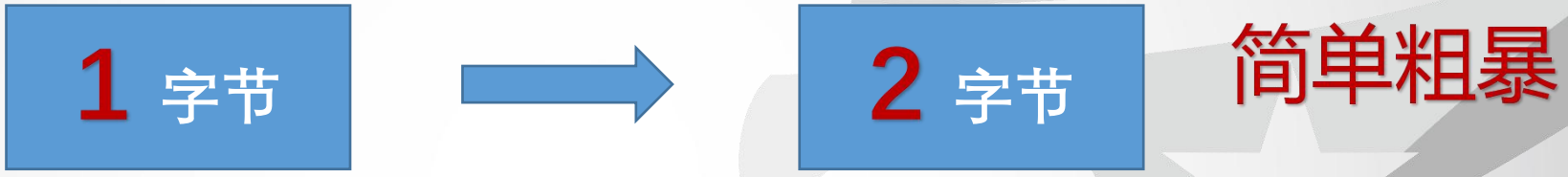
Мы тоже пользуемся компьютерами!

Wir nutzen ebenfalls den computer!

Part 3
第2阶段：ANSI 编码

ANSI简介

ANSI：扩展编码长度，以支持更多的字符集。



简单粗暴

ANSI同时**兼容**ASCII：

编码字节以0开头是原来的ASCII取1个字节，若以1开头的是新的ANSI取2个字节

ASCII　0XXX XXXX　　长度仍然是1个字节

ANSI　1XXX XXXX　XXXX XXXX　　长度是2个字节

不同的国和地区都编制了自己的ANSI

不同的国家和地区制定了各自的标准，由此产生了 GB2312、GBK、GB18030、Big5、Shift_JIS 等各自的编码标准。

如：

- 简体中文Windows操作系统中，ANSI 编码为GB2312；
- 繁体中文Windows操作系统中，ANSI编码为Big5；
- 日文Windows操作系统中ANSI 编码为 JIS。

关于中文的ANSI编码

- 1980年，国家发布GB2312，共收录了7445个字符，包括6763个汉字和682个其它符号。汉字区的内码范围高字节从B0-F7，低字节从A1-FE，占用的码位是72*94=6768。其中有5个空位是D7FA-D7FE。
- 由于GB2312支持的汉字太少，1995年的汉字扩展规范GBK1.0收录了21886个符号，它分为汉字区和图形符号区。汉字区包括21003个字符。从ASCII、GB2312到GBK，这些编码方法是向下兼容的，即同一个字符在这些方案中总是有相同的编码，后面的标准支持更多的字符。在这些编码中，英文和中文可以统一地处理。区分中文编码的方法是高字节的最高位不为0。按照程序员的称呼，GB2312、GBK都属于双字节字符集 (DBCS)。
- 2000年国家又发布了GB18030是取代GBK1.0的正式国家标准。该标准收录了27484个汉字，同时还收录了藏文、蒙文、维吾尔文等主要的少数民族文字。从汉字字汇上说，GB18030在GB13000.1的20902个汉字的基础上增加了CJK扩展A的6582个汉字（Unicode码0x3400-0x4db5），一共收录了27484个汉字。

ANSI的问题

- **表示空间有限**

  最大只能表示 2^{16}=65536种字符;

- **乱码问题**

  各个国家地区都定义了自己的标准，如中文的GB2312、GBK，台湾的Big5、日本的Shift_JIS等，导致，比如“码”的GB2312编码为 3475，而在Big5中3475对应的“鎷”，所以使用Big5打开GB2312保存的“码”字，就会显示成“鎷”，如果使用Shift_JIS打开，由于这个编码系统中不存在3475这个编码，就会显示为“ﾂ ”这样的乱码。这就是早期乱码问题的由来

- **核心问题**：在同一个系统中，难以显示多种不同的语言。

中华人民共和国国家标准信息交换用汉字编码字符集基本集GB2312-80 https://www.qqxiuzi.cn/zh/hanzi-gb2312-bianma.php

Part 4
第3阶段：Unicode 编码

Unicode编码简介

- 产生背景：互联网的快速发展迫切需要统一的语言编码方案以方便全球化交流。
- 核心目标：实现了一套编码实现对所有语言的支持。
- 历史上曾经有两家组织试图设计一套编码统一所有语言:
  - 国际标准化组织（ISO）的 ISO 10646项目。
  - 软件制造商的协会（unicode.org）的 Unicode项目。
- 标准合并
  - 早期竞争，两个组织在早期各自为政治，直到1991年，双方都认识到世界不需要两个不兼容的字符集。于是它们开始合并双方的工作成果，并为创立一个单一编码表而协同工作。从 Unicode 2.0开始，Unicode项目采用了与 ISO 10646-1相同的字库和字码。目前两个项目仍都存在，并独立地公布各自的标准。
  - 编码统一，Unicode协会现在的最新版本是2005年的Unicode 4.1.0，ISO的最新标准是ISO 10646-3:2003，两者内容上是统一的。

PS：“Unicode” 中的 “uni”表示唯一，“code”表示编码，即意为“统一的编码”。

Unicode 核心内容

- Unicode 学名 "Unicode Character Set"，简称为**UCS**。
- Unicode是一套编码规范，用于对全球主流语言的文字进行统一编码，即：

  全球主流语言的每一个文字（甚至不是文字，如emoji表情）都有唯一的编码
- Unicode具有的两个非常重要的特性：
  - 唯一性：任意1个字符只有与之对应的唯一编码；
  - 不变性：每个字符的编码是固定不变的，无论在任何语言的系统中都是一样的。
- UCS只是规定如何编码，与如何使用字节进行表示无关。例如字母“A”由于向下兼容，所以其编码仍然是65，可以使用单字节表示0x40，也可以使用双字节0x0040，还可以是4个字节表示0x00000040。

Unicode 编码分布

与IP地址分段类似，Unicode的编码空间分为17个平面，如下表所示。

平面编号	编码范围	平面名称
Plane 0	U+0000 ~ U+FFFF	基本多文种平面（Basic Multilingual Plane, BMP）
Plane 1	U+10000 ~ U+1FFFF	多文种补充平面（Supplementary Multilingual Plane, SMP）
Plane 2	U+20000 ~ U+2FFFF	表意文字补充平面（Supplementary Ideographic Plane, SIP）
Plane 3	U+30000 ~ U+3FFFF	表意文字第三平面（Tertiary Ideographic Plane, TIP）.
Plane 4 ~ 13	U+40000 ~ U+4FFFF	未使用（unassigned）
Plane 14	U+E0000 ~ U+EFFFF	特别用途补充平面（Supplementary Special-purpose Plane, SSP）
Plane 15 ~ 16	U+F0000 ~ U+10FFFF	保留作为私人使用区（Private Use Area, PUA）

参考 https://blog.csdn.net/oyji1992/article/details/80030366

Unicode的实现方式

- 据统计，2021年末世界上所有的语言共5651，即使按每种语言都有3万字符（实际远远达不到），那么使用3个字节（2^{24}=16,777,216) 就可以表示了，但考虑向下兼容和分区等原因，Unicode使用了4个字节进行表示。
- 使用4个字节会造成一定的空间浪费。例如使用Unicode编码表示“华云安AI.VUL”，就需要9*4=36个字节。但实测发现保存在记事本中只需要**15**个字节，内容如下：

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000000	e5	8d	8e	e4	ba	91	e5	ae	89	41	49	2e	56	55	4c	

- 这就是编码**实现方式**问题，查看可以发现在记事本中使用的编码为**UTF-8。**
- UTF-8 是Unicode的一种实现方式。

  类似的，UTF-16，UTF-32等都是Unicode的实现方式。

UTF-8、UTF-16和UTF-32

- 常见的UTF格式有UTF-8、UTF-16、UTF-32，都是被广泛接受的方案。
  - UTF-8，可变长的编码方式，使用1-4字节表示1个字符，能够在保证可编码范围的前提下，有效地减少了编码长度。唯一的缺点是无法直接估算一段字符的总字节长度。
  - UTF-16，固定长度2个字节，是UCS-32的子集、UCS-2的父集。节省空间，但可表示的范围有限。
  - UTF-32，固定使用4个字节来表示一个字符，具有使用简单，长度可计算等优点，在一些对计算性能有要求但速度或存储空间无要求的场合。
  - UTF-5，UTF-6，UTF-7等一般是USC的子集，在特殊领域使用，具体内容不再细表。

Character	Encoding	Bits	Bytes
A	UTF-8	010000001	1
A	UTF-16	00000000 010000001	2
A	UTF-32	00000000 00000000 00000000 010000001	4

PS: UTF是“UCS Transformation Format”的缩写，即Unicdoe统一转换编码。

汉字“码”的UTF-8编码内容
HEX: e7 a0 81
BIN: 11100111 10100000 10000001

Part 5
经验分享

推荐使用UTF-8编码

中文常见的编码 UTF-8，UTF-32，GB2312，推荐使用UTF-8，理由：

- 完全支持Unicode格式；
- 占用空间最少的Unicdoe编码；
- 文本数据使用量最为广泛；
- 主流的文本编辑器都支持UTF-8；
- 主流的编程语言都支持UTF8格式；
- CPU运算速度快，计算文本字节长度耗时可以忽略。

如何确定一个文本文件的编码格式

通常三种方法：标记法、设定法和推断法。

- 标记法：由于Unicde兼容ASCII，所以一些文件通常会在首行会标记编码类型，如：
  - XML文件第一行通常会有 `<?xml version="1.0" encoding="utf-8" ?>`;
  - Python文件第一行通常包含 `# -- coding: utf-8 -- 或直接写 # encoding=utf-8`

  由于兼容性，所以ASCII都可以正常读取，从而帮助文本编辑器确定后面的编码格式。
- 设定法：在文本编辑器中主动设定的编码格式方式进行判定，主流编辑器都支持。
- 推断法：软件根据文本的字节内容进行分析推断，比如Windows的记事本就具有这样的能力，一个没有编码说明的文件，记事本通常都能够以正确的编码显示文本内容。

为何支持Unicode显示仍有乱码

主要有以下原因：

- 缺少字体

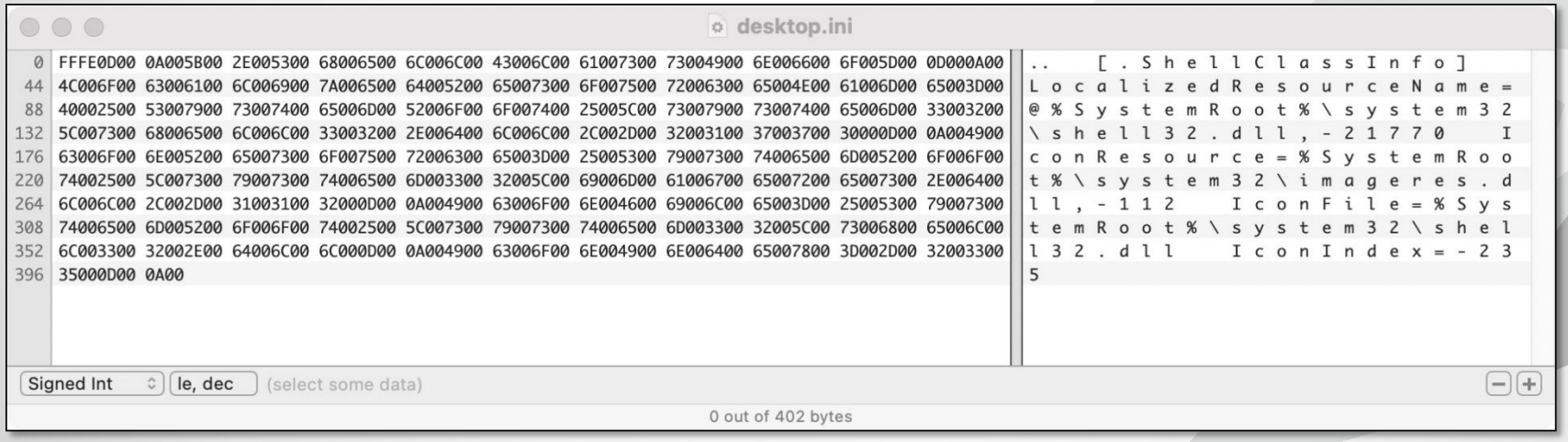
  通常是因为缺少相应字体导致的，例如中文字体的体积较大，很多系统中没有内置中文字体。类似的，如果在中文系统中使用Unicode编码也无法显示一些字符，如阿拉伯文，同样可能是由于缺少字体文件造成的。

- 使用了UTF-16等编码方式

  由于UTF-16只是Unicode的一个子集，所以会导致部分文字无法正常显示。

带有二进制显示功能的文本编辑器

- Windows推荐Notepad++
  - 通过安装插件 HEX Editor 实现二进制查看
- MacOS 推荐 Hex Fiend



- 跨平台推荐 VSCode
  - 通过安装插件 Hex Dump 实现二进制查看

全文总结

- 编码发展阶段
  - 第1阶段：ASCII码，只支持少量英文字母和符号；
  - 第2阶段：ANSI，所有的语言都可以使用，但是语言间是独立的，同时也有多种编码方案。
  - 第3阶段：Unicode，统一的语言编码，支持全球绝大部分语言。
- Unicode编码实现方式
  - Unicode 有多种实现方式，如UTF-8，UTF-16，UTF-32等；
  - 编码建议：尽量使用UTF-8
- 推荐文本编码器
  - Windows：Notepad++ Hex Editor；
  - Mac OS: Hex Fiend;
  - 多平台: VSCode Hex Dump。

谢谢！