



NLP

自然语言处理技术基础

Natural Language Processing, NLP

网络空间安全与计算机学院

第6章 字符编码与字频统计

6.1 西文字符编码 ASCII

6.2 中文字符编码 GB2312、BIG5、Unicode、GBK、GB 18030

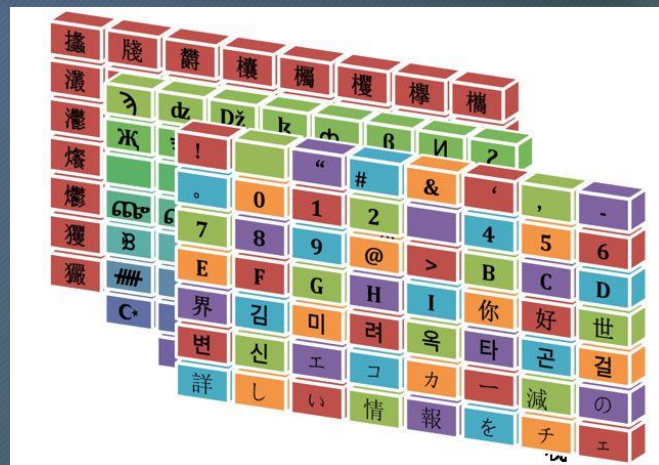
6.3 字符编码知识的作用

6.4 字频统计

字符 字符编码

字符

- 文字 + 符号
- 各国家文字、标点符号、图形符号、数字等
- 文本处理中最基本单位




字符编码

- 输入编码（外码）：输入字符时需要敲哪些键（输入法）
- 机内编码（内码）：用什么数字来表示和存储某个字符

6.1 西文字符编码

ASCII 码

- 美国标准信息交换码 American Standard Code for Information Interchange
- 美国国家标准局（ANSI）制定
- 国际标准化组织（ISO）定为国际标准：ISO/IEC 646



Standards About us News Taking part Store

ICS 35 35.040 35.040.10

ISO/IEC 646:1991

Information technology – ISO 7-bit coded character set for information interchange

THIS STANDARD WAS LAST REVIEWED AND CONFIRMED IN 2002.
THEREFORE THIS VERSION REMAINS CURRENT.

<https://www.iso.org/standard/4777.html>

ASCII 表																									
高四位		ASCII 码控制字符										ASCII 码打印字符													
		0000					0001					0010		0011		0100		0101		0110		0111			
		十进制	字符	Ctrl	代码	转义字符	十进制	字符	Ctrl	代码	转义字符	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	十进制	字符	Ctrl	
低四位		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
0000	0	0	^@	NUL	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 横向指标	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	l				
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	SOH	\s 移出	30	^_	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~				
1111	F	15	^O	SE	\s 移入	31	^_	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	?				

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入

Backspace 代码 DEL

6.2 中文字符编码

6.2.1 国标码 GB 2312-1980

6.2.2 大五码 big5

6.2.3 Unicode与ISO/IEC 10646

6.2.4 国标扩展码 GBK

6.2.5 GB 18030

6.2.1 国标码 GB2312码

GB2312码: 1980年国家标准总局发布国标码

- 全称：《信息交换用汉字编码字符集——基本集》
- 两个字节表示一个汉字，ASCII码都大于127：**161(A1)-254(FF)**间的整数
- 编码空间共**8836**个：**汉字6763个**、非汉字字符**682**个、空位**1391**个

CODE: 输出字符和该字符两个字节的ASCII码

161(A1)-254(FF)间的整数

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int main()
7  {
8      cout << "Begin GB2312:" << endl;
9      FILE * outfile;
10     outfile = fopen("gb2312-80.chr", "wt");
11
12     unsigned char i, j;
13
14     for(i = 161; i < 255; i++)
15         for(j = 161; j < 255; j++)
16             {
17                 fprintf(outfile, "%c%c,%d,%d\n", i, j, i, j);
18             }
19     fclose(outfile);
20
21     cout << "END GB2312!" << endl;
22
23     return 0;
24 }
25
```

```
for i in range(161, 254):
    for j in range(161, 254):
        s = hex(i) + hex(j)
        s = s.replace('0x', '')
        bi = [int(s[0:2], 16),]
        bi = bi + [int(s[2:4], 16),]
        bs = bytearray(bi)
        print(bytes(bs).decode('gb18030'), i, j, ' ', end='')
        if j==9:
            print('\n')
```

161 161	、	161 162	。 161 163	• 161 164	˘ 161 165	˙ 161 166	˚ 161 167	˛ 161 168	ˇ 161 169	˜ 161 170
˘ 161 171	‖ 161 172	… 161 173	‘ 161 174	’ 161 175	“ 161 176	” 161 177	(161 178) 161 179	< 161 180	
> 161 181	《 161 182	》 161 183	「 161 184	」 161 185	〔 161 186	〕 161 187	〔 161 188	〕 161 189	【 161 190	
】 161 191	± 161 192	× 161 193	÷ 161 194	:	161 195	∧ 161 196	√ 161 197	Σ 161 198	Π 161 199	∪ 161 200
∩ 161 201	∈ 161 202	:: 161 203	✓ 161 204	⊥ 161 205	// 161 206	∠ 161 207	∩ 161 208	⊙ 161 209	∫ 161 210	
∫ 161 211	≡ 161 212	≅ 161 213	≈ 161 214	∞ 161 215	≠ 161 216	≠ 161 217	≠ 161 218	≠ 161 219	≠ 161 220	
≥ 161 221	∞ 161 222	∞ 161 223	∞ 161 224	∞ 161 225	∞ 161 226	∞ 161 227	∞ 161 228	∞ 161 229	∞ 161 230	
\$ 161 231	⊙ 161 232	⊙ 161 233	⊙ 161 234	⊙ 161 235	⊙ 161 236	⊙ 161 237	⊙ 161 238	⊙ 161 239	⊙ 161 240	
● 161 241	◎ 161 242	◊ 161 243	◆ 161 244	◻ 161 245	■ 161 246	△ 161 247	▲ 161 248	※ 161 249	→ 161 250	
← 161 251	↑ 161 252	↓ 161 253	i 162 161	ii 162 162	iii 162 163	iv 162 164	v 162 165	vi 162 166	vii 162 167	
viii 162 168	ix 162 169	x 162 170	162 171	162 172	162 173	162 174	162 175	162 176	162 177	
2. 162 178	3. 162 179	4. 162 180	5. 162 181	6. 162 182	7. 162 183	8. 162 184	9. 162 185	10. 162 186	11. 162 187	
12. 162 188	13. 162 189	14. 162 190	15. 162 191	16. 162 192	17. 162 193	18. 162 194	19. 162 195	20. 162 196	(1) 162 197	
(2) 162 198	(3) 162 199	(4) 162 200	(5) 162 201	(6) 162 202	(7) 162 203	(8) 162 204	(9) 162 205	(10) 162 206	(11) 162 207	
(12) 162 208	(13) 162 209	(14) 162 210	(15) 162 211	(16) 162 212	(17) 162 213	(18) 162 214	(19) 162 215	(20) 162 216	(21) 162 217	
(22) 162 218	(23) 162 219	(24) 162 220	(25) 162 221	(26) 162 222	(27) 162 223	(28) 162 224	(29) 162 225	(30) 162 226	€ 162 227	
162 228	(-) 162 229	(-) 162 230	(-) 162 231	(-) 162 232	(-) 162 233	(-) 162 234	(-) 162 235	(-) 162 236	(-) 162 237	
(+) 162 238	162 239	162 240	I 162 241	II 162 242	III 162 243	IV 162 244	V 162 245	VI 162 246	VII 162 247	
VIII 162 248	IX 162 249	X 162 250	XI 162 251	XII 162 252	162 253	! 163 161	" 163 162	# 163 163	¥ 163 164	
% 163 165	& 163 166	' 163 167	(163 168) 163 169	* 163 170	+ 163 171	- 163 172	= 163 173	163 174	

GB2312中各类字符分布情况

首字节ASCII码	字符类型
161	标点、一般符号 202
162	序号 60
163	数字 22
164	拉丁字母 52
165	日文假名 169
166	希腊字母 48
167	俄文字母 66
168	汉语拼音符号 26 汉语注音字母 37
176-215	一级汉字 3755 按汉语拼音排序
216-247	二级汉字 3008 按偏旁部首排序

国务院关于公布《通用规范汉字表》的通知 2013年6月5日

http://www.gov.cn/zwgc/2013-08/19/content_2469793.htm

共收字8105个，分为三级。

一级字表：收字3500个

- 常用字集
- 主要满足基础教育和文化普及的基本用字需要

二级字表：收字3000个

- 使用度仅次于一级字
- 一、二级字表主要满足出版印刷、辞书编纂和信息处理等方面的一般用字需要

三级字表：收字1605个

- 姓氏人名、地名、科学技术术语
- 中小学语文教材文言文用字中未进入一、二级字表的较通用的字
- 主要满足信息化时代与大众生活密切相关的专门领域的用字需要

区位码： 汉字在方阵中的坐标

区码： 前两位

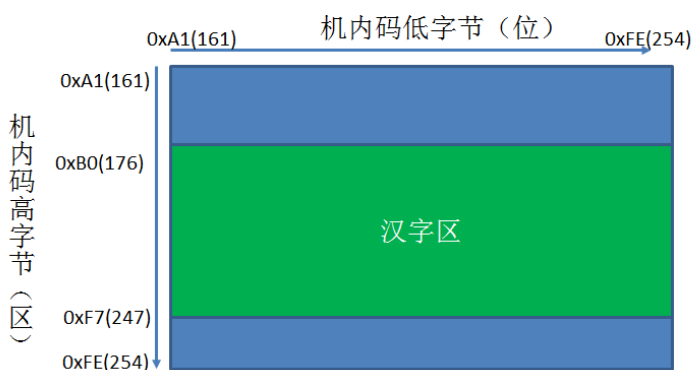
位码： 后两位

编码空间： 94×94

每一行叫一个“区”

每个区有94个“位”

GB2312



每个区包含94 (254-161+1) 个字符

“爸”字在16区54位，
所以“万”字的区位码是： 16 54

啊	阿	埃	挨	哎	唉	哀	皑	癌
蔼	矮	艾	碍	爱	隘	鞍	氨	安
俺	按	暗	岸	胺	案	肮	昂	盎
凹	熬	翱	袄	傲	奥	懊	澳	芭
捌	扒	叭	吧	笆	八	疤	巴	拔
跋	靶	把	耙	坝	霸	罢	爸	白
柏	百	摆	佰	败	拜	裨	斑	班
搬	扳	般	颁	板	版	扮	拌	伴
瓣	半	办	绊	邦	帮	梆	榜	膀
绑	棒	磅	蚌	镑	傍	谤	苞	胞
包	褒	剥						

16区

<https://www.cnblogs.com/wangmantou/p/13852024.html>

“万”字在45区82位，
所以“万”字的区位码是： 45 82

汀	廷	停	亭	庭	挺	艇	通	桐
酮	瞳	同	铜	彤	童	桶	捅	筒
统	痛	偷	投	头	透	凸	秃	突
图	徒	途	涂	屠	土	吐	兔	湍
团	推	颓	腿	蜕	褪	退	吞	屯
臀	拖	托	脱	鸵	陀	驮	驼	橐
妥	拓	唾	挖	蛙	洼	娃	瓦	袜
歪	外	腕	弯	湾	玩	顽	丸	烷
完	碗	挽	晚	皖	惋	宛	婉	万
腕	汪	王	亡	枉	网	往	旺	望
忘	妄	威						

45区

<https://zhuanlan.zhihu.com/p/27120673>

CODE: 区位码

根据 区位码 显示单个汉字

```
def show_word(zone_num, bit_num):# 根据 区位码 显示单个汉字
    r = zone_num + 0xA0
    b = bit_num + 0xA0
    s = hex(r) + hex(b)
    s = s.replace('0x', '')
    bi = [int(s[0:2], 16), ]
    bi = bi + [int(s[2:4], 16), ]
    bs = bytearray(bi)
    return bytes(bs).decode('gb2312')
```

薄	雹	保	堡	饱	宝	抱	报	暴
豹	鲍	爆	杯	碑	悲	卑	北	辈
背	贝	钡	倍	狈	备	惫	焙	被
奔	本	笨	崩	绷	甬	泵	蹦	迸
逼	鼻	比	鄙	笔	彼	碧	蓖	蔽
毕	毙	比	币	庇	痹	闭	蔽	弊
必	辟	壁	臂	避	陛	鞭	边	编
贬	扁	便	变	卞	辨	辩	辨	遍
标	彪	膘	表	鳖	憋	别	瘕	彬
斌	濒	滨	宾	摈	兵	冰	柄	丙
秉	饼	炳						

、	。	·	-	~	“	”	々
—	~		...	‘	’	“	”
()	《	》	「	」	『	』
【	】	±	×	÷	:	^	v
Σ	Π	U	∩	∈	::	√	⊥
//	∠	∩	⊙	∫	∫	≡	≡
≈	≈	≈	≈	≈	≈	≈	≈
≤	≥	∞	∴	∴	∴	∴	∴
°	°	°	°	°	°	°	°
°	°	°	°	°	°	°	°
°	°	°	°	°	°	°	°
→	←	↑	↓	≡			

输出 16区 所有文字

```
for i in range(10):
    for j in range(10):
        if (i==0 and j==0):
            print(' ', end='')
            continue
        if (i==9 and j>4):
            print(' ', end='')
            continue
        i_j = str(i)+str(j)
        print(show_word(16, int(i_j)), ' ', end='')
        if j==9:
            print('\n')
```

啊	阿	埃	挨	哎	唉	哀	皑	癌
蔼	矮	艾	碍	爱	隘	鞍	氨	安
俺	按	暗	岸	胺	案	肮	昂	盎
凹	敖	熬	翱	袄	傲	奥	懊	澳
芭	捌	扒	叭	吧	笆	八	疤	巴
拔	跋	靶	把	耙	坝	霸	罢	爸
白	柏	百	摆	佰	败	拜	裨	斑
班	搬	扳	般	颁	版	扮	拌	伴
瓣	半	办	绊	邦	帮	梆	榜	膀
绑	棒	磅	蚌	镑	傍	谤	苞	胞
包	褒	剥						

6.2.2 大五码 Big5

- 中国台湾、香港与澳门地区，使用的**繁体中文字符集**。
- 1984年，为统一繁体字符集编码，台湾**五大厂商**制定编码方案。
- 因其来源被称为五大码，英文写作**Big5**，普遍被称为**大五码**。

五大厂商：宏碁 Acer、神通 MiTAC、佳佳、零壹 Zero One、大众 FIC

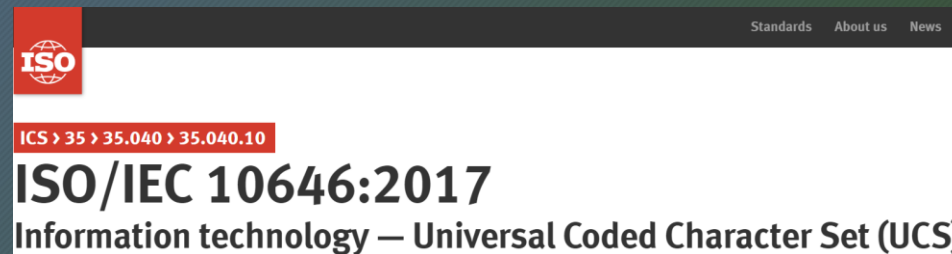
6.2.3 Unicode 与 ISO 10646

Unicode (统一码)容纳世界上所有文字和符号的字符编码方案。
ISO针对各国文字符号统一编码制定 **ISO 10646**定义标准字符集。

- 1991年，两个项目合并，同步发展 Unicode 和 ISO 10646。
- Unicode 3.0 与 ISO 10646 使用相同的字库和字码。
- 两个项目仍都存在，并独立地公布各自的标准。



Unicode集团是由美国的HP、Microsoft、IBM、Apple等几家知名的大型计算机企业所组成的联盟集团
<https://home.unicode.org/>



<https://www.iso.org/standard/69119.html>

GB13000等同采用国际标准ISO/IEC 10646-2003

ISO/IEC 10646

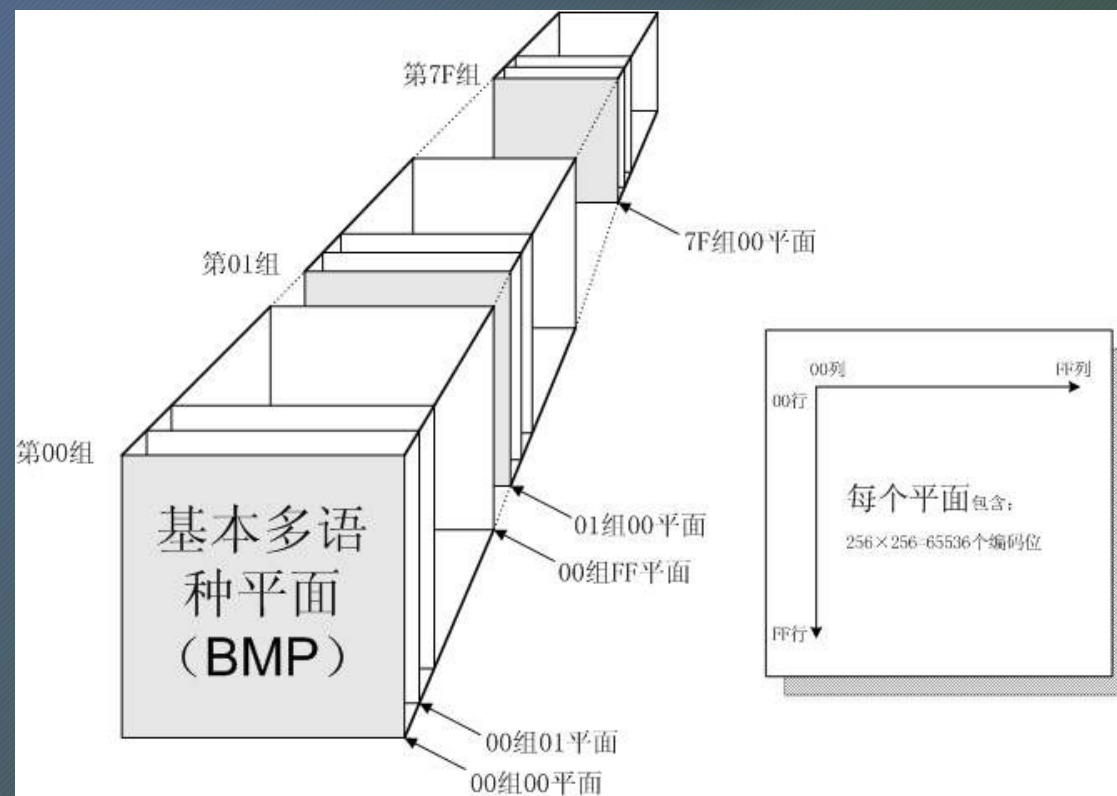
英文全称: Information technology - **U**niversal Multiple-Octet Coded Character **S**et, 简称 **UCS**。

中文全称: **信息技术-通用多八位编码字符集**, 亦称 **大字符集**。

四维编码空间, 采用十六进制全编码

- 总体分为 128个 三维组 (group), 范围: 从00到7F
- 每个组含 256个 平面 (plane), 范围: 从00到FF
- 每平面含 256个 行 (row), 范围: 从00到FF
- 每一行含 256个 码位 (cell), 范围: 从00到FF

每个字符由**四个八位序列**表示,
按照**组八位**、**面八位**、**行八位**、**列八位**的顺序:
Group-octet Plane-octet Row-octet Cell-octet



ISO/IEC 10646

基本多文种平面 BMP (Basic Multilingual Plane)

- 第0平面，即：Group0的Plane0
- 是目前实际应用的Unicode版本（2字节）
- 此平面上用行、列八位即可表示一个编码字符
- 中日韩统一表意文字 CJK Unified Ideographs

Un Roadmaps to Unicode

Tables

Roadmap Introduction

Roadmap to the BMP (Plane 0)

Roadmap to the SMP (Plane 1)

Roadmap to the SIP (Plane 2)

Roadmap to the TIP (Plane 3)

Roadmap to the SSP (Plane 14)

Not the Roadmap

<https://www.unicode.org/roadmaps/bmp/index.html>

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	C0 Controls				Basic Latin				C1 Controls				Latin 1 Supplement			
01	Latin Extended-A				IPA Extensions				Latin Extended-B				Spacing Modifiers			
02	Latin Extended-B				Combining Diacritics				Greek							
03																
04	Cyrillic Sup.				Armenian				Cyrillic				Hebrew			
05																
06																
07	Syriac				Arabic Sup.				Arabic				Thaana			
08					Syr. Sup.				(Arabic Extended-B)				Arabic Extended-A			
09	Samaritan				Mandaic								N'Ko			
0A					Devanagari								Bengali			
0B					Gurmukhi								Gujarati			
0C					Oriya								Tamil			
0D					Telugu								Kannada			
0E					Malayalam								Sinhala			
0F					Thai								Lao			
10																
11					Tibetan											
12																
13					Myanmar								Georgian			
14																
15																
16																
17																
18																
19																
1A																
1B																
1C																
1D																
1E																
1F																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
2A																
2B																
2C																
2D																
2E																
2F																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
3A																
3B																
3C																

基本多文种平面的示意图



黑 = 拉丁文字及符号
浅蓝 = Linguistic scripts
蓝 = 其他欧洲文字
橘 = Middle Eastern and SW Asian scripts
浅橘 = 非洲文字
绿 = 南亚文字
紫 = 东南亚文字
红 = 东亚文字
浅红 = 中日韩汉字 CJK
黄 = Aboriginal scripts
紫红 = 符号
深灰 = Diacritics
浅灰 = UTF-16 surrogates and private use
蓝青 = Miscellaneous characters
白 = 未使用

每个写着数字的格子代表256个码点

Unicode转换格式 (Unicode Translation Format , UTF)

UTF是Unicode的实现方式，即怎样将Unicode定义的数字转换成程序数据

例如，“汉字”对应的数字是0x6c49和0x5b57，而编码的程序数据是：

```
BYTE    data_utf8[]    = {0xE6, 0xB1, 0x89, 0xE5, 0xAD, 0x97}; // UTF-8 编码
WORD    data_utf16[]   = {0x6c49, 0x5b57};                      // UTF-16编码
DWORD   data_utf32[]  = {0x6c49, 0x5b57};                      // UTF-32编码
```

BYTE、WORD、DWORD分别表示无符号8位整数，无符号16位整数和无符号32位整数。

“汉字”的UTF-8	编码需要六个BYTE，	6个字节。
“汉字”的UTF-16	编码需要两个WORD，	4个字节。
“汉字”的UTF-32	编码需要两个DWORD，	8个字节。

UTF-8

UTF-8的特点是对**不同范围的字符**使用**不同长度的编码**

- 对于0x00-0x7F之间的字符，UTF-8编码与ASCII编码完全相同。
- UTF-8编码的最大长度是4个字节。
- 从下表可以看出，4字节模板有21个x，即可以容纳21位二进制数字。
- Unicode的最大码位0x10FFFF只有21位。

Unicode编码(16进制)	UTF-8 字节流(二进制)
000000 - 00007F	0xxxxxxx
000080 - 0007FF	110xxxxx 10xxxxxx
000800 - 00FFFF	1110xxxx 10xxxxxx 10xxxxxx
010000 - 10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

UTF-8

例1: “汉”字的Unicode编码是: 0x6C49。

① 0x6C49在0x0800-0xFFFF之间, 使用3字节模板

② 0x6C49写成二进制是: **0110 1100 0100 1001**

③ 比特流代替模板的x: **11100110 10110001 10001001**, 即: E6 B1 89

Unicode编码(16进制)	UTF-8 字节流(二进制)
000000 - 00007F	0xxxxxxx
000080 - 0007FF	110xxxxx 10xxxxxx
000800 - 00FFFF	1110 xxxx 10 xxxxxx 10 xxxxxx
010000 - 10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

UTF-8

例2: “ 岱 ”字Unicode编码: 0x20C30 (中日韩统一表意文字扩展)

- ① 0x20C30在0x010000-0x10FFFF之间, 使用4字节模板。
- ② 0x20C30写成21位二进制数字 (不足21位就在前面补0) : 0 0010 0000 1100 0011 0000
- ③ 比特流依次代替模板中的x, 得到:
11110000 10100000 10110000 10110000, 即: F0 A0 B0 B0。

Unicode编码(16进制)	UTF-8 字节流(二进制)
000000 - 00007F	0xxxxxxx
000080 - 0007FF	110xxxxx 10xxxxxx
000800 - 00FFFF	1110xxxx 10xxxxxx 10xxxxxx
010000 - 10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx



CODE

Unicode 转 汉字、UTF-8

```
▶ s = '\u6c49'
print(chardet.detect(str.encode(s)))
print(s)
print(s.encode('utf-8'))

{'encoding': 'Windows-1252', 'confidence': 0.73, 'language': ''}
汉
b'\xe6\xba\x89'
```

```
▶ ss = '\u20C30' # 中日韩统一表意文字扩展 会 输出乱码, UTF-8 也不对
print(chardet.detect(str.encode(ss)))
print(ss)
print(ss.encode())

{'encoding': 'Windows-1252', 'confidence': 0.73, 'language': ''}
0
b'\xe2\x83\x830'
```

汉字 转 Unicode

```
▶ def to_unicode(string):
    ret = ''
    for v in string:
        ret = ret + hex(ord(v)).upper().replace('0X', '\\u')
    return ret

print(to_unicode("汉"))

\u6C49
```

```
▶ s = '河'
print(s)
print(ord(s))
print(bin(ord(s)))
print(s.encode('utf-8'))
print(s.encode('gb2312'))

河
27827
0b110110010110011
b'\xe6\xb2\xb3'
b'\xba\xdc'
```


6.2.4 国标扩展码 GBK

GBK即： 汉字内码扩展规范（1995.12.1颁布）

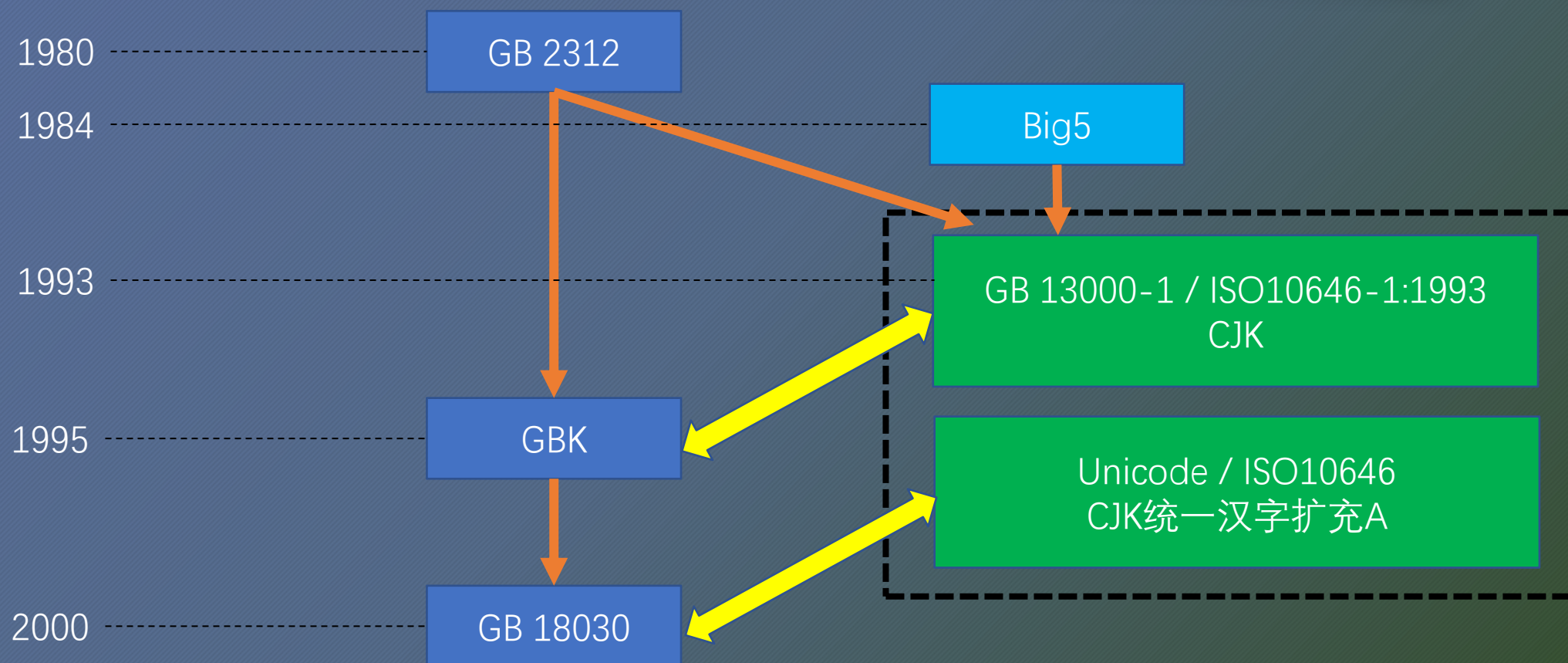
- 向下兼容GB2312， 向上支持ISO 10646， 承上启下
- 保持GB2312原貌， 扩充至与ISO 10646的CJK等量， 包含Big5
- GBK共20902个汉字（GB2312共6763个汉字）

6.2.5 GB 18030

GB 18030 《信息技术 中文编码字符集》

- GB18030-2000 《信息技术 信息交换用汉字编码字符集基本集的扩充》
- GB18030-2005
- 向下兼容GB2312 和 GBK
- 与Unicode的码位一一对应
- GB 18030共收录汉字70,244个

主要中文字符编码体系之间的关系



6.3 字符编码知识的作用

1. 便于表示控制字符
2. 便于在编程中对字符进行分类

CODE

1. 便于表示控制字符

回车, ASCII码13

换行, ASCII码10

空格, ASCII码32

```
▶ print("10位ASCII码转为字符:", chr(13))
```

10位ASCII码转为字符:

```
▶ str = input("请输入: ");  
if (str == chr(32)):  
    print ("你输入的内容是: 空格")  
else:  
    print ("你输入的内容是: ", str)
```

请输入:
你输入的内容是: 空格

2. 编程中对字符分类

ord() 返回对应的 ASCII 数值, 或者 Unicode 数值

```
▶ str = input("请输入: ");  
if (ord(str) < 128):  
    print ("你输入的内容是: 西文字符")  
elif(ord(str) >= 176):  
    print ("你输入的内容是: 中文字符")  
else:  
    print ("你输入的内容是: 其他字符")
```

请输入: 3
你输入的内容是: 西文字符

6.4 字频统计

6.4.1 字频统计的应用

6.4.2 单字字频统计

6.4.3 双字字频统计

6.4.1 字频统计的应用

1. 汉字输入
2. 汉字识别
3. 中文文本校对
4. 词汇获取

1 汉字输入

中文输入法：汉字输入系统

- 高字频，减少输入长度
- 高字频、词频，排序靠前

2 汉字识别

印刷汉字识别

手写汉字识别

字与字的同现关系，能提高汉字识别的正确率

同现关系指的是词汇共同出现的倾向性。

在语篇中，围绕一定的话题，一定的词往往会同时出现，而答其他一些词汇就不大可能出现或根本不会出现。这种词的同现关系与语篇范围关系非常密切。

3 中文文本校对

检查文本中 语法、 词汇、 文字 方面的错误

使用正常语料中统计出来的字频数据是 正字 出现规律,
可以用来帮助识别 别字

4 词汇获取

未登录词

1. 从大规模真实文本中统计双字、三字、四字……的连续同现频率
2. 然后计算某种统计量
3. 把统计量在某个阈值之上的双字、三字、四字……作为候选词
4. 再利用其他方法（如人工检查）对候选词进行甄别

6.4.2 单字字频统计

输入：

文本文件

输出：

文件中不同汉字的个数

每个汉字出现的次数

流程图、算法、编程实现

```
import re
import operator

my_str = "输出: hello world, 你好世界, 世界你好, 你好Python输入."

def str_count(strs):
    str_dict = {}
    result = re.compile(u'[\u4e00-\u9fa5]') # 正则表达式 判断汉字
    for i in strs:
        if result.search(i):
            str_dict[i] = str_dict.get(i, 0) + 1 # 利用字典中的get方法,
    return str_dict

x = str_count(my_str)
print (x)
sorted_x = sorted(x.items(), key=operator.itemgetter(1), reverse=True)
for i in sorted_x:
    print(i)

{'输': 2, '出': 1, '你': 3, '好': 3, '世': 2, '界': 2, '入': 1}
('你', 3)
('好', 3)
('输', 2)
('世', 2)
('界', 2)
('出', 1)
('入', 1)
```


6.4.3 双字字频统计

输入：
文本文件

输出：
文件中不同字对的个数
每个字对出现的次数

流程图、算法、编程实现

```
import operator
import re
strs = "发展中国家（Developing country）也称作开发中国家、欠发达国家，指经济、技术、人民生活水平程度较低的国家，与发达国家相对。"

str_dict = {}
word_dict = {}
temp = ''

for i in strs:
    result = re.compile(u'[\u4e00-\u9fa5]')
    if result.search(i):
        str_dict[i] = str_dict.get(i, 0) + 1
        word = temp + i
        if temp != '':
            word_dict[word] = word_dict.get(word, 0) + 1
        temp = i
    else:
        temp = ''

print (str_dict)
# print (word_dict)

('发': 4, '展': 1, '中': 2, '国': 5, '家': 5, '也': 1, '称': 1, '作': 1, '开': 1, '欠': 1, '达': 2, '指': 1, '经': 1, '济': 1, '技': 1, '术': 1, '人': 1, '民': 1, '生': 1, '活': 1, '水': 1, '平': 1, '程': 1, '度': 1, '较': 1, '低': 1, '的': 1, '与': 1, '相': 1, '对': 1)

sorted_x = sorted(word_dict.items(), key=operator.itemgetter(1), reverse=True)
for i in sorted_x:
    print(i)

('国家', 5)
('中国', 2)
('发达', 2)
('还国', 2)
('发展', 1)
('展中', 1)
('也称', 1)
('称作', 1)
('作开', 1)
('开发', 1)
('发中', 1)
('欠发', 1)
('指经', 1)
('经济', 1)
```



THE END