### **汉字标准及字符编码探究**

**前言：**

本文档主要写了“漢字”在各国家及地区的使用标准的制定历程 及 汉字在计算机中的数字编码方案制定历程，便于人们了解汉字在各地区的实际使用差异 以及计算机是如何处理汉字的输入及显示。一共分为4个部分：用字标准（字形），计算机汉字编码方案，计算机汉字输入方案，计算机汉字显示字形（字体）

**版权声明**：

本文档以开源的形式发布，所有条款如下：

1. 无担保：作者不保证文档内容的准确无误，亦不承担由于使用此文档所导致的任何后果

2. 自由使用：任何人可以出于任何目的而自由地 阅读/链接/打印/转载/引用/再创作 此文档，无需任何附加条件

若您 阅读/链接/打印/转载/引用/再创作 本文档，则说明接受以上2个条款。

作者：李茂福

更新日期：2022年8月24日

**第1章、汉字标准规范历程**

随着近代西方列强的崛起和清朝的衰弱，中国经历了上百年的屈辱挨打史，汉字文化的影响力也日渐式微。二战结束后，原来汉字文化圈的成员们（比如朝鲜，韩国，日本，越南等）便失去了对汉字的崇拜，同时也为了实现民族独立，为了民族自尊，实现文言一致，纷纷去汉字化，使用各自的表音文字。

越南就使用了拉丁字母及其变体作为拼写文字；朝鲜及韩国使用1444年创制的谚文（韩语拼音文字）作为官方文字；日本也规范了假名（假名是日语拼音文字）的使用及常用汉字的使用。除了日本，其他几个国家用拼音文字还是比较顺利的，勉强能用。但由于日语的特殊性（它的音节少，同音词非常多）而只有汉字能有效地区分语义，所以日本仍保留了汉字。

于是目前仍使用汉字作为官方文字的地区就只剩下：中国（大陆、港、澳、台）和日本。（一些南洋国家也把汉语汉字作为官方语言文字之一，这里不详谈，只作附录列出）

再加上汉字书写系统的臃肿及各种思想上的因素，闲下来的人们开始对汉字动刀了。为什么要制定汉字使用标准呢？原因如下：

①汉字的异体字特别多，比如“回”字有四种写法（回囘囬𡇌），异体字的话它只是形体不同，意思和用法是一样的，所以为了简便，就规定只能用其中一种写法，规范异体字。

②当时部分人有汉字误国的认知，把国家的落后原因扣在汉字头上，于是想实现汉字拼音化，即最终实现使用拉丁/拼音字母去拼写汉语。

③汉字本身笔划多不方便书写，首先就是想简化一下汉字的部件（听说可以提高识字率）

④汉字数量非常多，而当时（二战后）的打字机还是机械打字机，不方便打汉字，汉字数量多，排版时间长，所以跟拼音字母比，汉字在印刷界使用比较困难。

⑤刚实现当家做主的人们，誓与旧时代划清界线，不使用旧的字形

**★日本**

最开始是日本方面在1946年发布了《當用漢字表》，收录1850个汉字，告诉人们以后就尽量只用这么多个汉字，其他的用日文表示。

|  |  |  |
| --- | --- | --- |
| 1946年 | 現代假名遣 | 《現代假名遣》规范了假名及音训读的标准 |
| 1946年 | 當用漢字表 | 国語審議会公布了《當用漢字表》，收录1850汉字 |
| 1981年10月1日 | 常用漢字表 | 内阁告示第1号发布《常用漢字表》收1945字为常用字，2187音读，1900训读（共4087） |
| 2010年6月7日 | 改定常用漢字表 | 日本文化審議会发布《改定常用漢字表》2136字，（2352音读，2036训读） |
| 2010年11月30日 |  | 在《常用漢字表》基础上追加196常用字，删除5字，共2136字，以平成22年内阁告示第2号《常用漢字表》发布 |
| 2016年 |  | 文化審議会国語分科颁布《关于常用汉字表字体、字形的指导方针（报告）》 |

**★中国大陆**

|  |  |  |
| --- | --- | --- |
| 1955年10月 | 第一批异体字整理草案 | 在北京召开全国文字改革会议，通过《第一批异体字整理草案》，12月文化部、文改会联合发布此草案，要求1956年2月起在全国实施。该表收异体字810组，选出810个正体字，淘汰1055个异体字 |
| 1964年3月7日 | 简化字总表 |  |
| 1965年1月 | 印刷通用汉字字形表 | 6196字 |
| 1977年 | 二简字 | 《第二次汉字简化方案（草案）》收853字  1977年12月20日～1986年6月24日 废除 |
| 1986年 | 废除二简字 | 重发布《简化字总表》1986年10月10日 |
| 1988年1月26日 | 现代汉语常用字表 | 制订《现代汉语常用字表》共3500字  2500常用字，1000次常用字 |
| 1988年3月25日 | 现代汉语通用字表 | 《现代汉语通用字表》收7000字 |
| 1997年4月7日 | 现代汉语通用字笔顺规范 | 《现代汉语通用字笔顺规范》，7000字 |
| 2013年 | 通用规范汉字表 | 为了落实2000年的国家通用语言文字法，发布了《通用规范汉字表》8105字  3500常用字，3000二级字表，1605三级字表 |
|  |  |  |

**★中国台湾**

|  |  |  |
| --- | --- | --- |
| 1979年 |  | 常用國字標準字體表（简称甲表）4808字 |
| 1982年12月 |  | 次常用國字標準字體表（简称乙表）6341字 |
| 1983年10月 |  | 罕用國字標準字體表（简称丙表）18388字 |
| 1984年 |  | 異體國字字表（简称丁表）18588字，補遺22字  于2004年修订版收70833字 |
| 1995年 |  | 常用國字標準字體手册 |

**★中国香港**

|  |  |  |
| --- | --- | --- |
| 1984年7月 |  | 香港教育署语言教育学院中文系的李学铭教授同名院校的学者组成「常用字標準字形研究委員會」 |
| 1986年9月 | 字形表 | 1985年9月完成《字形表》，于1986年9月出版，收4721字。之后进行多次修订 |
| 1988年 |  | 《字形表》公布后，香港教育署在1988年分页的xxx以字形表为依据 |
| 2002年 | 参考2000版字形表 | 中咨会发布《香港电脑汉字楷体字形参考指引》和  《香港电脑汉字主体（印刷体）字形参考指引》 |
| 2016年 | 参照2007版字形表 | 中咨会公布《香港电脑汉字参考字形》参照2007版字形表 |

《字形表》版本 收字

1986 4721

1993修订本 4759

1997修订本 4759

2000修订本 4759 主要参考此版本，影响大

2007重排本 4762 附录于《香港小学学习字词表》

2012重排本 4762

**★港台繁体字字形差异**

在

①用字不同

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 香港 | 台湾 |  | 香港 | 台湾 |  |
| 裏 | 裡 |  | 羣 | 群 |  |
| 峯 | 峰 |  |  |  |  |
| 宂 | 冗 |  | 牀 | 床 |  |
| 吿 | 告 |  | 台 | 臺 | 香港的“台臺”合并 |
| 卧 | 臥 |  |  |  |  |
| 着著 | 著 | 台湾的“着著”合并 | 沉 |  |  |
| 污 | 汙 |  |  |  |  |

②字形略微不同

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 香港 | 台湾 |  | 香港 | 台湾 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**★中日简体字异同**

在

**第2章、计算机字符集及编码**

严格来讲字符集和编码是2个不同的概念，字符集是字符的集合，表示这个集合收录了多少个字符，而编码是使用数字去唯一地表示这个字符集里的每个字符。

一般，一个字符集只使用一套编码方案，而有的字符集可以使用多种编码方案。

**★ASCII码**

1961年美国国家标准局(ANSI)制定了ASCII码（American Standard Code for Information Interchange，美国标准信息[交换码](http://www.baike.com/sowiki/%E4%BA%A4%E6%8D%A2%E7%A0%81?prd=content_doc_search" \o "交换码)），它已被国际标准化组织（ISO）定为国际标准，称为ISO 646标准，标准ASCII码为7位，扩充为8位（最高位为0）

7位二进制数可以表示(2^7)==128个数字（0到127），每个数字都唯一地对应一个字符，这些数字就是对应字符的编码，ASCII编码对应 控制字符及可打印字符：

0～32号及第127号（共34个）是控制字符或通讯专用字符（32号为空格）

33～126号（共94个）是可打印字符

ASCII码主要是给美国用的（英语文字），欧洲及亚洲其他国家的文字符号不在ACSII的收录范围内，于是他们把ASCII扩展了，使用8位的ASCII，把最高位置1，这样比原来的ASCII又多出128个编码可分配，不同的国家甚至厂商都有自己的标准，不利于规范，于是国际标准化组织（ISO）和国际电工委员会(IEC)联合制定了一系列8位字符集的标准：ISO 8859，全称ISO/IEC 8859

**ISO/IEC 8859-1** (Latin-1) 西欧语言

**ISO/IEC 8859-2** (Latin-2) 中欧语言

**ISO/IEC 8859-3** (Latin-3) 南欧语言。世界语也可用此字符集显示。

**ISO/IEC 8859-4** (Latin-4) 北欧语言

**ISO/IEC 8859-5** (Cyrillic) 斯拉夫语言

**ISO/IEC 8859-6** (Arabic) 阿拉伯语

**ISO/IEC 8859-7** (Greek) 希腊语

**ISO/IEC 8859-8** (Hebrew) 希伯来语(视觉顺序)

**ISO/IEC 8859-8-I** (Hebrew) 希伯来语(逻辑顺序)

**ISO/IEC 8859-9** (Latin-5 或 Turkish) 它把Latin-1的冰岛语字母换走，加入土耳其语字母

**ISO/IEC 8859-10** (Latin-6 或 Nordic) 北日耳曼语支，用来代替Latin-4

**ISO/IEC 8859-11** (Thai) 泰语，从泰国的 TIS620 标准字集演化而来

**ISO/IEC 8859-13** (Latin-7 或 Baltic Rim) 波罗的语族

**ISO/IEC 8859-14** (Latin-8 或 Celtic) 凯尔特语族

**ISO/IEC 8859-15** (Latin-9) 西欧语言，加入Latin-1欠缺的芬兰语字母和大写法语重音字母，以及欧元符号

**ISO/IEC 8859-16** (Latin-10) 东南欧语言。主要供罗马尼亚语使用，并加入欧元符号

**ASCII码表**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数字 | Hex | 含义 | 数字 | Hex | 含义 | 数字 | Hex | 含义 | 数字 | Hex | 含义 | 数字 | Hex | 含义 |
| 0 | 00 | NUL | 26 | 1A | SUB | 52 | 34 | 4 | 78 | 4E | N | 104 | 68 | h |
| 1 | 01 | SOH | 27 | 1B | ESC | 53 | 35 | 5 | 79 | 4F | O | 105 | 69 | i |
| 2 | 02 | STX | 28 | 1C | FS | 54 | 36 | 6 | 80 | 50 | P | 106 | 6A | j |
| 3 | 03 | ETX | 29 | 1D | GS | 55 | 37 | 7 | 81 | 51 | Q | 107 | 6B | k |
| 4 | 04 | EOT | 30 | 1E | RS | 56 | 38 | 8 | 82 | 52 | R | 108 | 6C | l |
| 5 | 05 | ENQ | 31 | 1F | US | 57 | 39 | 9 | 83 | 53 | S | 109 | 6D | m |
| 6 | 06 | ACK | 32 | 20 | 空格 | 58 | 3A | : | 84 | 54 | T | 110 | 6E | n |
| 7 | 07 | BEL | 33 | 21 | ! | 59 | 3B | ; | 85 | 55 | U | 111 | 6F | o |
| 8 | 08 | BS | 34 | 22 | " | 60 | 3C | < | 86 | 56 | V | 112 | 70 | p |
| 9 | 09 | HT | 35 | 23 | # | 61 | 3D | = | 87 | 57 | W | 113 | 71 | q |
| 10 | 0A | LF | 36 | 24 | $ | 62 | 3E | > | 88 | 58 | X | 114 | 72 | r |
| 11 | 0B | VT | 37 | 25 | % | 63 | 3F | ? | 89 | 59 | Y | 115 | 73 | s |
| 12 | 0C | FF | 38 | 26 | & | 64 | 40 | @ | 90 | 5A | Z | 116 | 74 | t |
| 13 | 0D | CR | 39 | 27 | ' | 65 | 41 | A | 91 | 5B | [ | 117 | 75 | u |
| 14 | 0E | S0 | 40 | 28 | ( | 66 | 42 | B | 92 | 5C | \ | 118 | 76 | v |
| 15 | 0F | S1 | 41 | 29 | ) | 67 | 43 | C | 93 | 5D | ] | 119 | 77 | w |
| 16 | 10 | DLE | 42 | 2A | \* | 68 | 44 | D | 94 | 5E | ^ | 120 | 78 | x |
| 17 | 11 | DC1 | 43 | 2B | + | 69 | 45 | E | 95 | 5F | \_ | 121 | 79 | y |
| 18 | 12 | DC2 | 44 | 2C | , | 70 | 46 | F | 96 | 60 | ` | 122 | 7A | z |
| 19 | 13 | DC3 | 45 | 2D | - | 71 | 47 | G | 97 | 61 | a | 123 | 7B | { |
| 20 | 14 | DC4 | 46 | 2E | . | 72 | 48 | H | 98 | 62 | b | 124 | 7C | | |
| 21 | 15 | NAK | 47 | 2F | / | 73 | 49 | I | 99 | 63 | c | 125 | 7D | } |
| 22 | 16 | SYN | 48 | 30 | 0 | 74 | 4A | J | 100 | 64 | d | 126 | 7E | ~ |
| 23 | 17 | ETB | 49 | 31 | 1 | 75 | 4B | K | 101 | 65 | e | 127 | 7F | DEL |
| 24 | 18 | CAN | 50 | 32 | 2 | 76 | 4C | L | 102 | 66 | f |  |  |  |
| 25 | 19 | EM | 51 | 33 | 3 | 77 | 4D | M | 103 | 67 | g |  |  |  |

\t 09 HT

\r 0D CR

\n 0A LF

EOF 1A SUB

**★JIS C 6220**

日本二战后，使用计算机也比较早，他们也想搞一套日文的编码标准，但仅是日语假名（日语拼音文字，有平假名和片假名2种）的数量就已经超过128个了，还不包括汉字。用一个字节（8比特）恐怕是不够编码的了，但当时没办法，为了和ASCII兼容，仍只用一个字节去编码，所以他们只给片假名进行编码，于1969年发布了片假名编码方案JIS C 6220（后来称为jis x 0201），这套编码对应的是半角的字形，即同英文字母一样只占半个宽度。

JIS X 0201标准内的半角标点及片假名（0xA1-0xDF）

这套编码方案用了几年，甚是不方便（前面说过，日语若不使用汉字，同音词非常多，容易造成歧义），最终还是不得不制定新的标准，为了能表示所有的日语假名及常用汉字，得用2个字节去表示，因为2字节能表示十进制的0至65535，编码数量上是够用了（常用汉字不过三五千字）

**★Shift\_JIS**

1978年日本基于ISO 2022制订了最早的汉字编码JIS C 6226（现称jis x 0208）

jis c 6226也称为Shift\_JIS，能容纳全角及半角拉丁字母、平假名、片假名、符号及日语汉字。

它被命名为Shift\_JIS的原因，是因为它在放置全角字符时，要避开原本在0xA1-0xDF放置的半角假名字符（即之前的jis c 6220）。后来广义的shift\_jis包含了jis c 6220及jis c 6226和增加的388个汉字及符号

在微软及IBM的日语电脑系统中，即使用了这个编码表，这个编码表称为CP932

**字节结构编辑：**

以下字符在Shift\_JIS使用一个字节来表示：

ASCII字符(0x00-0x7F)，但“\”被“¥”取代

JIS X 0201标准内的半角标点及片假名（0xA1-0xDF）

在部分操作系统中，0xA0用来放置“不换行空格”

以下字符在Shift\_JIS使用两个字节来表示：

JIS X 0208 字集的所有字符

“第一个字节”使用0x81-0x9F、0xE0-0xEF（共47个）

“第二个字节”使用0x40-0x7E、0x80-0xFC（共188个）

**★GB 2312**

为了满足国内在计算机中使用汉字的需要，中国国家标准总局发布了一系列的汉字[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)国家标准编码，统称为GB码（国标码）。其中最有影响的是于1980年发布的《信息交换用汉字编码[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)基本集》，标准号为GB 2312-1980，因其使用非常普遍，也常被通称为国标码，GB2312编码通行于我国内地及新加坡。

GB 2312是一个[简体中文](https://baike.so.com/doc/5991655-6204624.html" \t "https://baike.so.com/doc/_blank)[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)，由6763个常用汉字和682个[全角](https://baike.so.com/doc/5379132-5615364.html" \t "https://baike.so.com/doc/_blank)的非汉字字符组成。其中汉字根据使用的频率分为两级。一级汉字3755个，二级汉字3008个。由于字符数量比较大，GB2312采用了二维矩阵编码法对所有字符进行编码。首先构造一个94行94列的方阵，对每一行称为一个"区"，每一列称为一个"位"，然后将所有字符按照分组填写到方阵中。这样所有的字符在方阵中都有一个唯一的位置，这个位置可以用区号、位号合成表示，称为字符的区位码。如第一个汉字"啊"出现在第16区的第1位上，其区位码为1601。因为区位码同字符的位置是完全对应的，因此区位码同字符之间也是一一对应的。这样所有的字符都可通过其区位码转换为数字编码信息。

**GB 2312字符编码分布表**

|  |  |
| --- | --- |
| 分区范围 | 符号类型 |
| 第01区 | 中文标点、数学符号以及一些特殊字符 |
| 第02区 | 各种各样的数学序号 |
| 第03区 | 全角西文字符 |
| 第04区 | 日文平假名 |
| 第05区 | 日文片假名 |
| 第06区 | 希腊字母表 |
| 第07区 | 俄文字母表 |
| 第08区 | 中文拼音字母表 |
| 第09区 | 制表符号 |
| 第10-15区 | 无字符 |
| 第16-55区 | 一级汉字(以拼音字母排序) |
| 第56-87区 | 二级汉字(以部首笔画排序) |
| 第88-94区 | 无字符 |

GB2312字符在计算机中存储是以其区位码为基础的，其中汉字的区码和位码分别占一个字节，每个汉字占两个字节。由于区码和位码的取值范围都是在1-94之间，这样的范围与ASCII码冲突。例如汉字'珀'在GB2312中的区位码为7174，其两[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)表示形式为71，74;而两个英文字符'GJ'的存储码也是71,74。这种冲突将导致在解释编码时到底表示的是一个汉字还是两个ASCII字符将无法判断。

为避免同ASCII编码发生冲突，GB2312字符在进行存储时，通过将原来的每个[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)第8bit（最高位）设置为1与ASCII区分开来，如果第8bit为0，则表示西文字符，否则表示GB2312中的字符。

实际存储时，采用了将区位码的每个[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)分别加上0xA0(160)的方法转换为存储码，计算机存储规则是此编码的补码，而且是位码在前，区码在后。例如汉字'啊'的区位码为1601，其存储码为0xB0A1，其转换过程为:

|  |  |  |  |
| --- | --- | --- | --- |
| 区位码 | 区码转换 | 位码转换 | 存储码 |
| 1001 | 10+A0=B0 | 01+A0=A1 | B0A1 |

**★GB13000**

1993年发布GB13000，国际标准Unicode 1.1版本推出，收录中国大陆、台湾、日本及韩国通用字符集的汉字，总共有20902个。

中国大陆制定了等同于Unicode 1.1版本的GB 13000.1-93，简称为GB13000.

**★GBK**

GB 2312只有简体汉字，没有繁体字，也没有较常用的人名地名等汉字，于是在1995年对GB 2312进行了扩展，称为GBK，字母K表示“扩”字的首字母

GBK即汉字内码扩展规范，英文全称Chinese Internal Code Specification。GBK编码标准兼容GB2312，共收录汉字21003个、符号883个，并提供1894个造字码位，简、繁体字融于一库。中华人民共和国全国信息技术标准化技术委员会1995年12月1日制订，国家技术监督局标准化司、电子工业部科技与质量监督司1995年12月15日联合以技监标函1995 229号文件的形式，将它确定为技术规范指导性文件。这一版的GBK规范为1.0版。GBK是对GB2312-80的扩展，也就是CP936字码表(Code Page 936)的扩展(之前CP936和GB 2312-80一模一样)。

[GBK](https://baike.so.com/doc/2389354-2526441.html" \t "https://baike.so.com/doc/_blank)采用双[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)表示，总体编码范围为8140-FEFE，首字节在81-FE之间，尾字节在40-FE之间，剔除xx7F一条线。总计23940个码位，共收入21886个汉字和图形符号，其中汉字(包括部首和构件)21003个，图形符号883 个。P-Windows3.2和苹果OS以GB2312为基本汉字编码，Windows 95/98则以GBK为基本汉字编码。

**GBK编码区分三部分：**

**汉字区**包括

GBK/2：OXBOA1-F7FE, 收录GB2312汉字6763个，按原序排列

GBK/3：OX8140-AOFE，收录CJK汉字6080个

GBK/4：OXAA40-FEAO，收录CJK汉字和增补的汉字8160个

**图形符号区**包括

GBK/1：OXA1A1-A9FE，除GB2312的符号外，还增补了其它符号

GBK/5：OXA840-A9AO，扩除非汉字区

**用户自定义区**

GBK区域中的空白区，用户可以自己定义字符

**★GB18030**

国家标准GB18030-2005《信息技术中文编码字符集》是我国继GB2312-1980和GB13000.1-1993之后最重要的[汉字编码](https://baike.so.com/doc/4239801-4441897.html" \t "https://baike.so.com/doc/_blank)标准，是我国计算机系统必须遵循的基础性标准之一。

GB18030有两个版本：

GB18030-2000和GB18030-2005。GB18030-2000是GBK的取代版本，它的主要特点是在GBK基础上增加了CJK统一汉字扩充A的汉字。GB18030-2005的主要特点是在GB18030-2000基础上增加了CJK统一汉字扩充B的汉字。

GB18030-2000编码标准是由信息产业部和国家质量技术监督局在2000年3月17日联合发布的，并且将作为一项国家标准在2001年的1月正式强制执行。

GB18030-2005《信息技术中文编码[字符](https://baike.so.com/doc/2529629-2672477.html" \t "https://baike.so.com/doc/_blank)集》是我国自主研制的以汉字为主并包含多种我国少数民族文字(如藏、蒙古、傣、彝、朝鲜、维吾尔文等)的超大型中文编码字符集强制性标准，其中收入汉字70000余个。

GB18030最新版本是GB18030-2005。GB18030-2005与GB18030-2000的编码体系结构是完全相同的。那么，GB18030的2000版和2005版有什么区别和联系呢?

2000年发布的GB18030-2000，全名是《信息技术 汉字编码[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank) 基本集的扩充》。GB18030-2000仅规定了常用非汉字符号和27533个汉字(包括部首、部件等)的编码。

GB18030-2000是全文强制性标准，市场上销售的产品必须符合。2005年发布的GB18030-2005在GB18030-2000的基础上增加了42711个汉字和多种我国少数民族文字的编码，增加的这些内容是推荐性的。原GB18030-2000中的内容是强制性的，市场上销售的产品必须符合。故GB18030-2005为部分强制性标准，自发布之日起代替GB18030-2000。GB18030-2005的单字节编码部分、双字节编码部分和四字节编码部分的CJK统一汉字扩充A(即0x8139EE39-0x82358738)部分为强制性。

GB18030在windows中的代码页是CP54936。

标准采用单[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)、双字节和四字节三种方式对[字符编码](https://baike.so.com/doc/6261384-6474804.html" \t "https://baike.so.com/doc/_blank)。单字节部分采用GB/T 11383的编码结构与规则，使用0×00至0×7F码位(对应于ASCII码的相应码位)。双字节部分，首字节码位从0×81至0×FE，尾字节码位分别是0×40至0×7E和0×80至0×FE。四字节部分采用GB/T 11383未采用的0×30到0×39作为对双字节编码扩充的后缀，这样扩充的四字节编码，其范围为0×81308130到0×FE39FE39。其中第一、三个字节编码码位均为0×81至0×FE，第二、四个字节编码码位均为0×30至0×39。

**GB18030-2000汉字**

如下表所示，GB18030-2000收录了27533个汉字:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 码位范围 | 码位数 | 字符数 | 字符类型 |
| 双字节部分 | 第一字节0xB0-0xF7 | 6768 | 6763 | 汉字 |
| 第二字节0xA1-0xFE |
| 第一字节0x81-0xA0 | 6080 | 6080 | 汉字 |
| 第二字节0x40-0xFE |
| 第一字节0xAA-0xFE | 8160 | 8160 | 汉字 |
| 第二字节0x40-0xA0 |
| 四字节部分 | 第一字节0x81-0x82 | 25200 | 6530 | CJK统一汉字扩充A |
| 第二字节0x30-0x39 |
| 第三字节0x81-0xFE |
| 第四字节0x30-0x39 |

27533就是6763+6080+8160+6530。双字节部分的6763+6080+8160=21003个汉字就是GBK的21003个汉字。

在Unicode中，CJK统一汉字扩充A有6582个汉字，为什么这里只有6530个汉字?

这是因为在GBK时代，双字节部分已经收录过CJK统一汉字扩充A的52个汉字，所以还余6530个汉字

****GB18030-2005汉字****

如下表所示，GB18030-2005收录了70244个汉字:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 码位范围 | 码位数 | 字符数 | 字符类型 |
| 双字节部分 | 第一字节0xB0-0xF7 | 6768 | 6763 | 汉字 |
| 第二字节0xA1-0xFE |
| 第一字节0x81-0xA0 | 6080 | 6080 | 汉字 |
| 第二字节0x40-0xFE |
| 第一字节0xAA-0xFE | 8160 | 8160 | 汉字 |
| 第二字节0x40-0xA0 |
| 四字节部分 | 第一字节0x81-0x82 | 25200 | 6530 | CJK统一汉字扩充A |
| 第二字节0x30-0x39 |
| 第三字节0x81-0xFE |
| 第四字节0x30-0x39 |
| 第一字节0x95-0x98 | 50400 | 42711 | CJK统一汉字扩充B |
| 第二字节0x30-0x39 |
| 第三字节0x81-0xFE |
| 第四字节0x30-0x39 |

收起

70244就是6763+6080+8160+6530+42711。

版本变化

GB18030-2005相对于GB18030-2000主要有以下变化:

1、在四字节字符表中增加CJK统一汉字扩充B和已经在GB13000中编码的我国少数民族文字字符的字形。其实GB18030-2000已经映射了这些码位，但GB18030-2000没有给出这些字符的字形。

2、调整字符?的编码。

3、去掉了单字节编码的欧元符号(0x80)。

(纠正:其实GBK和GB18030-2000都没有单字节编码的欧元符号，微软的CP936才有这个符号并定义在0x80

**★BIG5**

在台湾、香港与澳门地区，使用的是繁体字。而1980年发布的GB2312面向[简体中文](https://baike.so.com/doc/5991655-6204624.html" \t "https://baike.so.com/doc/_blank)[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)，并不支持繁体字。在这些使用繁体字的地区，一度出现过很多不同厂商提出的字符集编码，这些编码彼此互不兼容，造成了信息交流的困难。为统一繁体[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)编码，1984年，台湾五大厂商宏碁、神通、佳佳、零壹以及大众一同制定了一种繁体中文编码方案，因其来源被称为五大码，英文写作Big5，后来按英文翻译回汉字后，普遍被称为大五码。

大五码是一种繁体中文汉字[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)，其中繁体汉字13053个，808个标点符号、希腊字母及特殊符号。大五码的编码码表直接针对[存储](https://baike.so.com/doc/4223154-4424731.html" \t "https://baike.so.com/doc/_blank)而设计，每个字符统一使用两个[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)存储表示。第1[字节](https://baike.so.com/doc/1114609-1179328.html" \t "https://baike.so.com/doc/_blank)范围81H-FEH，避开了同ASCII码的冲突，第2字节范围是40H-7EH和A1H-FEH。因为Big5的字符编码范围同GB2312字符的存储码范围存在冲突，所以在同一正文不能对两种[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)的字符同时支持。

Big5编码的分布如表1-5所示，Big5字符主要部分集中在三个段内：

标点符号、希腊字母及特殊符号;常用汉字;非常用汉字。其余部分保留给其他厂商支持。

表1-5 Big5字符编码分布表

|  |  |
| --- | --- |
| 编码范围 | 符号类别 |
| 8140H-A0FEH | 保留(用作造字区) |
| A140H-A3BFH | 标点符号、希腊字母及特殊符号 |
| A3C0H-A3FEH | 保留(未开放用于造字区) |
| A440H-C67EH | 常用汉字(先按笔划，再按部首排序) |
| C6A1H-C8FEH | 保留(用作造字区) |
| C940H-F9D5H | 非常用汉字(先按笔划，再按部首排序) |
| F9D6H-FEFEH | 保留(用作造字区) |

Big5编码推出后，得到了繁体中文软件厂商的广泛支持，在使用繁体汉字的地区迅速普及使用。目前，Big5编码在台湾、香港、澳门及其他海外华人中普遍使用，成为了繁体中文编码的事实标准。

**★EUC-KR**

EUC-KR用来储存[韩国](http://www.baike.com/sowiki/%E9%9F%A9%E5%9B%BD?prd=content_doc_search" \o "韩国)KSX1001字集(旧称KSC5601)的[字符](http://www.baike.com/sowiki/%E5%AD%97%E7%AC%A6?prd=content_doc_search" \o "字符)。

此规格由KSX2901(旧称[KS C 5861](http://www.baike.com/wiki/javascript:linkredwin('KS C 5861');" \o "KS C 5861" \t "http://www.baike.com/wiki/))定义。   
KS X 1001字元使用两个字节来表示。   
[“高位字节”](http://www.baike.com/wiki/javascript:linkredwin('%E2%80%9C%E9%AB%98%E4%BD%8D%E5%AD%97%E8%8A%82%E2%80%9D');" \o "\“高位字节\”" \t "http://www.baike.com/wiki/) 使用0xA1-0xFE   
[“低位字节”](http://www.baike.com/wiki/javascript:linkredwin('%E2%80%9C%E4%BD%8E%E4%BD%8D%E5%AD%97%E8%8A%82%E2%80%9D');" \o "\“低位字节\”" \t "http://www.baike.com/wiki/) 使用0xA1-0xFE

**★Unicode**

传统的双字节编码方案都有地域性，即不同的国家或地区都使用自己的那套编码方案，虽然都兼容了ASCII，但各自之间互不兼容，比如日本使用shift\_jis，中国大陆使用GBK，他们编码相同的部分，表示的字符却不相同，如果不额外说明，则用户软件无法主动识别，会导致乱码。

Unicode（统一码、万国码）是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的编码，以满足跨语言、跨平台进行文本转换、处理的要求。1990年开始研发，1990年完成ISO 10646初版草案，1994年正式公布

Unicode是基于通用[字符集](https://baike.so.com/doc/201915-213495.html" \t "https://baike.so.com/doc/_blank)UCS（Universal Character Set）的标准来发展，并且同时也以书本的形式（The Unicode Standard）对外发表。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 时间 | 版本 | 汉字编码（新增部分） | 新增字数 | 累计字数 |
| 1993年6月 | 1.1 | U+4E00~9FFF |  | 20902字 |
| 1999 | 3.0 | 扩A 3400~4DFF | 6582 | 27484 |
| 2001 |  | 扩B 20000~2A6FF | 42711 | 70195 |
| 2009年10月 | 5.2 | 扩C 2A700~2B734 | 4149 | 74344 |
| 2010 | 6.0 | 扩D 2B740~2B81F | 222 | 74566 |
| 2015年6月 | 8.0 | 扩E 2B820~2CEAF | 5762 | 80328 |
| 2017年6月 | 10.0 | 扩F 2CEB0~2EBEF  cjk兼容表意compact F900~FAFF  cjk兼容表意扩充compact-Ext 2F800~2FA1F | 7473  474  542 | 87801 |
| 2020年3月 | 13.0 | 扩G 30000~3134A | 4939 | 92740 |

在unicode联盟网站上可以查看完整的核心规范。

Unicode定义了大到足以代表人类所有可读字符的字符集。

Unicode是一个字符集，可以有多种编码方案，如UTF-8，UTF-16，UTF-32等

**①编码方式**

Unicode是[国际组织](https://baike.so.com/doc/103802-109562.html" \t "https://baike.so.com/doc/_blank)制定的可以容纳世界上所有文字和符号的字符编码方案。Unicode用数字0x0~0x10FFFF来映射这些字符，最多可以容纳1114112个字符，或者说有1114112个码位。码位就是可以分配给字符的数字（相当于唯一的身份id）。UTF-8、UTF-16、UTF-32都是将数字转换到程序数据的编码方案。

通用字符集（Universal Character Set，UCS）是由ISO制定的ISO 10646标准所定义的标准字符集（或称ISO/IEC 10646）。UCS-2用两个字节编码，UCS-4用4个字节编码。

历史上存在两个独立的尝试创立单一字符集的组织，即国际标准化组织（ISO）和多语言软件制造商组成的统一码联盟。前者开发ISO/IEC 10646项目，后者开发的统一码项目。因此最初制定了不同的标准。

1991年前后，两个项目的参与者都认识到，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采用了与ISO 10646-1相同的[字库](https://baike.so.com/doc/263424-278849.html" \t "https://baike.so.com/doc/_blank)和字码；ISO也承诺，ISO 10646将不会替超出U+10FFFF的UCS-4编码赋值，以使得两者保持一致。两个项目仍都存在，并独立地公布各自的标准。但统一码联盟和ISO/IEC JTC1/SC2都同意保持两者标准的码表兼容，并紧密地共同调整任何未来的扩展。在发布的时候，Unicode一般都会采用有关字码最常见的字型，但ISO 10646一般都尽可能采用Century字型。

UCS-4根据最高位为0的最高字节分成2^7=128个group。每个group再根据次高字节分为256个平面（plane）。每个平面根据第3个字节分为256行（row），每行256个码位（cell）。group 0的平面0被称作BMP（Basic Multilingual Plane）。将UCS-4的BMP去掉前面的两个零字节就得到了UCS-2。

每个平面有2^16=65536个码位。Unicode计划使用了17个平面，一共有17\*65536=1114112个码位。在Unicode 5.0.0版本中，已定义的码位只有238605个，分布在平面0、平面1、平面2、平面14、平面15、平面16。其中平面15和平面16上只是定义了两个各占65534个码位的专用区（Private Use Area），分别是0xF0000-0xFFFFD和0x100000-0x10FFFD。所谓专用区，就是保留给大家放[自定义](https://baike.so.com/doc/6405601-6619265.html" \t "https://baike.so.com/doc/_blank)字符的区域，可以简写为PUA。

平面0也有一个专用区0xE000-0xF8FF，有6400个码位。平面0的0xD800-0xDFFF（共2048个码位）是一个被称作代理区（Surrogate）的特殊区域。代理区的目的是使用两个UTF-16字符表示BMP以外的字符。

如前所述在Unicode 5.0.0版本中，238605-65534\*2-6400-2048=99089。余下的99089个已定义码位分布在平面0、平面1、平面2和平面14上，它们对应着Unicode目前定义的99089个字符，其中包括71226个汉字。平面0、平面1、平面2和平面14上分别定义了52080、3419、43253和337个字符。平面2的43253个字符都是汉字。平面0上定义了27973个汉字。

**②实现方式**

在Unicode中，汉字“字”对应的数字是23383，我们有很多方式将数字23383表示成程序中的数据，包括：UTF-8、UTF-16、UTF-32。UTF是“UCS Transformation Format”的缩写，可以翻译成Unicode字符集转换格式，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

这里用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。UTF-8、UTF-16、UTF-32分别以BYTE、WORD、DWORD作为编码单位。“汉字”的UTF-8编码需要6个字节。“汉字”的UTF-16编码需要两个WORD，大小是4个字节。“汉字”的UTF-32编码需要两个DWORD，大小是8个字节。根据字节序的不同，UTF-16可以被实现为UTF-16LE或UTF-16BE，UTF-32可以被实现为UTF-32LE或UTF-32BE。下面介绍UTF-8、UTF-16、UTF-32、字节序和BOM。

****★UTF-8****

UTF-8以字节为单位对Unicode进行编码。从Unicode到UTF-8的编码方式如下：

|  |  |
| --- | --- |
| ****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。从上表可以看出，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也只有21位。

例1：“汉”字的Unicode编码是0x6C49。0x6C49在0x0800-0xFFFF之间，使用用3字节模板了：1110xxxx 10xxxxxx 10xxxxxx。将0x6C49写成二进制是：0110 1100 0100 1001， 用这个比特流依次代替模板中的x，得到：11100110 10110001 10001001，即E6 B1 89。

例2：Unicode编码0x20C30在0x010000-0x10FFFF之间，使用用4字节模板了：11110xxx 10xxxxxx 10xxxxxx 10xxxxxx。将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，用这个比特流依次代替模板中的x，得到：11110000 10100000 10110000 10110000，即F0 A0 B0 B0。

****★UTF-16****

UTF-16编码以16位无符号整数为单位。我们把Unicode编码记作U。编码规则如下：

如果U<0x10000，U的UTF-16编码就是U对应的16位无符号整数（为书写简便，下文将16位无符号整数记作WORD）。

如果U≥0x10000，我们先计算U'=U-0x10000，然后将U'写成二进制形式：yyyy yyyy yyxx xxxx xxxx，U的UTF-16编码（二进制）就是：110110yyyyyyyyyy 110111xxxxxxxxxx。

为什么U'可以被写成20个二进制位？Unicode的最大码位是0x10ffff，减去0x10000后，U'的最大值是0xfffff，所以肯定可以用20个[二进制](https://baike.so.com/doc/1420585-1501617.html" \t "https://baike.so.com/doc/_blank)位表示。例如：Unicode编码0x20C30，减去0x10000后，得到0x10C30，写成二进制是：0001 0000 1100 0011 0000。用前10位依次替代模板中的y，用后10位依次替代模板中的x，就得到：1101100001000011 1101110000110000，即0xD843 0xDC30。

按照上述规则，Unicode编码0x10000-0x10FFFF的UTF-16编码有两个WORD，第一个WORD的高6位是110110，第二个WORD的高6位是110111。可见，第一个WORD的取值范围（二进制）是11011000 00000000到11011011 11111111，即0xD800-0xDBFF。第二个WORD的取值范围（二进制）是11011100 00000000到11011111 11111111，即0xDC00-0xDFFF。

为了将一个WORD的UTF-16编码与两个WORD的UTF-16编码区分开来，Unicode编码的设计者将0xD800-0xDFFF保留下来，并称为代理区（Surrogate）：

|  |  |  |
| --- | --- | --- |
| D800－DB7F | High Surrogates | 高位替代 |
| DB80－DBFF | High Private Use Surrogates | 高位专用替代 |
| DC00－DFFF | Low Surrogates | 低位替代 |

高位替代就是指这个范围的码位是两个WORD的UTF-16编码的第一个WORD。低位替代就是指这个范围的码位是两个WORD的UTF-16编码的第二个WORD。那么，高位专用替代是什么意思？我们来解答这个问题，顺便看看怎么由UTF-16编码推导Unicode编码。

如果一个字符的UTF-16编码的第一个WORD在0xDB80到0xDBFF之间，那么它的Unicode编码在什么范围内？我们知道第二个WORD的取值范围是0xDC00-0xDFFF，所以这个字符的UTF-16编码范围应该是0xDB80 0xDC00到0xDBFF 0xDFFF。我们将这个范围写成二进制：

1101101110000000 11011100 00000000 - 1101101111111111 1101111111111111

按照编码的相反步骤，取出高低WORD的后10位，并拼在一起，得到

1110 0000 0000 0000 0000 - 1111 1111 1111 1111 1111即0xe0000-0xfffff，按照编码的相反步骤再加上0x10000，得到0xf0000-0x10ffff。这就是UTF-16编码的第一个WORD在0xdb80到0xdbff之间的Unicode编码范围，即平面15和平面16。因为Unicode标准将平面15和平面16都作为专用区，所以0xDB80到0xDBFF之间的保留码位被称作高位专用替代。

****★UTF-32****

UTF-32编码以32位无符号整数为单位。Unicode的UTF-32编码就是其对应的32位无符号整数。

*****字节序*****

根据字节序的不同，UTF-16可以被实现为UTF-16LE或UTF-16BE，UTF-32可以被实现为UTF-32LE或UTF-32BE。例如：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****Unicode编码**** | ****UTF-16LE**** | ****UTF-16BE**** | ****UTF32-LE**** | ****UTF32-BE**** |
| 0x006C49 | 49 6C | 6C 49 | 49 6C 00 00 | 00 00 6C 49 |
| 0x020C30 | 43 D8 30 DC | D8 43 DC 30 | 30 0C 02 00 | 00 02 0C 30 |

原本ISO 10646标准定义了一个32位元的编码形式，称作UCS-4，使用通用字符集（UCS）的每一个字符，会在0到十六进制的7FFFFFFF这样的字码空间中，被表示成一个的32位元的码值。

　　UCS-4足以用来表示所有的Unicode的字码空间，其最大的码位为十六进制的10FFFF，所以其空间约有百万个码位。有些人认为保留如此大的字码空间却只为了对应这很小的码集是浪费的。所以一个新的编码UTF-32被提出来了。UTF-32 是一个 UCS-4 的子集，使用32-位元的码值，只在0到10FFFF的字码空间。

　　UTF-32 原本是 UCS-4 的子集，但JTC1/SC2/WG2声明，所有未来对字符的指定都将会限制在BMP及其14个补充平面，并移除先前在 E0 到 FF 平面的 60 到 7F 群的私用空间。

　　于是就现状而言，除了 UTF-32 标准包含额外的 Unicode 意涵，UCS-4 和 UTF-32 大体是相同的。

**★UTF-16代理对**

在

**★Unicode基础分区**

在

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 十进制 | | 十六进制 | | 字符数量 | 说明 |  |
| 起始 | 终止 | 起始 | 终止 | (个) |  |  |
| 0 | 127 | 0 | 007F | 128 | C0控制符及基本拉丁文 | C0 Control and Basic Latin |
| 128 | 255 | 80 | 00FF | 128 | C1控制符及拉丁文补充-1 | C1 Control and Latin 1 Supplement |
| 256 | 383 | 100 | 017F | 128 | 拉丁文扩展-A | Latin Extended-A |
| 384 | 591 | 180 | 024F | 208 | 拉丁文扩展-B | Latin Extended-B |
| 592 | 687 | 250 | 02AF | 96 | 国际音标扩展 | IPA Extensions |
| 688 | 767 | 02B0 | 02FF | 80 | 空白修饰字母 | Spacing Modifiers |
| 768 | 879 | 300 | 036F | 112 | 结合用读音符号 | Combining Diacritics Marks |
| 880 | 1023 | 370 | 03FF | 144 | 希腊文及科普特文 | Greek and Coptic |
| 1024 | 1279 | 400 | 04FF | 256 | 西里尔字母 | Cyrillic |
| 1280 | 1327 | 500 | 052F | 48 | 西里尔字母补充 | Cyrillic Supplement |
| 1328 | 1423 | 530 | 058F | 96 | 亚美尼亚语 | Armenian |
| 1424 | 1535 | 590 | 05FF | 112 | 希伯来文 | Hebrew |
| 1536 | 1791 | 600 | 06FF | 256 | 阿拉伯文 | Arabic |
| 1792 | 1871 | 700 | 074F | 80 | 叙利亚文 | Syriac |
| 1872 | 1919 | 750 | 077F | 48 | 阿拉伯文补充 | Arabic Supplement |
| 1920 | 1983 | 780 | 07BF | 64 | 马尔代夫语 | Thaana |
| 1984 | 2047 | 07C0 | 07FF | 64 | 西非書面語言 | N'Ko |
| 2048 | 2143 | 800 | 085F | 96 | 阿维斯塔语及巴列维语 | Avestan and Pahlavi |
| 2144 | 2175 | 860 | 087F | 32 | Mandaic | Mandaic |
| 2176 | 2223 | 880 | 08AF | 48 | 撒马利亚语 | Samaritan |
| 2304 | 2431 | 900 | 097F | 128 | 天城文书 | Devanagari |
| 2432 | 2559 | 980 | 09FF | 128 | 孟加拉语 | Bengali |
| 2560 | 2687 | 0A00 | 0A7F | 128 | 锡克教文 | Gurmukhi |
| 2688 | 2815 | 0A80 | 0AFF | 128 | 古吉拉特文 | Gujarati |
| 2816 | 2943 | 0B00 | 0B7F | 128 | 奥里亚文 | Oriya |
| 2944 | 3071 | 0B80 | 0BFF | 128 | 泰米尔文 | Tamil |
| 3072 | 3199 | 0C00 | 0C7F | 128 | 泰卢固文 | Telugu |
| 3200 | 3327 | 0C80 | 0CFF | 128 | 卡纳达文 | Kannada |
| 3328 | 3455 | 0D00 | 0D7F | 128 | 德拉维族语 | Malayalam |
| 3456 | 3583 | 0D80 | 0DFF | 128 | 僧伽罗语 | Sinhala |
| 3584 | 3711 | 0E/00 | 0E7F | 128 | 泰文 | Thai |
| 3712 | 3839 | 0E/80 | 0EFF | 128 | 老挝文 | Lao |
| 3840 | 4095 | 0F00 | 0FFF | 256 | 藏文 | Tibetan |
| 4096 | 4255 | 1000 | 109F | 160 | 缅甸语 | Myanmar |
| 4256 | 4351 | 10A0 | 10FF | 96 | 格鲁吉亚语 | Georgian |
| 4352 | 4607 | 1100 | 11FF | 256 | 朝鲜文 | Hangul Jamo |
| 4608 | 4991 | 1200 | 137F | 384 | 埃塞俄比亚语 | Ethiopic |
| 4992 | 5023 | 1380 | 139F | 32 | 埃塞俄比亚语补充 | Ethiopic Supplement |
| 5024 | 5119 | 13A0 | 13FF | 96 | 切罗基语 | Cherokee |
| 5120 | 5759 | 1400 | 167F | 640 | 统一加拿大土著语音节 | Unified Canadian Aboriginal Syllabics |
| 5760 | 5791 | 1680 | 169F | 32 | 欧甘字母 | Ogham |
| 5792 | 5887 | 16A0 | 16FF | 96 | 如尼文 | Runic |
| 5888 | 5919 | 1700 | 171F | 32 | 塔加拉语 | Tagalog |
| 5920 | 5951 | 1720 | 173F | 32 | Hanunóo | Hanunóo |
| 5952 | 5983 | 1740 | 175F | 32 | Buhid | Buhid |
| 5984 | 6015 | 1760 | 177F | 32 | Tagbanwa | Tagbanwa |
| 6016 | 6143 | 1780 | 17FF | 128 | 高棉语 | Khmer |
| 6144 | 6319 | 1800 | 18AF | 176 | 蒙古文 | Mongolian |
| 6320 | 6399 | 18B0 | 18FF | 80 | Cham | Cham |
| 6400 | 6479 | 1900 | 194F | 80 | Limbu | Limbu |
| 6480 | 6527 | 1950 | 197F | 48 | 德宏泰语 | Tai Le |
| 6528 | 6623 | 1980 | 19DF | 96 | 新傣仂语 | New Tai Lue |
| 6624 | 6655 | 19/E0 | 19FF | 32 | 高棉语记号 | Kmer Symbols |
| 6656 | 6687 | 1A00 | 1A1F | 32 | Buginese | Buginese |
| 6688 | 6751 | 1A20 | 1A5F | 64 | Batak | Batak |
| 6784 | 6895 | 1A80 | 1AEF | 112 | Lanna | Lanna |
| 6912 | 7039 | 1B00 | 1B7F | 128 | 巴厘语 | Balinese |
| 7040 | 7088 | 1B80 | 1BB0 | 49 | 巽他语 | Sundanese |
| 7104 | 7167 | 1BC0 | 1BFF | 64 | Pahawh Hmong | Pahawh Hmong |
| 7168 | 7247 | 1C00 | 1C4F | 80 | 雷布查语 | Lepcha |
| 7248 | 7295 | 1C50 | 1C7F | 48 | Ol Chiki | Ol Chiki |
| 7296 | 7391 | 1C80 | 1CDF | 96 | 曼尼普尔语 | Meithei/Manipuri |
| 7424 | 7551 | 1D00 | 1D7F | 128 | 语音学扩展 | Phonetic Extensions |
| 7552 | 7615 | 1D80 | 1DBF | 64 | 语音学扩展补充 | Phonetic Extensions Supplement |
| 7616 | 7679 | 1DC0 | 1DFF | 64 | 结合用读音符号补充 | Combining Diacritics Marks Supplement |
| 7680 | 7935 | 1E/00 | 1EFF | 256 | 拉丁文扩充附加 | Latin Extended Additional |
| 7936 | 8191 | 1F00 | 1FFF | 256 | 希腊语扩充 | Greek Extended |
| 8192 | 8303 | 2000 | 206F | 112 | 常用标点 | General Punctuation |
| 8304 | 8351 | 2070 | 209F | 48 | 上标及下标 | Superscripts and Subscripts |
| 8352 | 8399 | 20A0 | 20CF | 48 | 货币符号 | Currency Symbols |
| 8400 | 8447 | 20D0 | 20FF | 48 | 组合用记号 | Combining Diacritics Marks for Symbols |
| 8448 | 8527 | 2100 | 214F | 80 | 字母式符号 | Letterlike Symbols |
| 8528 | 8591 | 2150 | 218F | 64 | 数字形式 | Number Form |
| 8592 | 8703 | 2190 | 21FF | 112 | 箭头 | Arrows |
| 8704 | 8959 | 2200 | 22FF | 256 | 数学运算符 | Mathematical Operator |
| 8960 | 9215 | 2300 | 23FF | 256 | 杂项工业符号 | Miscellaneous Technical |
| 9216 | 9279 | 2400 | 243F | 64 | 控制图片 | Control Pictures |
| 9280 | 9311 | 2440 | 245F | 32 | 光学识别符 | Optical Character Recognition |
| 9312 | 9471 | 2460 | 24FF | 160 | 封闭式字母数字 | Enclosed Alphanumerics |
| 9472 | 9599 | 2500 | 257F | 128 | 制表符 | Box Drawing |
| 9600 | 9631 | 2580 | 259F | 32 | 方块元素 | Block Element |
| 9632 | 9727 | 25A0 | 25FF | 96 | 几何图形 | Geometric Shapes |
| 9728 | 9983 | 2600 | 26FF | 256 | 杂项符号 | Miscellaneous Symbols |
| 9984 | 10175 | 2700 | 27BF | 192 | 印刷符号 | Dingbats |
| 10176 | 10223 | 27C0 | 27EF | 48 | 杂项数学符号-A | Miscellaneous Mathematical Symbols-A |
| 10224 | 10239 | 27F0 | 27FF | 16 | 追加箭头-A | Supplemental Arrows-A |
| 10240 | 10495 | 2800 | 28FF | 256 | 盲文点字模型 | Braille Patterns |
| 10496 | 10623 | 2900 | 297F | 128 | 追加箭头-B | Supplemental Arrows-B |
| 10624 | 10751 | 2980 | 29FF | 128 | 杂项数学符号-B | Miscellaneous Mathematical Symbols-B |
| 10752 | 11007 | 2A00 | 2AFF | 256 | 追加数学运算符 | Supplemental Mathematical Operator |
| 11008 | 11263 | 2B00 | 2BFF | 256 | 杂项符号和箭头 | Miscellaneous Symbols and Arrows |
| 11264 | 11359 | 2C00 | 2C5F | 96 | 格拉哥里字母 | Glagolitic |
| 11360 | 11391 | 2C60 | 2C7F | 32 | 拉丁文扩展-C | Latin Extended-C |
| 11392 | 11519 | 2C80 | 2CFF | 128 | 古埃及语 | Coptic |
| 11520 | 11567 | 2D00 | 2D2F | 48 | 格鲁吉亚语补充 | Georgian Supplement |
| 11568 | 11647 | 2D30 | 2D7F | 80 | 提非纳文 | Tifinagh |
| 11648 | 11743 | 2D80 | 2DDF | 96 | 埃塞俄比亚语扩展 | Ethiopic Extended |
| 11776 | 11903 | 2DE0 | 2E7F | 128 | 追加标点 | Supplemental Punctuation |
| 11904 | 12031 | 2E/80 | 2EFF | 128 | CJK 部首补充 | CJK Radicals Supplement |
| 12032 | 12255 | 2F00 | 2FDF | 224 | 康熙字典部首 | Kangxi Radicals |
| 12272 | 12287 | 2FF0 | 2FFF | 16 | 表意文字描述符 | Ideographic Description Characters |
| 12288 | 12351 | 3000 | 303F | 64 | CJK 符号和标点 | CJK Symbols and Punctuation |
| 12352 | 12447 | 3040 | 309F | 96 | 日文平假名 | Hiragana |
| 12448 | 12543 | 30A0 | 30FF | 96 | 日文片假名 | Katakana |
| 12544 | 12591 | 3100 | 312F | 48 | 注音字母 | Bopomofo |
| 12592 | 12687 | 3130 | 318F | 96 | 朝鲜文兼容字母 | Hangul Compatibility Jamo |
| 12688 | 12703 | 3190 | 319F | 16 | 象形字注释标志 | Kanbun |
| 12704 | 12735 | 31A0 | 31BF | 32 | 注音字母扩展 | Bopomofo Extended |
| 12736 | 12783 | 31C0 | 31EF | 48 | CJK 笔画 | CJK Strokes |
| 12784 | 12799 | 31F0 | 31FF | 16 | 日文片假名语音扩展 | Katakana Phonetic Extensions |
| 12800 | 13055 | 3200 | 32FF | 256 | 封闭式 CJK 文字和月份 | Enclosed CJK Letters and Months |
| 13056 | 13311 | 3300 | 33FF | 256 | CJK 兼容 | CJK Compatibility |
| 13312 | 19903 | 3400 | 4DBF | 6592 | CJK 统一表意符号扩展 A | CJK Unified Ideographs Extension A |
| 19904 | 19967 | 4DC0 | 4DFF | 64 | 易经六十四卦符号 | Yijing Hexagrams Symbols |
| 19968 | 40895 | 4E/00 | 9FBF | 20928 | CJK 统一表意符号 | CJK Unified Ideographs |
| 40960 | 42127 | A000 | A48F | 1168 | 彝文音节 | Yi Syllables |
| 42128 | 42191 | A490 | A4CF | 64 | 彝文字根 | Yi Radicals |
| 42240 | 42527 | A500 | A61F | 288 | Vai | Vai |
| 42592 | 42751 | A660 | A6FF | 160 | 统一加拿大土著语音节补充 | Unified Canadian Aboriginal Syllabics Supplement |
| 42752 | 42783 | A700 | A71F | 32 | 声调修饰字母 | Modifier Tone Letters |
| 42784 | 43007 | A720 | A7FF | 224 | 拉丁文扩展-D | Latin Extended-D |
| 43008 | 43055 | A800 | A82F | 48 | Syloti Nagri | Syloti Nagri |
| 43072 | 43135 | A840 | A87F | 64 | 八思巴字 | Phags-pa |
| 43136 | 43231 | A880 | A8DF | 96 | Saurashtra | Saurashtra |
| 43264 | 43391 | A900 | A97F | 128 | 爪哇语 | Javanese |
| 43392 | 43487 | A980 | A9DF | 96 | Chakma | Chakma |
| 43520 | 43583 | AA00 | AA3F | 64 | Varang Kshiti | Varang Kshiti |
| 43584 | 43631 | AA40 | AA6F | 48 | Sorang Sompeng | Sorang Sompeng |
| 43648 | 43743 | AA80 | AADF | 96 | Newari | Newari |
| 43776 | 43871 | AB00 | AB5F | 96 | 越南傣语 | Vi?t Thái |
| 43904 | 43936 | AB80 | ABA0 | 33 | Kayah Li | Kayah Li |
| 44032 | 55215 | AC00 | D7AF | 11184 | 朝鲜文音节 | Hangul Syllables |
| 55296 | 56319 | D800 | DBFF | 1024 | High-half zone of UTF-16 | High-half zone of UTF-16 |
| 56320 | 57343 | DC00 | DFFF | 1024 | Low-half zone of UTF-16 | Low-half zone of UTF-16 |
| 57344 | 63743 | E000 | F8FF | 6400 | 自行使用區域 | Private Use Zone |
| 63744 | 64255 | F900 | FAFF | 512 | CJK 兼容象形文字 | CJK Compatibility Ideographs |
| 64256 | 64335 | FB00 | FB4F | 80 | 字母表達形式 | Alphabetic Presentation Form |
| 64336 | 65023 | FB50 | FDFF | 688 | 阿拉伯表達形式A | Arabic Presentation Form-A |
| 65024 | 65039 | FE00 | FE0F | 16 | 变量选择符 | Variation Selector |
| 65040 | 65055 | FE10 | FE1F | 16 | 竖排形式 | Vertical Forms |
| 65056 | 65071 | FE20 | FE2F | 16 | 组合用半符号 | Combining Half Marks |
| 65072 | 65103 | FE30 | FE4F | 32 | CJK 兼容形式 | CJK Compatibility Forms |
| 65104 | 65135 | FE50 | FE6F | 32 | 小型变体形式 | Small Form Variants |
| 65136 | 65279 | FE70 | FEFF | 144 | 阿拉伯表達形式B | Arabic Presentation Form-B |
| 65280 | 65519 | FF00 | FFEF | 240 | 半型及全型形式 | Halfwidth and Fullwidth Form |
| 65520 | 65535 | FFF0 | FFFF | 16 | 特殊 | Specials |

["cjk统一"] = { first = 0x4E00, last = 0x9FFF },

["拉丁补-1"] = { first = 0x0080, last = 0x00FF },

["拉丁扩a"] = { first = 0x0100, last = 0x017F },

["拉丁扩b"] = { first = 0x0180, last = 0x024F },

["私用区"] = { first = 0xE000, last = 0xF8FF },

["希腊"] = { first = 0x0370, last = 0x03FF },

["平仮名"] = { first = 0x3040, last = 0x309F },

["片仮名"] = { first = 0x30A0, last = 0x30FF },

["cjk部首补充"] = { first = 0x2E80, last = 0x2EFF },

["康熙字典部首"] = { first = 0x2F00, last = 0x2FDF },

["cjk标点符号"] = { first = 0x3000, last = 0x303F },

["cjk笔画"] = { first = 0x31C0, last = 0x31EF },

["注音"] = { first = 0x3100, last = 0x312F },

["朝鲜文兼容字母"] = { first = 0x3130, last = 0x318F },

["cjk兼容表意文字"] = { first = 0xF900, last = 0xFAFF },

["半角片仮名"] = { first = 0xFF61, last = 0xFF9F },

["扩A"] = { first = 0x3400, last = 0x4DBF },

["扩B"] = { first = 0x20000, last = 0x2A6DF },

["扩C"] = { first = 0x2A700, last = 0x2B73F },

["扩D"] = { first = 0x2B740, last = 0x2B81F },

["扩E"] = { first = 0x2B820, last = 0x2CEAF },

["扩F"] = { first = 0x2CEB0, last = 0x2EBEF },

["扩G"] = { first = 0x30000, last = 0x3134A },

["cjk兼容表意文字扩充"] = { first = 0x2F800, last = 0x2FA1F }

**★字节顺序标记BOM**

BOM（Byte Order Mark）字节顺序标记

在UCS编码中有一个叫做Zero Width No-Break Space（零宽无间断间隔）的字符，它的编码是FEFF。而FFFE在UCS中是不存在的字符，所以不应该出现在实际传输中。UCS规范建议在传输字节流前，先传输字符Zero Width No-Break Space。这样如果接收者收到FEFF时就表明这个字节流是Big-Endian的；如果收到FFFE，就表明这个字节流是Little-Endian的。因此字符"Zero Width No-Break Space"（零宽无间断间隔）又被称作BOM。

UTF-8不需要BOM来表明字节顺序，但可以用BOM来表明编码方式。字符"Zero Width No-Break Space"的UTF-8编码是EF BB BF。所以如果接收者收到以EF BB BF开头的字节流，就知道这是UTF-8编码了。Windows就是使用BOM来标记文本文件的编码方式的。

字符U+FEFF如果出现在字节流的开头，则用来标识该字节流的字节序，是高位在前还是低位在前。如果它出现在字节流的中间，则表示零宽度非换行空格的意义，用户看起来就是一个空格。从Unicode3.2开始，U+FEFF只能出现在字节流的开头，只能用于标识字节序，除此之外的用法已被舍弃。取而代之的是，使用U+2060来表达零宽度无断空白。

一般，文本编辑软件，在保存一个以UTF-8编码的文件时，会在文件开始的地方插入三个不可见的字符（0xEF 0xBB 0xBF，即BOM）。它是一串隐藏的字符，用于让文本编辑器识别这个文件是否以UTF-8编码。

**不同编码的字节顺序标记的表示：**

|  |  |
| --- | --- |
| 编码 | BOM表示(十六进制) |
| UTF-8 | EF BB BF |
| UTF-16（大端字节序） | FE FF |
| UTF-16（小端字节序） | FF FE |
| UTF-32（大端字节序） | 00 00 FE FF |
| UTF-32（小端字节序） | FF FE 00 00 |
| UTF-7 | 2B 2F 76和以下的一个字节： [ 38 | 39 | 2B | 2F ] |
| en:UTF-1 | F7 64 4C |
| en:UTF-EBCDIC | DD 73 66 73 |
| en: Standard Compression Scheme for Unicode | 0E FE FF |
| en:BOCU-1 | FB EE 28及可能跟随着FF |
| GB-18030 | 84 31 95 33 |

**★代码页code page**

d

**★字符集编码对比**

**双字节编码**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字符集 | 首字节（区） | 尾字节（位） | 码位 | 汉字数 |
| Shift\_JIS | A1~DF  81~9F  E0~EF | 无  40~7E  80~FC | 63 (jis x 0201)  47x188=8836 | 6600左右 |
| GB 2312 | A1~FE | A1~FE | 94x94=8836 | 6763 |
| BIG 5 | 81~FE | 40~7E  A1~FE | 126x157=19782 | 13053 |
| GBK | 81~FE | 40~7E  80~FE | 126x190=23940 | 21003 |

**多字节编码**

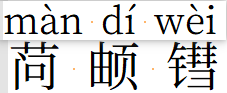
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编码 | 第1字节 | 第2字节 | 第3字节 | 第4字节 |
| GB 18030 | 81~FE | 30~39  40~7E  80~FE | 81~FE | 30~39 |
| UTF-8 | 00~7F  192~223  224~239  240~247 | （128）  128~191  128~191  128~191 | （2048）  128~191  128~191 | （65536）  128~191 |

**★通用规范汉字超出GBK范围**

2013年中国大陆发布的《通用规范汉字表》8105字中，有273字不在GBK字符集的范围内，且有些都已超出操作系统自带字体库的范围了，甚至常用输入法不支持输入这些通用规范汉字（取名字时尽量要在GBK的范围内）

**超出GBK的273个通用规范汉字：**

|  |  |  |
| --- | --- | --- |
| Unicode | 数量 | 汉字（有的操作系统字体不支持，这里以图片展示） |
| 扩A | 77 |  |
| 扩B | 36 |  |
| 扩C | 44 |  |
| 扩D | 8 |  |
| 扩E | 108 |  |

像这三个字（𬜬𬱖𬭬）是比较常见的通规字，常用于人姓名中，但都处于unicode扩E的分区里，常规输入法无法输入，大多数字体也不支持显示

**第3章、汉字的计算机输入方案**

在

**第4章、字形在计算机中的表示**

在

**★Windows系统各版本自带字体支持情况**

在

|  |  |  |
| --- | --- | --- |
| 版本 | 支持显示到 | 不支持显示 |
| Windows XP sp3 | cjk基础，cjk兼容表意19个 | 扩A及以上 |
| Windows 7 sp1 | cjk扩B | 扩C及以上 |
| Windows 8.1 | cjk扩D | 扩E及以上 |
| Windows 10 | cjk扩E部分 | 扩E全部及以上 |

**★字形错误**

在