编码集

-----软工11502班刘雅婷

摘要：通过查阅维基百科找到一些编码集如ASCLL码、Unicode码、Big5码、GB2312码、UTF-8/16码等的信息，例如这些编码的是因为何种原因产生的，是用来描述什么，是否可以用来描述中文，具有什么样的特性等等。

关键字：编码 特性 产生原因

目录

[1. ASCLL码 3](#_Toc492656939)

[1.1 ASCLL码的来历 3](#_Toc492656940)

[1.2 ASCLL码的特性 3](#_Toc492656941)

[1.3 ASCLL码的应用 4](#_Toc492656942)

[1.4 ASCLL码的缺点 4](#_Toc492656943)

[2. Unicode码 4](#_Toc492656944)

[2.1 Unicode码的来历 4](#_Toc492656945)

[2.2 Unicode码的特性 6](#_Toc492656946)

[2.2.1Unicode设计原则 6](#_Toc492656947)

[2.2.2Unicode编码方式 6](#_Toc492656948)

[2.2.3Unicode的输入方法 7](#_Toc492656949)

[2.3 Unicode码的应用 8](#_Toc492656950)

[3. Big5码 9](#_Toc492656951)

[3.1 Big5码的来历 9](#_Toc492656952)

[3.2 Big5码的特性 10](#_Toc492656953)

[3.3 Big5码的应用 11](#_Toc492656954)

[4. GB2312码 11](#_Toc492656955)

[4.1 GB2312码的概述 11](#_Toc492656956)

[4.2 GB2312码的特性 11](#_Toc492656957)

[4.3GB2312码的修订 13](#_Toc492656958)

[5. UTF-8码 14](#_Toc492656959)

[5.1 UTF-8码的来历 14](#_Toc492656960)

[5.2 UTF-8码的特性 14](#_Toc492656961)

[5.3 UTF-8的编码字节含义 15](#_Toc492656962)

[6. UTF-16码 17](#_Toc492656963)

[6.1 UTF-16码的来历 17](#_Toc492656964)

[6.2 UTF-16码的特性 17](#_Toc492656965)

[7. 各种编码间的转换 18](#_Toc492656966)

[8. 参考文献 18](#_Toc492656967)

[8.1 ASCLL码 18](#_Toc492656968)

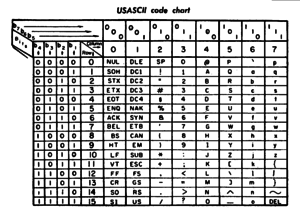
[8.2 Unicode码 19](#_Toc492656969)

[8.3 Big5码 19](#_Toc492656970)

# 1. ASCLL码

## 1.1 ASCLL码的来历

**ASCII**（发音： [/ˈæski/](https://zh.wikipedia.org/wiki/Help:%E8%8B%B1%E8%AA%9E%E5%9C%8B%E9%9A%9B%E9%9F%B3%E6%A8%99) [***ass****-kee*](https://zh.wikipedia.org/wiki/Wikipedia:%E7%99%BC%E9%9F%B3%E9%87%8D%E6%8B%BC)[[1]](https://zh.wikipedia.org/wiki/ASCII#cite_note-1)，**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange，**美国信息交换标准代码**）是基于[拉丁字母](https://zh.wikipedia.org/wiki/%E6%8B%89%E4%B8%81%E5%AD%97%E6%AF%8D)的一套[电脑](https://zh.wikipedia.org/wiki/%E7%94%B5%E8%84%91)[编码](https://zh.wikipedia.org/wiki/%E7%BC%96%E7%A0%81)系统。

[](https://zh.wikipedia.org/wiki/File:US-ASCII_code_chart.png)

1968年版ASCII编码速见表

ASCII第一次以规范标准的类型发表是在1967年，最后一次更新则是在1986年，至今为止共定义了128个字符；其中33个字符无法显示（一些终端提供了扩展，使得这些字符可显示为诸如笑脸、[扑克牌](https://zh.wikipedia.org/wiki/%E6%92%B2%E5%85%8B%E7%89%8C" \o "扑克牌)花式等8-bit符号），且这33个字符多数都已是陈废的[控制字符](https://zh.wikipedia.org/wiki/%E6%8E%A7%E5%88%B6%E5%AD%97%E5%85%83" \o "控制字符)。控制字符的用途主要是用来操控已经处理过的文字。在33个字符之外的是95个可显示的字符。用[键盘](https://zh.wikipedia.org/wiki/%E9%8D%B5%E7%9B%A4)敲下空白键所产生的空白字符也算1个可显示字符（显示为空白）。

## 1.2 ASCLL码的特性

ASCLL码分为控制字符和可显示字符。

说明：

[Unicode](https://zh.wikipedia.org/wiki/Unicode)表示法：当我们想在画面或纸张上表示这些控制字符时，就会显示成这个样子。过于老旧的系统或浏览器可能会看不到。使用微软任一中文输入法，输入`U2400即可看到␀，输入`U2401可看到␁，依此类推。

[脱出字符](https://zh.wikipedia.org/wiki/%E8%84%AB%E5%87%BA%E5%AD%97%E5%85%83)表示法：通常用于[终端机](https://zh.wikipedia.org/wiki/%E7%B5%82%E7%AB%AF%E6%A9%9F)连接（例如[Telnet](https://zh.wikipedia.org/wiki/Telnet)通信协议），以脱出字符^开头，再接一个符号，用来让这些控制字符得以在画面上显现。虽然看起来是两个字符，但在终端机上实际只有一个字符。在绝大部分的终端机系统中，包括[Windows](https://zh.wikipedia.org/wiki/Windows)的[命令提示字符](https://zh.wikipedia.org/wiki/%E5%91%BD%E4%BB%A4%E6%8F%90%E7%A4%BA%E5%AD%97%E5%85%83)（cmd.exe）、[Linux](https://zh.wikipedia.org/wiki/Linux)和[FreeBSD](https://zh.wikipedia.org/wiki/FreeBSD)，都可用[Ctrl](https://zh.wikipedia.org/wiki/%E6%8E%A7%E5%88%B6%E9%94%AE)代表脱出字符，输入想要的ASCII控制字符。例如想输入空字符，就要输入[Ctrl](https://zh.wikipedia.org/wiki/%E6%8E%A7%E5%88%B6%E9%94%AE)+2，而非^@，后者会显示成两字符，前者只会显示成一字符。

## 1.3 ASCLL码的应用

它主要用于显示[现代英语](https://zh.wikipedia.org/wiki/%E7%8F%BE%E4%BB%A3%E8%8B%B1%E8%AA%9E)，而其扩展版本[EASCII](https://zh.wikipedia.org/wiki/EASCII)则可以部分支持其他[西欧](https://zh.wikipedia.org/wiki/%E8%A5%BF%E6%AC%A7)[语言](https://zh.wikipedia.org/wiki/%E8%AF%AD%E8%A8%80)，并等同于国际标准[**ISO/IEC 646**](https://zh.wikipedia.org/wiki/ISO/IEC_646)。

## 1.4 ASCLL码的缺点

ASCII的局限在于只能显示26个基本拉丁字母、阿拉伯数目字和英式标点符号，因此只能用于显示现代美国英语（而且在处理英语当中的外来词如naïve、café、élite等等时，所有重音符号都不得不去掉，即使这样做会违反拼写规则）。而EASCII虽然解决了部分西欧语言的显示问题，但对更多其他语言依然无能为力。

# 2. Unicode码

## 2.1 Unicode码的来历

**Unicode**（[中文](https://zh.wikipedia.org/wiki/%E4%B8%AD%E6%96%87)：万国码、国际码、统一码、单一码）是[计算机科学](https://zh.wikipedia.org/wiki/%E9%9B%BB%E8%85%A6%E7%A7%91%E5%AD%B8)领域里的一项业界标准。

Unicode是为了解决传统的[字符编码](https://zh.wikipedia.org/wiki/%E5%AD%97%E5%85%83%E7%B7%A8%E7%A2%BC)方案的局限而产生的，例如[ISO 8859-1](https://zh.wikipedia.org/wiki/ISO/IEC_8859)所定义的字符虽然在不同的国家中广泛地使用，可是在不同国家间却经常出现不兼容的情况。很多传统的编码方式都有一个共同的问题，即容许电脑处理双语环境（通常使用[拉丁字母](https://zh.wikipedia.org/wiki/%E6%8B%89%E4%B8%81%E5%AD%97%E6%AF%8D)以及其本地语言），但却无法同时支持多语言环境（指可同时处理多种语言混合的情况）。

Unicode编码包含了不同写法的字，如“ɑ／a”、“強／强”、“戶／户／戸”。然而在[汉字](https://zh.wikipedia.org/wiki/%E6%B1%89%E5%AD%97)方面引起了一字多形的认定争议（详见[中日韩统一表意文字](https://zh.wikipedia.org/wiki/%E4%B8%AD%E6%97%A5%E9%9F%93%E7%B5%B1%E4%B8%80%E8%A1%A8%E6%84%8F%E6%96%87%E5%AD%97)主题）。

在文字处理方面，统一码为每一个字符而非字形定义唯一的代码（即一个整数）。换句话说，统一码以一种抽象的方式（即数字）来处理字符，并将视觉上的演绎工作（例如字体大小、外观形状、字体形态、文体等）留给其他软件来处理，例如网页浏览器或是文字处理器。

目前，几乎所有电脑系统都支持基本拉丁字母，并各自支持不同的其他编码方式。Unicode为了和它们相互兼容，其首256字符保留给ISO 8859-1所定义的字符，使既有的西欧语系文字的转换不需特别考量；并且把大量相同的字符重复编到不同的字符码中去，使得旧有纷杂的编码方式得以和Unicode编码间互相直接转换，而不会丢失任何信息。举例来说，[全角](https://zh.wikipedia.org/wiki/%E5%85%A8%E5%BD%A2)格式区块包含了主要的拉丁字母的全角格式，在中文、日文、以及韩文字形当中，这些字符以全角的方式来呈现，而不以常见的半角形式显示，这对竖排文字和等宽排列文字有重要作用。

在表示一个Unicode的字符时，通常会用“U+”然后紧接着一组十六进制的数字来表示这一个字符。在[基本多文种平面](https://zh.wikipedia.org/wiki/%E5%9F%BA%E6%9C%AC%E5%A4%9A%E6%96%87%E7%A8%AE%E5%B9%B3%E9%9D%A2)（英文：Basic Multilingual Plane，简写BMP。又称为“零号平面”、plane 0）里的所有字符，要用四个数字（即两个char,16bit ,例如U+4AE0，共支持六万多个字符）；在零号平面以外的字符则需要使用五个或六个数字。旧版的Unicode标准使用相近的标记方法，但却有些微小差异：在Unicode 3.0里使用“U-”然后紧接着八个数字，而“U+”则必须随后紧接着四个数字。

位于美国加州的Unicode组织允许任何愿意支付会费的公司和个人加入，其成员包含了主要的电脑软硬件厂商，例如[奥多比系统](https://zh.wikipedia.org/wiki/Adobe_Systems" \o "Adobe Systems)、[苹果公司](https://zh.wikipedia.org/wiki/%E8%98%8B%E6%9E%9C%E5%85%AC%E5%8F%B8)、[惠普](https://zh.wikipedia.org/wiki/HP)、[IBM](https://zh.wikipedia.org/wiki/IBM)、[微软](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BB%9F" \o "微软)、[施乐](https://zh.wikipedia.org/wiki/%E6%96%BD%E4%B9%90)等。

20世纪80年代末，组成Unicode组织的商业机构，和国际合作的[国际标准化组织](https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E6%A8%99%E6%BA%96%E5%8C%96%E7%B5%84%E7%B9%94" \o "国际标准化组织)因为电脑普及和信息国际化的前提下，分别各自成立了Unicode组织[[3]](https://zh.wikipedia.org/wiki/Unicode" \l "cite_note-3)和ISO-10646工作小组。他们不久便发现对方的存在，大家为着相同的目的而工作。1991年，Unicode Consortium与ISO/IEC JTC1/SC2同意保持Unicode码表与ISO 10646标准保持兼容并密切协调各自标准进一步的扩展。虽然实际上两者的字集编码相同，但实质上两者确实为两个不同的标准。Unicode 1.1对应于ISO 10646-1:1993，Unicode 3.0对应于ISO 10646-1:2000，Unicode 3.2对应于ISO 10646-2:2001，Unicode 4.0对应于ISO 10646:2003，Unicode 5.0对应于ISO 10646:2003及附录1–3。

Unicode自版本2.0开始保持了向后兼容，即新的版本仅仅增加字符，原有字符不会被删除或更名。

统一码联盟在1991年首次发布了**The Unicode Standard**。Unicode的开发结合了[国际标准化组织](https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E6%A8%99%E6%BA%96%E5%8C%96%E7%B5%84%E7%B9%94" \o "国际标准化组织)所制定的[ISO/IEC 10646](https://zh.wikipedia.org/wiki/ISO/IEC_10646)，即[通用字符集](https://zh.wikipedia.org/w/index.php?title=%E9%80%9A%E7%94%A8%E5%AD%97%E5%85%83%E9%9B%86&action=edit&redlink=1" \o "通用字符集（页面不存在）)。Unicode与ISO/IEC 10646在编码的运作原理相同，但**The Unicode Standard**包含了更详尽的实现信息、涵盖了更细节的主题，诸如比特编码（bitwise encoding）、校对以及呈现等。**The Unicode Standard**也枚举了诸多的字符特性，包含了那些必须支持两种阅读方向的文字（由左至右或由右至左的文字阅读方向，例如阿拉伯文是由右至左）。Unicode与ISO/IEC 10646这两个标准在术语上的使用有些微的不同。

在2005年，Unicode的第十万个字符被引入成为标准之一，该字符被用于[马拉雅拉姆语](https://zh.wikipedia.org/wiki/%E9%A6%AC%E6%8B%89%E9%9B%85%E6%8B%89%E5%A7%86%E8%AA%9E" \o "马拉雅拉姆语)。

## 2.2 Unicode码的特性

### 2.2.1Unicode设计原则

在《The Unicode Standard Version 6.2 – Core Specification》[[27]](https://zh.wikipedia.org/wiki/Unicode" \l "cite_note-28) ，给出了Unicode的十大设计原则：

* Universality：提供单一、综合的字符集，编码一切现代与大部分历史文献的字符。
* Efficiency：易于处理与分析。
* Characters, not glyphs：字符，而不是字形。
* Semantics：字符要有良好定义的语义
* Plain text：仅限于文本字符
* Logical order：默认内存表示是其逻辑序
* Unification：把不同语言的同一书写系统（scripts）中相同字符统一起来。
* Dynamic composition：[附加符号](https://zh.wikipedia.org/wiki/%E9%99%84%E5%8A%A0%E7%AC%A6%E5%8F%B7" \o "附加符号)可以动态组合。
* Stability：已分配的字符与语义不再改变。

Convertibility：Unicode与其他著名字符集可以精确转换

### 2.2.2Unicode编码方式

统一码的编码方式与[ISO 10646](https://zh.wikipedia.org/wiki/ISO_10646" \o "ISO 10646)的[通用字符集](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86" \o "通用字符集)概念相对应。目前实际应用的统一码版本对应于[UCS-2](https://zh.wikipedia.org/wiki/UCS-2)，使用16[位](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的编码空间。也就是每个字符占用2个[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)。这样理论上一共最多可以表示216（即65536）个字符。基本满足各种语言的使用。实际上当前版本的统一码并未完全使用这16位编码，而是保留了大量空间以作为特殊使用或将来扩展。

上述16位统一码字符构成[基本多文种平面](https://zh.wikipedia.org/wiki/%E5%9F%BA%E6%9C%AC%E5%A4%9A%E6%96%87%E7%A8%AE%E5%B9%B3%E9%9D%A2" \o "基本多文种平面)。最新（但未实际广泛使用）的统一码版本定义了16个[辅助平面](https://zh.wikipedia.org/wiki/%E8%BE%85%E5%8A%A9%E5%B9%B3%E9%9D%A2" \o "辅助平面)，两者合起来至少需要占据21位的编码空间，比3字节略少。但事实上辅助平面字符仍然占用4字节编码空间，与[UCS-4](https://zh.wikipedia.org/wiki/UCS-4)保持一致。未来版本会扩充到ISO 10646-1实现级别3，即涵盖UCS-4的所有字符。UCS-4是一个更大的尚未填充完全的31位字符集，加上恒为0的首位，共需占据32位，即4字节。理论上最多能表示231个字符，完全可以涵盖一切语言所用的符号。

基本多文种平面的字符的编码为*U+hhhh*，其中每个*h*代表一个[十六进制](https://zh.wikipedia.org/wiki/%E5%8D%81%E5%85%AD%E8%BF%9B%E5%88%B6" \o "十六进制)数字，与UCS-2编码完全相同。而其对应的4字节UCS-4编码后两个字节一致，前两个字节则所有位均为0。

关于统一码和ISO 10646及UCS的详细关系，见[通用字符集](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86" \o "通用字符集)

### 2.2.3Unicode的输入方法

不同的操作系统，各有直接输入Unicode字符的方法：

* 基于[X Window System](https://zh.wikipedia.org/wiki/X_Window_System" \o "X Window System)的[Linux](https://zh.wikipedia.org/wiki/Linux" \o "Linux)系统，如[Ubuntu](https://zh.wikipedia.org/wiki/Ubuntu" \o "Ubuntu)的Gnome Terminal，首先按下[Ctrl](https://zh.wikipedia.org/wiki/%E6%8E%A7%E5%88%B6%E9%94%AE" \o "控制键)+[Shift](https://zh.wikipedia.org/wiki/%E6%8D%A2%E6%A1%A3%E9%94%AE)+U，然后输入16进制Unicode数，如[interpunct间隔符](https://zh.wikipedia.org/wiki/%E9%97%B4%E9%9A%94%E5%8F%B7" \o "间隔号)输入00b7，最后按空格键；
* [Microsoft Windows](https://zh.wikipedia.org/wiki/Microsoft_Windows)系统，按下[Alt](https://zh.wikipedia.org/wiki/%E8%BD%AC%E6%8D%A2%E9%94%AE" \o "转换键)+0183表示[interpunct间隔符](https://zh.wikipedia.org/wiki/%E9%97%B4%E9%9A%94%E5%8F%B7" \o "间隔号)。

在[SGML](https://zh.wikipedia.org/wiki/SGML" \o "SGML)、[HTML](https://zh.wikipedia.org/wiki/HTML" \o "HTML)、[XML](https://zh.wikipedia.org/wiki/XML" \o "XML)的文本中，使用[字符值引用](https://zh.wikipedia.org/wiki/%E5%AD%97%E7%AC%A6%E5%80%BC%E5%BC%95%E7%94%A8" \o "字符值引用)或[字符实体引用](https://zh.wikipedia.org/wiki/XML%E4%B8%8EHTML%E5%AD%97%E7%AC%A6%E5%AE%9E%E4%BD%93%E5%BC%95%E7%94%A8%E5%88%97%E8%A1%A8)表示一个Unicode字符。

截至2011年10月，可以使用[微软拼音](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BD%AF%E6%8B%BC%E9%9F%B3" \o "微软拼音)2003或2007版本，[仓颉输入法](https://zh.wikipedia.org/wiki/%E5%80%89%E9%A0%A1%E8%BC%B8%E5%85%A5%E6%B3%95" \o "仓颉输入法)第三代第五代第六代版本，[郑码](https://zh.wikipedia.org/wiki/%E9%83%91%E7%A0%81)Unicode版本，[海峰五笔](https://zh.wikipedia.org/wiki/%E6%B5%B7%E5%B3%B0%E4%BA%94%E7%AC%94" \o "海峰五笔)9.3版本，新[注音输入法](https://zh.wikipedia.org/wiki/%E6%B3%A8%E9%9F%B3%E8%BC%B8%E5%85%A5%E6%B3%95" \o "注音输入法)和[VimIM](https://zh.wikipedia.org/wiki/VimIM)进行输入。

* [微软拼音](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BD%AF%E6%8B%BC%E9%9F%B3)在[输入法](https://zh.wikipedia.org/wiki/%E8%BC%B8%E5%85%A5%E6%B3%95)引导状态下，单击语言栏上的“功能菜单”按钮，指向“辅助输入法”即可发现“Unicode码输入方式”，利用它可以直接输入Unicode相应[十六进制](https://zh.wikipedia.org/wiki/%E5%8D%81%E5%85%AD%E9%80%B2%E5%88%B6" \o "十六进制)值的方式输入相应文字。例如中文“胥”输入“5066”，朝鲜文字“셅”输入“c145”（不需要在前面加0x或x）。
* [仓颉输入法](https://zh.wikipedia.org/wiki/%E5%80%89%E9%A0%A1%E8%BC%B8%E5%85%A5%E6%B3%95)已为Unicode汉字、类汉字编码，可以在仓颉输入法方式下通过仓颉码输入方式输入Unicode内的中日韩汉字及韩文。以仓颉输入法第五代为例，例如[汉字](https://zh.wikipedia.org/wiki/%E6%BC%A2%E5%AD%97" \o "汉字)“㗎”输入“口大口木”，汉字“胥”输入“弓人月”，汉字“㿱”输入“手中木竹水”，朝鲜文字“닮”输入“尸卜尸女口”。
* [郑码](https://zh.wikipedia.org/wiki/%E9%83%91%E7%A0%81)已为Unicode汉字、类汉字编码，可以在郑码输入法方式下通过[郑码](https://zh.wikipedia.org/wiki/%E9%83%91%E7%A0%81" \o "郑码)字码输入方式输入Unicode内的中日韩汉字及韩文。例如[汉字](https://zh.wikipedia.org/wiki/%E6%BC%A2%E5%AD%97" \o "汉字)“㗎”输入“JYJF”，汉字“胥”输入“XIQ”，汉字“㿱”输入“DPXI”，朝鲜文字“길”输入“XIYY”。
* [海峰五笔](https://zh.wikipedia.org/wiki/%E6%B5%B7%E5%B3%B0%E4%BA%94%E7%AC%94)此输入法已经直接支持通过[五笔](https://zh.wikipedia.org/wiki/%E4%BA%94%E7%AD%86)码输入方式输入Unicode内的任意中日韩汉字，但无法使用键入Unicode码的方式输入。例如[汉字](https://zh.wikipedia.org/wiki/%E6%BC%A2%E5%AD%97" \o "汉字)（Unicode部分）“㗎”为“keks”，[CJK](https://zh.wikipedia.org/wiki/CJK" \o "CJK)扩展B区的“𣿱”为“iyho”和CJK扩展C区的“𫆦”为“muih”。
* 新[注音输入法](https://zh.wikipedia.org/wiki/%E6%B3%A8%E9%9F%B3%E8%BC%B8%E5%85%A5%E6%B3%95)在输入法引导状态时，打入键盘上的“多功能前导字符键”（及通用键盘上之“`”），第一次使用会弹出说明。输入Unicode字符“胥”则是在键盘上键入“`U5066”。而韩语中的“셅”，则输入“`UC145”。而要输入日语自制汉字“峠”，则是“`U5CE0”。
* [VimIM](https://zh.wikipedia.org/wiki/VimIM)在Vim环境中，可以[直接键入](https://web.archive.org/web/20100413100909/http:/vimim.googlecode.com/svn/vimim/vimim.html" \l "unicode)十进制或十六进制Unicode码。既不需要引导输入法，也不需要码表。

[呒虾米输入法](https://zh.wikipedia.org/wiki/%E5%98%B8%E8%9D%A6%E7%B1%B3%E8%BC%B8%E5%85%A5%E6%B3%95)，支持Unicode，以26个[英文](https://zh.wikipedia.org/wiki/%E8%8B%B1%E6%96%87" \o "英文)字母为组字按键，支持基本繁体中文、简体中文外，还可直接输入符号、[日文汉字](https://zh.wikipedia.org/wiki/%E6%97%A5%E6%96%87%E6%BC%A2%E5%AD%97)及[平假名](https://zh.wikipedia.org/wiki/%E5%B9%B3%E5%81%87%E5%90%8D)、[片假名](https://zh.wikipedia.org/wiki/%E7%89%87%E5%81%87%E5%90%8D)。

除了输入法外，[操作系统](https://zh.wikipedia.org/wiki/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%B5%B1" \o "操作系统)也会提供另外几种方法输入Unicode。像是[Windows 2000](https://zh.wikipedia.org/wiki/Windows_2000" \o "Windows 2000)之后的Windows系统就提供一个可点击的[字符映射表](https://zh.wikipedia.org/wiki/%E5%AD%97%E7%AC%A6%E6%98%A0%E5%B0%84%E8%A1%A8" \o "字符映射表)。又或者在[Microsoft Word](https://zh.wikipedia.org/wiki/Microsoft_Word)下，按下Alt键不放，输入0和某个字符的Unicode编码（[十进制](https://zh.wikipedia.org/wiki/%E5%8D%81%E8%BF%9B%E5%88%B6" \o "十进制)），再松开Alt键即可得到该字符，如Alt + 033865会得到Unicode字符*叶*。另外按Alt + X组合键，MS Word也会将光标前面的字符同其[十六进制](https://zh.wikipedia.org/wiki/%E5%8D%81%E5%85%AD%E8%BF%9B%E5%88%B6" \o "十六进制)的四位Unicode编码进行互相转换。

## 2.3 Unicode码的应用

Unicode伴随着[通用字符集](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86" \o "通用字符集)的标准而发展，同时也以书本的形式[[1]](https://zh.wikipedia.org/wiki/Unicode#cite_note-1)对外发表。Unicode至今仍在不断增修，每个新版本都加入更多新的字符。目前最新的版本为2017年6月20日公布的10.0.0[[2]](https://zh.wikipedia.org/wiki/Unicode#cite_note-Unicode10.0-2)，已经收录超过十万个[字符](https://zh.wikipedia.org/wiki/%E5%AD%97%E7%AC%A6" \o "字符)（第十万个字符在2005年获采纳）。Unicode涵盖的数据除了视觉上的字形、编码方法、标准的[字符编码](https://zh.wikipedia.org/wiki/%E5%AD%97%E7%AC%A6%E7%BC%96%E7%A0%81)外，还包含了字符特性，如大小写字母。

Unicode发展由非营利机构[统一码联盟](https://zh.wikipedia.org/wiki/%E7%B5%B1%E4%B8%80%E7%A2%BC%E8%81%AF%E7%9B%9F" \o "统一码联盟)负责，该机构致力于让Unicode方案取代既有的字符编码方案。因为既有的方案往往空间非常有限，亦不适用于[多语](https://zh.wikipedia.org/wiki/%E5%A4%9A%E8%AA%9E" \o "多语)环境。

Unicode备受认可，并广泛地应用于电脑软件的[国际化与本地化](https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E5%8C%96%E8%88%87%E6%9C%AC%E5%9C%B0%E5%8C%96" \o "国际化与本地化)过程。有很多新科技，如[可扩展置标语言](https://zh.wikipedia.org/wiki/%E5%8F%AF%E6%89%A9%E5%B1%95%E7%BD%AE%E6%A0%87%E8%AF%AD%E8%A8%80)(Extensible Markup Language，简称：XML)、[Java编程语言](https://zh.wikipedia.org/wiki/Java" \o "Java)以及现代的[操作系统](https://zh.wikipedia.org/wiki/%E4%BD%9C%E6%A5%AD%E7%B3%BB%E7%B5%B1)，都采用Unicode编码。

# 3. Big5码

## 3.1 Big5码的来历

**Big5**，又称为**大五码**或**五大码**，是使用[繁体中文](https://zh.wikipedia.org/wiki/%E7%B9%81%E4%BD%93%E4%B8%AD%E6%96%87)（正体中文）社区中最常用的电脑[汉字](https://zh.wikipedia.org/wiki/%E6%B1%89%E5%AD%97" \o "汉字)[字符集](https://zh.wikipedia.org/wiki/%E5%AD%97%E7%AC%A6%E9%9B%86)标准，共收录13,060个汉字[[1]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC" \l "cite_note-1)。

“大五码”（Big5）是由台湾[财团法人信息产业策进会](https://zh.wikipedia.org/wiki/%E8%B3%87%E8%A8%8A%E5%B7%A5%E6%A5%AD%E7%AD%96%E9%80%B2%E6%9C%83" \o "信息产业策进会)为[五大中文套装软件](https://zh.wikipedia.org/wiki/%E4%BA%94%E5%A4%A7%E4%B8%AD%E6%96%87%E5%A5%97%E8%A3%9D%E8%BB%9F%E9%AB%94)所设计的中文共通内码，在1983年12月完成公告[[2]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC" \l "cite_note-2)[[3]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-3)，隔年3月，信息产业策进会与[台湾](https://zh.wikipedia.org/wiki/%E8%87%BA%E7%81%A3" \o "台湾)13家厂商签定“16位个人电脑套装软件合作开发（BIG-5）项目（[五大中文套装软件](https://zh.wikipedia.org/wiki/%E4%BA%94%E5%A4%A7%E4%B8%AD%E6%96%87%E5%A5%97%E8%A3%9D%E8%BB%9F%E9%AB%94" \o "五大中文套装软件)）”[[4]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-4)，因为此中文内码是为台湾自行制作开发之“五大中文套装软件”所设计的，所以就称为Big5中文内码[[5]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC" \l "cite_note-5)[[6]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-6)[[7]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-7)[[8]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-8)。五大中文套装软件虽然并没有如预期的取代国外的套装软件，但随着采用Big5码的[国乔中文系统](https://zh.wikipedia.org/w/index.php?title=%E5%9C%8B%E5%96%AC%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1&action=edit&redlink=1" \o "国乔中文系统（页面不存在）)及[倚天中文系统](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1)先后在台湾市场获得成功，使得Big5码深远地影响繁体中文电脑[内码](https://zh.wikipedia.org/wiki/%E5%85%A7%E7%A2%BC" \o "内码)，直至今日。“五大码”的英文名称“Big5”后来被人按英文字序译回中文，以致现在有“五大码”和“大五码”两个中文名称。

Big5码的产生，是因为当时[个人电脑](https://zh.wikipedia.org/wiki/%E5%80%8B%E4%BA%BA%E9%9B%BB%E8%85%A6" \o "个人电脑)没有共通的内码，导致厂商推出的中文应用软件无法推广，并且与[IBM 5550](https://zh.wikipedia.org/wiki/IBM_5550)、[王安码](https://zh.wikipedia.org/w/index.php?title=%E7%8E%8B%E5%AE%89%E7%A2%BC&action=edit&redlink=1" \o "王安码（页面不存在）)等内码，彼此不能兼容；另一方面，台湾当时尚未推出中文编码标准。在这样的时空背景下，为了使台湾早日进入信息时代，所采行的一个项目；同时，这个项目对于以台湾为核心的亚洲繁体汉字圈也产生了久远的影响。

Big5产生前，研发[中文电脑](https://zh.wikipedia.org/wiki/%E4%B8%AD%E6%96%87%E9%9B%BB%E8%85%A6" \o "中文电脑)的[朱邦复](https://zh.wikipedia.org/wiki/%E6%9C%B1%E9%82%A6%E5%BE%A9)认为内码字集应该广纳所有的正异体字，以顾及如户政等应用上的需要，故在当时的内码会议中，建议希望采用他的五万多字的字库。工程师认为虽其技术可行，但是三个字节（超过两个字节）长度的内码却会造成英文显示屏画面映射成中文画面会发生文字无法对齐的问题，因为当时盛行之倚天中文系统画面系以两个字节文字宽度映射成一个中文字图样，英文软件中只要以两个英文字宽度去显示一个中文字，画面就不会乱掉，造成中文系统业者偏爱二个字节长度的内码[[9]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-9)；此外以仓颉输入码压缩成的内码不具排序等功能，因此未被采用。1983年有人诬指朱邦复为[共产党](https://zh.wikipedia.org/wiki/%E5%85%B1%E7%94%A2%E9%BB%A8" \o "共产党)，其研究成果更不可能获采用。[[10]](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_note-10)

在Big5码诞生后，大部分台湾的电脑软件都使用了Big5码，加上后来[倚天中文系统](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1" \o "倚天中文系统)的高度普及，使后来的[微软](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BD%AF)[Windows 3.x](https://zh.wikipedia.org/wiki/Windows_3.x)等亦予以采用。虽然后来台湾还有各种想要取代Big5码，像是倚天中文系统所推行的[倚天码](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E7%A2%BC" \o "倚天码)、台北市电脑公会所推动的[公会码](https://zh.wikipedia.org/w/index.php?title=%E5%85%AC%E6%9C%83%E7%A2%BC&action=edit&redlink=1)等，但是由于Big5字码已沿用多年，因此在习惯不易改变的情况下，始终无法成为主流字码。而台湾后来发展的[国家标准CNS 11643中文标准交换码](https://zh.wikipedia.org/wiki/CNS11643" \o "CNS11643)由于非一般的内码系统，是以交换使用为目的，受先天所限，必须使用至少三个字节来表示一个汉字，所以普及率远远不及Big5码。

在1990年代初期，当[中国大陆](https://zh.wikipedia.org/wiki/%E4%B8%AD%E5%9C%8B%E5%A4%A7%E9%99%B8" \o "中国大陆)的[电子邮件](https://zh.wikipedia.org/wiki/%E9%9B%BB%E9%83%B5)和转码软件还未普遍之时，在[深圳](https://zh.wikipedia.org/wiki/%E6%B7%B1%E5%9C%B3)的港商和台商公司亦曾经使用Big5系统，以方便与总部的文件交流、以及避免为大陆的办公室再写一套不同内码的系统。使用[简体中文](https://zh.wikipedia.org/wiki/%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87" \o "简体中文)的社区，最常用的是[GB 2312](https://zh.wikipedia.org/wiki/GB_2312)、[GBK](https://zh.wikipedia.org/wiki/GBK" \o "GBK)及其后续的[国标码](https://zh.wikipedia.org/wiki/%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86%E4%BB%A3%E7%A0%81" \o "国家标准代码)（[GB 18030](https://zh.wikipedia.org/wiki/GB_18030)）。

除了台湾外，其他使用繁体汉字的地区，如[香港](https://zh.wikipedia.org/wiki/%E9%A6%99%E6%B8%AF)（[香港增补字符集](https://zh.wikipedia.org/wiki/%E9%A6%99%E6%B8%AF%E5%A2%9E%E8%A3%9C%E5%AD%97%E7%AC%A6%E9%9B%86)）、[澳门](https://zh.wikipedia.org/wiki/%E6%BE%B3%E9%96%80)，及使用繁体汉字的海外华人，都曾普遍使用Big5码做为中文内码及交换码。

## 3.2 Big5码的特性

Big5码是一套[双字节字符集](https://zh.wikipedia.org/wiki/%E5%8F%8C%E5%AD%97%E8%8A%82%E5%AD%97%E7%AC%A6%E9%9B%86" \o "双字节字符集)，使用了双八码存储方法，以两个字节来安放一个字。第一个字节称为“高位字节”，第二个字节称为“低位字节”。

“高位字节”使用了0x81-0xFE，“低位字节”使用了0x40-0x7E，及0xA1-0xFE。在Big5的分区中：

|  |  |
| --- | --- |
| 0x8140-0xA0FE | 保留给用户自定义字符（[造字](https://zh.wikipedia.org/wiki/%E9%80%A0%E5%AD%97)区） |
| 0xA140-0xA3BF | [标点符号](https://zh.wikipedia.org/wiki/%E6%A8%99%E9%BB%9E%E7%AC%A6%E8%99%9F)、[希腊字母](https://zh.wikipedia.org/wiki/%E5%B8%8C%E8%85%8A%E5%AD%97%E6%AF%8D)及特殊符号， 包括在0xA259-0xA261，安放了九个[计量用汉字](https://zh.wikipedia.org/wiki/%E8%A8%88%E9%87%8F%E7%94%A8%E6%BC%A2%E5%AD%97" \o "计量用汉字)：兙兛兞兝兡兣嗧瓩糎。 |
| 0xA3C0-0xA3FE | 保留。此区没有开放作造字区用。 |
| 0xA440-0xC67E | [常用汉字](https://zh.wikipedia.org/wiki/%E5%B8%B8%E7%94%A8%E6%BC%A2%E5%AD%97)，先按[笔划](https://zh.wikipedia.org/wiki/%E7%AD%86%E5%8A%83)再按[部首](https://zh.wikipedia.org/wiki/%E9%83%A8%E9%A6%96)排序。 |
| 0xC6A1-0xC8FE | 保留给用户自定义字符（造字区） |
| 0xC940-0xF9D5 | [次常用汉字](https://zh.wikipedia.org/wiki/%E6%AC%A1%E5%B8%B8%E7%94%A8%E5%AD%97)，亦是先按笔划再按部首排序。 |
| 0xF9D6-0xFEFE | 保留给用户自定义字符（造字区） |

值得留意的是，Big5重复收录了两个相同的字：“兀、兀”（0xA461[U+5140]及0xC94A[U+FA0C]）、“嗀、嗀”（0xDCD1[U+55C0]及0xDDFC[U+FA0D]）。此外“十”、“卅”也在符号区又重复了一次，在检索系统中常会造成查询不到字。

## 3.3 Big5码的应用

Big5虽普及于[台湾](https://zh.wikipedia.org/wiki/%E5%8F%B0%E7%81%A3" \o "台湾)、[香港](https://zh.wikipedia.org/wiki/%E9%A6%99%E6%B8%AF)与[澳门](https://zh.wikipedia.org/wiki/%E6%BE%B3%E9%96%80)等繁体中文通行区，但长期以来并非当地的国家/地区标准或官方标准，而只是**[业界标准](https://zh.wikipedia.org/wiki/De_facto" \o "De facto)**。[倚天中文系统](https://zh.wikipedia.org/wiki/%E5%80%9A%E5%A4%A9%E4%B8%AD%E6%96%87%E7%B3%BB%E7%B5%B1)、[Windows](https://zh.wikipedia.org/wiki/Windows)繁体中文版等主要系统的字符集都是以Big5为基准，但厂商又各自增加不同的造字与造字区，派生成多种不同版本。

# 4. GB2312码

## 4.1 GB2312码的概述

**GB 2312** 或 **GB 2312–80** 是[中华人民共和国国家标准](https://zh.wikipedia.org/wiki/%E4%B8%AD%E5%8D%8E%E4%BA%BA%E6%B0%91%E5%85%B1%E5%92%8C%E5%9B%BD%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86" \o "中华人民共和国国家标准)[简体中文](https://zh.wikipedia.org/wiki/%E7%AE%80%E4%BD%93%E4%B8%AD%E6%96%87)[字符集](https://zh.wikipedia.org/wiki/%E5%AD%97%E7%AC%A6%E9%9B%86)，全称《**信息交换用汉字编码字符集·基本集**》，又称[**GB0**](https://zh.wikipedia.org/wiki/%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86%E4%BB%A3%E7%A0%81)，由[中国国家标准总局](https://zh.wikipedia.org/w/index.php?title=%E4%B8%AD%E5%9B%BD%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86%E6%80%BB%E5%B1%80&action=edit&redlink=1" \o "中国国家标准总局（页面不存在）)发布，1981年5月1日实施。GB 2312编码通行于中国大陆；[新加坡](https://zh.wikipedia.org/wiki/%E6%96%B0%E5%8A%A0%E5%9D%A1" \o "新加坡)等地也采用此编码。中国大陆几乎所有的中文系统和国际化的软件都支持GB 2312。

GB 2312标准共收录6763个[汉字](https://zh.wikipedia.org/wiki/%E6%B1%89%E5%AD%97" \o "汉字)，其中[一级汉字](https://zh.wikipedia.org/wiki/%E5%B8%B8%E7%94%A8%E5%AD%97)3755个，[二级汉字](https://zh.wikipedia.org/wiki/%E6%AC%A1%E5%B8%B8%E7%94%A8%E5%AD%97" \o "次常用字)3008个；同时收录了包括[拉丁字母](https://zh.wikipedia.org/wiki/%E6%8B%89%E4%B8%81%E5%AD%97%E6%AF%8D" \o "拉丁字母)、[希腊字母](https://zh.wikipedia.org/wiki/%E5%B8%8C%E8%85%8A%E5%AD%97%E6%AF%8D)、[日文](https://zh.wikipedia.org/wiki/%E6%97%A5%E8%AF%AD)[平假名](https://zh.wikipedia.org/wiki/%E5%B9%B3%E5%81%87%E5%90%8D)及[片假名](https://zh.wikipedia.org/wiki/%E7%89%87%E5%81%87%E5%90%8D)字母、[俄语](https://zh.wikipedia.org/wiki/%E4%BF%84%E8%AF%AD)[西里尔字母](https://zh.wikipedia.org/wiki/%E6%96%AF%E6%8B%89%E5%A4%AB%E5%AD%97%E6%AF%8D)在内的682个字符。

GB 2312的出现，基本满足了汉字的计算机处理需要，它所收录的汉字已经覆盖中国大陆99.75%的使用频率。但对于[人名](https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%90%8D" \o "人名)、[古汉语](https://zh.wikipedia.org/wiki/%E5%8F%A4%E6%B1%89%E8%AF%AD)等方面出现的[罕用字](https://zh.wikipedia.org/wiki/%E7%BD%95%E7%94%A8%E5%AD%97)和[繁体字](https://zh.wikipedia.org/wiki/%E7%B9%81%E9%AB%94%E5%AD%97)，GB 2312不能处理，因此后来[GBK](https://zh.wikipedia.org/wiki/GBK" \o "GBK)及[GB 18030](https://zh.wikipedia.org/wiki/GB_18030" \o "GB 18030)汉字字符集相继出现以解决这些问题。

## 4.2 GB2312码的特性

GB 2312中对所收汉字进行了“分区”处理，每区含有94个汉字／符号。这种表示方式也称为[区位码](https://zh.wikipedia.org/wiki/ISO/IEC_2022" \o "ISO/IEC 2022)。

* 01–09区为特殊符号。
* 16–55区为一级汉字，按[拼音](https://zh.wikipedia.org/wiki/%E6%8B%BC%E9%9F%B3)排序。
* 56–87区为二级汉字，按[部首](https://zh.wikipedia.org/wiki/%E9%83%A8%E9%A6%96" \o "部首)／[笔画](https://zh.wikipedia.org/wiki/%E7%AC%94%E7%94%BB)排序。

举例来说，“啊”字是GB 2312之中的第一个汉字，它的区位码就是1601。

10–15区及88–94区则未有编码。但在附录3，则在第10区推荐作为 [GB 1988–80](https://zh.wikipedia.org/w/index.php?title=GB_1988&action=edit&redlink=1) 中的94个图形字符区域（即第3区字符之[半形](https://zh.wikipedia.org/wiki/%E5%8D%8A%E5%BD%A2" \o "半形)版本）。

在使用GB 2312的程序通常采用[EUC](https://zh.wikipedia.org/wiki/EUC" \o "EUC)储存方法，以便兼容于[ASCII](https://zh.wikipedia.org/wiki/ASCII" \o "ASCII)。这种格式称为**EUC-CN**。[浏览器](https://zh.wikipedia.org/wiki/%E7%BD%91%E9%A1%B5%E6%B5%8F%E8%A7%88%E5%99%A8" \o "网页浏览器)编码表上的“GB2312”就是指这种表示法。

每个汉字及符号以两个[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)来表示。第一个字节称为“高位字节”，第二个字节称为“低位字节”。

“高位字节”使用了0xA1–0xF7（把01–87区的区号加上0xA0），“低位字节”使用了0xA1–0xFE（把01–94加上0xA0）。 由于一级汉字从16区起始，汉字区的“高位字节”的范围是0xB0–0xF7，“低位字节”的范围是0xA1–0xFE，占用的码位是72\*94=6768。其中有5个空位是D7FA–D7FE。

例如“啊”字在大多数程序中，会以两个字节，0xB0（第一个字节）0xA1（第二个字节）储存。（与区位码对比：0xB0=0xA0+16,0xA1=0xA0+1）。

对“不规范简化字”和繁体字的收录：

1.收了两个不[合乎中华人民共和国标准的简化字](https://zh.wikipedia.org/wiki/%E8%A7%84%E8%8C%83%E6%B1%89%E5%AD%97" \o "规范汉字)：

* + 渖（68–41）：由“审[審]”类推简化而来，但《[简化字总表](https://zh.wikipedia.org/wiki/%E7%B0%A1%E5%8C%96%E5%AD%97%E7%B8%BD%E8%A1%A8" \o "简化字总表)》已将“瀋”简化归并为“沈”。旧版《[新华字典](https://zh.wikipedia.org/wiki/%E6%96%B0%E8%8F%AF%E5%AD%97%E5%85%B8" \o "新华字典)》收有此字，释为“汁”；新版取消，并入“沈”。
  + 镟（79–64）：由“钅[釒]”类推简化而来，但《简化字总表》已将“鏇”简化归并为“旋”。

2.收了两个[繁体字](https://zh.wikipedia.org/wiki/%E7%B9%81%E9%AB%94%E5%AD%97" \o "繁体字)：

* + 鍾（79–81）：原版收入使用繁体偏旁之“鍾”字，但《简化字总表》已将“鍾”和“鐘”简化归并为“钟”；后续的字模附录将之修正为“锺”（下详）[[1]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-1)。
  + 後（65–65）：该繁体字已经在《简化字总表》简化归并成“后”（26–83）字，而且没有说明在语义不清时用“後”来表示，可是GB 2312却多收此字。

GB 5007.1–85《信息交换用汉字 24x24 [点阵](https://zh.wikipedia.org/wiki/%E9%BB%9E%E9%99%A3%E5%AD%97)[字模](https://zh.wikipedia.org/wiki/%E5%AD%97%E6%A8%A1)集》首次附录对 GB 2312 之更正，包括：

* 调整[拉丁字母](https://zh.wikipedia.org/wiki/%E6%8B%89%E4%B8%81%E5%AD%97%E6%AF%8D" \o "拉丁字母)“[g](https://zh.wikipedia.org/wiki/G)”之[字形](https://zh.wikipedia.org/wiki/%E5%AD%97%E5%BD%A2" \o "字形)
* 补充六个[拼音](https://zh.wikipedia.org/wiki/%E6%8B%BC%E9%9F%B3" \o "拼音)符号 [ɑ](https://zh.wikipedia.org/wiki/%C9%91) [ḿ](https://zh.wikipedia.org/wiki/%E1%B8%BE) [ń](https://zh.wikipedia.org/wiki/%C5%83) [ň](https://zh.wikipedia.org/wiki/%C5%87) [ǹ](https://zh.wikipedia.org/wiki/%C7%B8) [ɡ](https://zh.wikipedia.org/wiki/%C9%A1)[[注 1]](https://zh.wikipedia.org/wiki/GB_2312#cite_note-2)
* “鍾”更正为“锺”
* 于第10区补充94个半字图形字符（第3区之[半形](https://zh.wikipedia.org/wiki/%E5%8D%8A%E5%BD%A2" \o "半形)版本，相当于 [GB 1988–80](https://zh.wikipedia.org/w/index.php?title=GB_1988&action=edit&redlink=1)）
* 于第11区加入第8区首32个拼音符号（包括以上补充六个）之半形版本。

## 4.3GB2312码的修订

GB 2312 本身一直未有修订，但此等修订部分收入相关字模集（下详）、[GB 12345](https://zh.wikipedia.org/wiki/GB_12345" \o "GB 12345)、后续之 [GBK](https://zh.wikipedia.org/wiki/GBK) 及 [GB 18030](https://zh.wikipedia.org/wiki/GB_18030)。

GB 2312 亦用于 [ISO-IR-165](https://zh.wikipedia.org/wiki/ISO-IR-165)。

有两种不同的GB 2312实现，在它们之间存在少量的差别，其中至少有一个是错误的。

|  |  |  |  |
| --- | --- | --- | --- |
| **字节序** | **GBK子集** | **GB2312.TXT** | **字符名称[[2]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-gb2312-80-3)** |
| A1A4 | U+00B7 “·” MIDDLE DOT | U+30FB “・” KATAKANA MIDDLE DOT | [间隔点](https://zh.wikipedia.org/wiki/%E9%97%B4%E9%9A%94%E7%82%B9) |
| A1AA | U+2014 “—” EM DASH | U+2015 “―” HORIZONTAL BAR | [破折号](https://zh.wikipedia.org/wiki/%E7%A0%B4%E6%8A%98%E5%8F%B7) |

GBK子集与GBK/GB 18030兼容，GB2312.TXT则不兼容。后者基于ftp.unicode.org曾经提供的GB2312.TXT实现，[[3]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-many-mappings-4)于2011年由官方弃用，[[4]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-5)2016年9月时已无原文件踪迹。此外还有很多种厂商实现。[[3]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-many-mappings-4)

截至2015年，微软.NET使用的是“GBK子集”实现。[ICU](https://zh.wikipedia.org/w/index.php?title=%E4%B8%87%E5%9B%BD%E7%A0%81%E5%9B%BD%E9%99%85%E5%8C%96%E7%BB%84%E4%BB%B6&action=edit&redlink=1)[[5]](https://zh.wikipedia.org/wiki/GB_2312#cite_note-6)、libiconv-1.14、[[6]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-7)php-5.6、ActivePerl-5.20、Java 1.7、Python 3.4[[7]](https://zh.wikipedia.org/wiki/GB_2312#cite_note-8)都使用“GB2312.TXT”实现。Ruby 2.2兼容两者编码，但内部使用“GBK子集”实现。W3C的编码技术指南规定，应将gb2312字节流视为GBK编码，与GB18030一并使用同一解码器解码。[[8]](https://zh.wikipedia.org/wiki/GB_2312" \l "cite_note-9)

# 5. UTF-8码

## 5.1 UTF-8码的来历

1992年初，为创建良好的[字节串编码系统](https://zh.wikipedia.org/w/index.php?title=%E4%BD%8D%E5%85%83%E7%B5%84%E4%B8%B2%E7%B7%A8%E7%A2%BC%E7%B3%BB%E7%B5%B1&action=edit&redlink=1" \o "字节串编码系统（页面不存在）)以供多[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)字符集使用，开始了一个正式的研究。[ISO/IEC 10646](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E5%AD%97%E7%AC%A6%E9%9B%86)的初稿中有一个非必须的附录，名为UTF。当中包含了一个供32[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的[字符](https://zh.wikipedia.org/wiki/UCS)使用的字节串编码系统。这个编码方式的性能并不令人满意，但它提出了将0-127的范围保留给ASCII以兼容旧系统的概念。

1992年7月，[X/Open](https://zh.wikipedia.org/wiki/X/Open" \o "X/Open)委员会XoJIG开始寻求一个较佳的编码系统。[Unix系统实验室](https://zh.wikipedia.org/wiki/Unix%E7%B3%BB%E7%BB%9F%E5%AE%9E%E9%AA%8C%E5%AE%A4" \o "Unix系统实验室)（USL）的Dave Prosser为此提出了一个编码系统的建议。它具备可更快速实现的特性，并引入一项新的改进。其中，7[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的[ASCII](https://zh.wikipedia.org/wiki/ASCII)符号只代表原来的意思，所有多字节序列则会包含第8[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的符号，也就是所谓的[最高有效比特](https://zh.wikipedia.org/wiki/%E6%9C%80%E9%AB%98%E6%9C%89%E6%95%88%E4%BD%8D%E5%85%83)。

1992年8月，这个建议由[IBM](https://zh.wikipedia.org/wiki/IBM" \o "IBM)[X/Open](https://zh.wikipedia.org/wiki/X/Open)的代表流传到一些感兴趣的团体。与此同时，[贝尔实验室](https://zh.wikipedia.org/wiki/%E8%B2%9D%E7%88%BE%E5%AF%A6%E9%A9%97%E5%AE%A4" \o "贝尔实验室)[九号项目](https://zh.wikipedia.org/wiki/%E8%B2%9D%E7%88%BE%E5%AF%A6%E9%A9%97%E5%AE%A4%E4%B9%9D%E8%99%9F%E8%A8%88%E7%95%AB)[操作系统](https://zh.wikipedia.org/wiki/%E4%BD%9C%E6%A5%AD%E7%B3%BB%E7%B5%B1)工作小组的[肯·汤普逊](https://zh.wikipedia.org/wiki/%E8%82%AF%C2%B7%E6%B1%A4%E6%99%AE%E9%80%8A)对这编码系统作出重大的修改，让编码可以自我同步，使得不必从字符串的开首读取，也能找出字符间的分界。1992年9月2日，[肯·汤普逊](https://zh.wikipedia.org/wiki/%E8%82%AF%C2%B7%E6%B1%A4%E6%99%AE%E9%80%8A" \o "肯·汤普逊)和[罗勃·派克](https://zh.wikipedia.org/wiki/%E7%BE%85%E5%8B%83%C2%B7%E6%B4%BE%E5%85%8B)一起在[美国](https://zh.wikipedia.org/wiki/%E7%BE%8E%E5%9C%8B)[新泽西州](https://zh.wikipedia.org/wiki/%E6%96%B0%E6%BE%A4%E8%A5%BF%E5%B7%9E)一架餐车的餐桌垫上描绘出此设计的要点。接下来的日子，Pike及汤普逊将它实现，并将这编码系统完全应用在[九号项目](https://zh.wikipedia.org/wiki/%E8%B2%9D%E7%88%BE%E5%AF%A6%E9%A9%97%E5%AE%A4%E4%B9%9D%E8%99%9F%E8%A8%88%E7%95%AB" \o "贝尔实验室九号项目)当中，及后他将有关成果回馈X/Open。

1993年1月25-29日的在[圣地牙哥](https://zh.wikipedia.org/wiki/%E8%81%96%E5%9C%B0%E7%89%99%E5%93%A5" \o "圣地牙哥)举行的[USENIX](https://zh.wikipedia.org/wiki/USENIX)会议首次正式介绍UTF-8。

自1996年起，[微软](https://zh.wikipedia.org/wiki/%E5%BE%AE%E8%BB%9F" \o "微软)的[CAB](https://zh.wikipedia.org/wiki/CAB)（MS Cabinet）规格在UTF-8标准正式落实前就明确容许在任何地方使用UTF-8编码系统。但有关的编码器实际上从来没有实现这方面的规格。

## 5.2 UTF-8码的特性

|  |  |
| --- | --- |
| **UTF-8图表说明** | |
|  | **UTF-8** |
| 最小码位 | 0000 |
| 最大码位 | 10FFFF |
| 每字节所占位数 | 8 bits |
| Byte order | N/A |
| 每个字符最小字节数 | 1 |
| 每个字符最大字节数 | 4 |

* UCS字符U+0000到U+007F（ASCII）被编码为字节0x00到0x7F（ASCII兼容），这也意味着只包含7位ASCII字符的文件在ASCII和UTF-8两种编码方式下是一样的。
* 所有>U+007F的UCS字符被编码为一个多个字节的串，每个字节都有标记位集。因此，ASCII字节（0x00-0x7F）不可能作为任何其他字符的一部分。
* 表示非ASCII字符的多字节串的第一个字节总是在0xC0到0xFD的范围里，并指出这个字符包含多少个字节。多字节串的其余字节都在0x80到0xBF范围里，这使得重新同步非常容易，并使编码无国界，且很少受丢失字节的影响。
* 可以编入所有可能的231个UCS代码
* UTF-8编码字符理论上可以最多到6个字节长，然而16位BMP字符最多只用到3字节长。
* Bigendian UCS-4字节串的排列顺序是预定的。
* 字节0xFE和0xFF在UTF-8编码中从未用到，同时，UTF-8以字节为编码单元，它的字节顺序在所有系统中都是一様的，没有字节序的问题，也因此它实际上并不需要[BOM](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83%E7%B5%84%E9%A0%86%E5%BA%8F%E8%A8%98%E8%99%9F)。
* 与UTF-16或其他Unicode编码相比，对于不支持Unicode和XML的系统，UTF-8更不容易造成问题。

## 5.3 UTF-8的编码字节含义

* 对于UTF-8编码中的任意字节B，如果B的第一位为0，则B独立的表示一个字符(ASCII码)；
* 如果B的第一位为1，第二位为0，则B为一个多字节字符中的一个字节(非ASCII字符)；
* 如果B的前两位为1，第三位为0，则B为两个字节表示的字符中的第一个字节；
* 如果B的前三位为1，第四位为0，则B为三个字节表示的字符中的第一个字节；
* 如果B的前四位为1，第五位为0，则B为四个字节表示的字符中的第一个字节；

因此，对UTF-8编码中的任意字节，根据第一位，可判断是否为ASCII字符；根据前二位，可判断该字节是否为一个字符编码的第一个字节；根据前四位（如果前两位均为1），可确定该字节为字符编码的第一个字节，并且可判断对应的字符由几个字节表示；根据前五位（如果前四位为1），可判断编码是否有错误或数据传输过程中是否有错误。

UTF-8的设计有以下的多字符组序列的特质：

* 单字节字符的[最高有效比特](https://zh.wikipedia.org/wiki/%E6%9C%80%E9%AB%98%E6%9C%89%E6%95%88%E4%BD%8D%E5%85%83)永远为0。
* 多字节序列中的首个字符组的几个[最高有效比特](https://zh.wikipedia.org/wiki/%E6%9C%80%E9%AB%98%E6%9C%89%E6%95%88%E4%BD%8D%E5%85%83)决定了序列的长度。最高有效位为110的是2字节序列，而1110的是三字节序列，如此类推。
* 多字节序列中其余的字节中的首两个最高有效比特为10。

UTF-8的这些特质，保证了一个字符的[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)序列不会包含在另一个字符的[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82)序列中。这确保了以字节为基础的部分字符串比对（sub-string match）方法可以适用于在文字中搜索字或词。有些比较旧的可变长度8位编码（如[Shift JIS](https://zh.wikipedia.org/wiki/Shift_JIS" \o "Shift JIS)）没有这个特质，故字符串比对的算法变得相当复杂。虽然这增加了UTF-8编码的字符串的[信息冗余](https://zh.wikipedia.org/wiki/%E4%BF%A1%E6%81%AF%E5%86%97%E4%BD%99" \o "信息冗余)，但是利多于弊。另外，[数据压缩](https://zh.wikipedia.org/wiki/%E8%B3%87%E6%96%99%E5%A3%93%E7%B8%AE)并非Unicode的目的，所以不可混为一谈。即使在发送过程中有部分字节因错误或干扰而完全丢失，还是有可能在下一个字符的起点重新同步，令受损范围受到限制。

另一方面，由于其字节序列设计，如果一个疑似为字符串的序列被验证为UTF-8编码，那么我们可以有把握地说它是UTF-8字符串。一段两字节随机序列碰巧为合法的UTF-8而非ASCII的概率为32分1。对于三字节序列的概率为256分1，对更长的序列的概率就更低了。

# 6. UTF-16码

## 6.1 UTF-16码的来历

**UTF-16**是[Unicode](https://zh.wikipedia.org/wiki/Unicode" \o "Unicode)字符编码五层次模型的第三层：字符编码表（Character Encoding Form，也称为"storage format"）的一种实现方式。即把Unicode字符集的抽象[码位](https://zh.wikipedia.org/wiki/%E7%A0%81%E4%BD%8D" \o "码位)映射为16位长的整数（即[码元](https://zh.wikipedia.org/wiki/%E7%A0%81%E5%85%83" \o "码元)）的序列，用于数据存储或传递。Unicode字符的码位，需要1个或者2个16位长的码元来表示，因此这是一个变长表示。

UTF是"Unicode/UCS Transformation Format"的首字母缩写，即把Unicode字符转换为某种格式之意。UTF-16正式定义于[ISO/IEC 10646-1](https://zh.wikipedia.org/wiki/ISO_10646" \o "ISO 10646)的附录C，而[RFC](https://zh.wikipedia.org/wiki/RFC" \o "RFC)2781也定义了相似的做法

Unicode的编码空间从U+0000到U+10FFFF，共有1,112,064个码位（code point）可用来映射字符. Unicode的编码空间可以划分为17个平面（plane），每个平面包含216（65,536）个码位。17个平面的码位可表示为从U+xx0000到U+xxFFFF，其中xx表示十六进制值从0016到1016，共计17个平面。第一个平面称为**基本多语言平面**（Basic Multilingual Plane, **BMP**），或称第零平面（Plane 0）。其他平面称为**辅助平面**（Supplementary Planes）。基本多语言平面内，从U+D800到U+DFFF之间的码位区块是永久保留不映射到Unicode字符。UTF-16就利用保留下来的0xD800-0xDFFF区段的码位来对辅助平面的字符的码位进行编码。

。

## 6.2 UTF-16码的特性

第一个Unicode平面（码位从U+0000至U+FFFF）包含了最常用的字符。该平面被称为基本多语言平面，缩写为*BMP*（Basic Multilingual Plane, BMP）。UTF-16与[UCS-2](https://zh.wikipedia.org/wiki/UCS-2" \o "UCS-2)编码这个范围内的码位为16比特长的单个码元，数值等价于对应的码位. BMP中的这些码位是仅有的可以在UCS-2中表示的码位。

辅助平面（Supplementary Planes）中的码位，在UTF-16中被编码为**一对**16比特长的码元

UTF-16是自同步（self-synchronizing）:可以通过仅检查一个码元就可以判定给定字符的下一个字符的起始码元.

UTF-16的大尾序和小尾序存储形式都在用。一般来说，以[Macintosh](https://zh.wikipedia.org/wiki/Macintosh)制作或存储的文字使用大尾序格式，以[Microsoft](https://zh.wikipedia.org/wiki/Microsoft" \o "Microsoft)或[Linux](https://zh.wikipedia.org/wiki/Linux" \o "Linux)制作或存储的文字使用小尾序格式。

为了弄清楚UTF-16文件的大小尾序，在UTF-16文件的开首，都会放置一个U+FEFF字符作为[Byte Order Mark](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83%E7%B5%84%E9%A0%86%E5%BA%8F%E8%A8%98%E8%99%9F" \o "字节顺序记号)（UTF-16LE以FF FE代表，UTF-16BE以FE FF代表），以显示这个文本文件是以UTF-16编码，其中U+FEFF字符在UNICODE中代表的意义是ZERO WIDTH NO-BREAK SPACE，顾名思义，它是个没有宽度也没有断字的空白。

# 7. 各种编码间的转换

Unicode与其他著名字符集可以精确转换。

如果一个仅包含基本7位[ASCII](https://zh.wikipedia.org/wiki/ASCII" \o "ASCII)字符的Unicode文件，如果每个字符都使用2字节的原Unicode编码传输，其第一字节的8位始终为0。这就造成了比较大的浪费。对于这种情况，可以使用UTF-8编码，这是一种变长编码，它将基本7位ASCII字符仍用7位编码表示，占用一个字节（首位补0）。而遇到与其他Unicode字符混合的情况，将按一定算法转换，每个字符使用1-3个字节编码，并利用首位为0或1进行识别。这样对以7位ASCII字符为主的西文文档就大幅节省了编码长度（具体方案参见*[UTF-8](https://zh.wikipedia.org/wiki/UTF-8" \o "UTF-8)*）。

“Unicode补完项目”前称“BIG5 Extension”，通过修改Microsoft Windows及[Mozilla](https://zh.wikipedia.org/wiki/Mozilla" \o "Mozilla)的编码表，从而用户能在网上传递及交换文字。

有鉴于“中国海字集”的成功，“Unicode补完项目”第二版采用了“中国海字集”原有的造字，再加上“中国海字集”所欠的部分简体中文字及香港粤语用字，建成一个能在Big5及Unicode之间转换的编码表；该计划目前已推出了64位测试版。

**UTF-8**（**8-bit Unicode Transformation Format**）是一种针对[Unicode](https://zh.wikipedia.org/wiki/Unicode" \o "Unicode)的可变长度[字符编码](https://zh.wikipedia.org/wiki/%E5%AD%97%E5%85%83%E7%B7%A8%E7%A2%BC" \o "字符编码)，也是一种[前缀码](https://zh.wikipedia.org/wiki/%E5%89%8D%E7%BC%80%E7%A0%81)。它可以用来表示Unicode标准中的任何字符，且其编码中的第一个[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)仍与[ASCII](https://zh.wikipedia.org/wiki/ASCII)兼容，这使得原来处理ASCII字符的[软件](https://zh.wikipedia.org/wiki/%E8%BB%9F%E9%AB%94" \o "软件)无须或只须做少部分修改，即可继续使用。

Unicode[字符](https://zh.wikipedia.org/wiki/UCS)的[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)被分区为数个部分，并分配到UTF-8的[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82" \o "字节)串中较低的[比特](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的位置。在U+0080的以下[字符](https://zh.wikipedia.org/wiki/UCS" \o "UCS)都使用内含其字符的单[字节](https://zh.wikipedia.org/wiki/%E5%AD%97%E8%8A%82)编码。这些编码正好对应7比特的ASCII字符。在其他情况，有可能需要多达4个字符组来表示一个字符。这些多字节的[最高有效比特](https://zh.wikipedia.org/wiki/%E6%9C%80%E9%AB%98%E6%9C%89%E6%95%88%E4%BD%8D%E5%85%83" \o "最高有效比特)会设置成1，以防止与7比特的ASCII字符混淆，并保持标准的字节主导字符串运作顺利。

[**Note 1**](https://zh.wikipedia.org/wiki/UTF-8#ref_D800Note_1) Unicode在范围D800-DFFF中不存在任何字符，[基本多文种平面](https://zh.wikipedia.org/wiki/%E5%9F%BA%E6%9C%AC%E5%A4%9A%E6%96%87%E7%A8%AE%E5%B9%B3%E9%9D%A2" \o "基本多文种平面)中约定了这个范围用于UTF-16扩展标识[辅助平面](https://zh.wikipedia.org/wiki/%E8%BE%85%E5%8A%A9%E5%B9%B3%E9%9D%A2" \o "辅助平面)（两个UTF-16表示一个[辅助平面](https://zh.wikipedia.org/wiki/%E8%BE%85%E5%8A%A9%E5%B9%B3%E9%9D%A2" \o "辅助平面)字符）。当然，任何编码都是可以被转换到这个范围，但在unicode中他们并不代表任何合法的值。

Unicode标准规定U+D800..U+DFFF的值不对应于任何字符。但是在使用UCS-2的时代，U+D800..U+DFFF内的值被占用，用于某些字符的映射。但只要不构成代理对，许多UTF-16编码解码还是能把这些不匹配Unicode标准的字符映射正确的辨识、转换成合规的码元[[2]](https://zh.wikipedia.org/wiki/UTF-16#cite_note-2).按照Unicode标准，这种码元序列本来应算作编码错误。

UTF-16比起[UTF-8](https://zh.wikipedia.org/wiki/UTF-8" \o "UTF-8)，好处在于大部分字符都以固定长度的字节（2字节）存储，但UTF-16却无法兼容于[ASCII](https://zh.wikipedia.org/wiki/ASCII" \o "ASCII)编码。

# 8. 参考文献

本文资料全部取自维基百科。

# 8.1 ASCLL码

1.[**跳转^**](https://zh.wikipedia.org/wiki/ASCII#cite_ref-1) [ASCII的口语发音](http://www.m-w.com/cgi-bin/audio.pl?ascii001.wav=ASCII). *Merriam Webster*. Accessed 2008-04-14

# 8.2 Unicode码

* 1. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-1) The Unicode Standard 第五版. Addison-Wesley Professional. [ISBN 0321480910](https://zh.wikipedia.org/wiki/Special:%E7%BD%91%E7%BB%9C%E4%B9%A6%E6%BA%90/0321480910).
  2. ^ [跳转至：**2.0**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-Unicode10.0_2-0) [**2.1**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-Unicode10.0_2-1) [Unicode 10.0.0](http://www.unicode.org/versions/Unicode10.0.0/). Unicode Consortium. [2017-06-20].
  3. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-3) [Unicode](http://www.unicode.org/)
  4. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-5) [Unicode Data 1.0.0](http://www.unicode.org/Public/reconstructed/1.0.0/UnicodeData.txt). [2010-03-16].
  5. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-6) [Unicode Data 1.0.1](http://www.unicode.org/Public/reconstructed/1.0.1/UnicodeData.txt). [2010-03-16].
  6. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-7) [Unicode Data 1995](http://www.unicode.org/Public/1.1-Update/UnicodeData-1.1.5.txt). [2010-03-16].
  7. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-8) [Unicode Data-2.0.14](http://www.unicode.org/Public/2.0-Update/UnicodeData-2.0.14.txt). [2010-03-16].
  8. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-9) [Unicode Data-2.1.2](http://www.unicode.org/Public/2.1-Update/UnicodeData-2.1.2.txt). [2010-03-16].
  9. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-10) [Unicode Data-3.0.0](http://www.unicode.org/Public/3.0-Update/UnicodeData-3.0.0.txt). [2010-03-16].
  10. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-11) [Unicode Data-3.1.0](http://www.unicode.org/Public/3.1-Update/UnicodeData-3.1.0.txt). [2010-03-16].
  11. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-12) [Unicode Data-3.2.0](http://www.unicode.org/Public/3.2-Update/UnicodeData-3.2.0.txt). [2010-03-16].
  12. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-13) [Unicode Data-4.0.0](http://www.unicode.org/Public/4.0-Update/UnicodeData-4.0.0.txt). [2010-03-16].
  13. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-14) [Unicode Data](http://www.unicode.org/Public/4.1.0/ucd/UnicodeData.txt). [2010-03-16].
  14. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-15) [Unicode Data 5.0.0](http://www.unicode.org/Public/5.0.0/ucd/UnicodeData.txt). [2010-03-17].
  15. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-16) [Unicode Data 5.1.0](http://www.unicode.org/Public/5.1.0/ucd/UnicodeData.txt). [2010-03-17].
  16. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-17) [Unicode Data 5.2.0](http://www.unicode.org/Public/5.2.0/ucd/UnicodeData.txt). [2010-03-17].
  17. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-18) [Unicode Data 6.0.0](http://www.unicode.org/Public/6.0.0/ucd/UnicodeData.txt). [2010-10-11].
  18. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-19) [Unicode Data 6.1.0](http://www.unicode.org/Public/6.1.0/ucd/UnicodeData.txt). [2012-01-31].
  19. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-20) [Unicode Data 6.2.0](http://www.unicode.org/Public/6.2.0/ucd/UnicodeData.txt). [2012-09-26].
  20. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-21) [Unicode Data 6.3.0](http://www.unicode.org/Public/6.3.0/ucd/UnicodeData.txt). [2013-09-30].
  21. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-22) [Unicode Data 7.0.0](http://www.unicode.org/Public/7.0.0/ucd/UnicodeData.txt). [2014-06-15].
  22. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-23) [Unicode 8.0.0](http://www.unicode.org/versions/Unicode8.0.0/). Unicode Consortium. [2015-06-17].
  23. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-24) [Unicode Data 8.0.0](http://www.unicode.org/Public/8.0.0/ucd/UnicodeData.txt). [2015-06-17].
  24. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-25) [Unicode 9.0.0](http://www.unicode.org/versions/Unicode9.0.0/). Unicode Consortium. [2016-06-21].
  25. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-26) [Unicode Data 9.0.0](http://www.unicode.org/Public/9.0.0/ucd/UnicodeData.txt). [2016-06-21].
  26. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-27) [Unicode Data 10.0.0](http://www.unicode.org/Public/10.0.0/ucd/UnicodeData.txt). [2017-06-20].
  27. [**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-28) [The Unicode Standard, Version 6.2, 2012 Unicode, Inc. Chapter 2 General Structure 2.2 Unicode Design Principles](http://www.unicode.org/versions/Unicode6.2.0/ch02.pdf)

[**跳转^**](https://zh.wikipedia.org/wiki/Unicode#cite_ref-han_cjk_29-0) [Why doesn't the Unicode Standard adopt a compositional model for encoding Han ideographs? Wouldn't that save a large number of code points?](http://www.unicode.org/faq/han_cjk.html#16)

# 8.3 Big5码

1. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-1) 普遍认为大五码包含13,053字，但在计算0xA259-0xA261的九个[度量衡单位用字](https://zh.wikipedia.org/wiki/%E8%A8%88%E9%87%8F%E7%94%A8%E6%BC%A2%E5%AD%97" \o "计量用汉字)（兙兛兞兝兡兣嗧瓩糎），再减去重收了两次的“兀”（0xC94A）和“嗀”（0xDDFC）后，应为13,060字。
2. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-2) 来源参考：**（中文）**[资策会大事纪要，1983年](http://www.iii.org.tw/about/history_1983.html) [互联网档案馆](https://zh.wikipedia.org/wiki/Wayback_Machine)的[存档](https://web.archive.org/web/20101104215940/http:/www.iii.org.tw/about/history_1983.html)，存档日期2010-11-04.
3. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-3) 来源参考：**（中文）**[资策会大事纪要](http://www.iii.org.tw/about/1_6.asp) [互联网档案馆](https://zh.wikipedia.org/wiki/Wayback_Machine)的[存档](https://web.archive.org/web/20101104200611/http:/www.iii.org.tw/about/1_6.asp)，存档日期2010-11-04.
4. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-4) 来源参考：**（中文）**[资策会大事纪要，1984年](http://www.iii.org.tw/about/history_1984.html) [互联网档案馆](https://zh.wikipedia.org/wiki/Wayback_Machine)的[存档](https://web.archive.org/web/20101104215950/http:/www.iii.org.tw/about/history_1984.html)，存档日期2010-11-04.
5. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-5) 来源参考：**（中文）**[行政院主计处电子处理数据中心中文全字库——中文码介绍-BIG-5码介绍](http://www.cns11643.gov.tw/AIDB/encodings.do#encode4)
6. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-6) 来源参考：**（中文）**[数字杂谈 - 中文的电脑或电脑的中文？，谌家雄](http://www.libertytimes.com.tw/2001/new/mar/11/today-i1.htm) [互联网档案馆](https://zh.wikipedia.org/wiki/Wayback_Machine)的[存档](https://web.archive.org/web/20070222072342/http:/www.libertytimes.com.tw/2001/new/mar/11/today-i1.htm)，存档日期2007-02-22.
7. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-7) 来源参考：**（中文）**[CMEX财团法人中文数字化技术推广基金会 - 认识中文码：九、Big5和Big5E](http://www.cmex.org.tw/page.do?path=service_1-2.html)
8. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-8) 来源参考：**（中文）**[与文字共舞\_\_\_中文数字化发展简介](http://www.stat.gov.tw/ct.asp?xItem=26971&ctNode=99)[[失效链接](https://zh.wikipedia.org/wiki/Wikipedia:%E5%A4%B1%E6%95%88%E9%93%BE%E6%8E%A5)]
9. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-9) 来源参考：倚天中文系统、国乔中文系统
10. [**跳转^**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-10) 来源参考：**（中文）**[朱邦复专栏——中文微电脑之历史见证](http://www.cbflabs.com/book/essay/agdi0.htm)
11. ^ [跳转至：**11.0**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-bigF1_11-0) [**11.1**](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC#cite_ref-bigF1_11-1) 参考：**（英文）**[perlop－Perl operators and precedence](http://linux.tnc.edu.tw/CPAN/perl/pod/perlop.html)以及**（中文）**[浅谈许盖功-中文冲码问题](http://www.twvbb.com/vbb/archive/t-163.html)

* Lunde, Ken. CJKV Information Processing First Edition. O'Reilly and Associates, Inc. 1999. [ISBN 978-1-56592-224-2](https://zh.wikipedia.org/wiki/Special:%E7%BD%91%E7%BB%9C%E4%B9%A6%E6%BA%90/9781565922242).
* **（中文）**《中文字码 万码奔腾 一码当先》作者：黄大一，永麒科技出版

# 8.4 GB2312码

* 1. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-1) 1965年《[印刷通用汉字字形表](https://zh.wikipedia.org/wiki/%E5%8D%B0%E5%88%B7%E9%80%9A%E7%94%A8%E6%BC%A2%E5%AD%97%E5%AD%97%E5%BD%A2%E8%A1%A8" \o "印刷通用汉字字形表)》将“锺”（使用简化偏旁）注明为“作为人名地名及引用文言文的时候仍有需要”；2013年《[通用规范汉字表](https://zh.wikipedia.org/wiki/%E9%80%9A%E7%94%A8%E8%A6%8F%E7%AF%84%E6%BC%A2%E5%AD%97%E8%A1%A8" \o "通用规范汉字表)》收录“锺”，可用於姓名
  2. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-gb2312-80_3-0) [GB 2312-1980: Information technology—Chinese ideogram coded character set for information interchange (basic set)](https://archive.org/details/GB2312-1980). [2 October 2016].
  3. ^ [跳转至：**3.0**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-many-mappings_4-0) [**3.1**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-many-mappings_4-1) Haible, Bruno. [GB2312 (Conversion Tables)](http://haible.de/bruno/charsets/conversion-tables/GB2312.html). [29 September 2016].
  4. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-5) [Readme - MAPPINGS/OBSOLETE/EASTASIA](ftp://ftp.unicode.org/Public/MAPPINGS/OBSOLETE/EASTASIA/ReadMe.txt). 9 August 2001 [29 September 2016].
  5. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-6) [java-EUC\_CN-1.3\_P.ucm](https://ssl.icu-project.org/trac/browser/data/trunk/charset/data/ucm/java-EUC_CN-1.3_P.ucm). [29 September 2016].
  6. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-7) [libiconv:lib/gb2312.h](http://git.savannah.gnu.org/cgit/libiconv.git/tree/lib/gb2312.h" \l "n26). GNU Savannah. [29 September 2016].
  7. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-8) [Issue 24036](http://bugs.python.org/issue24036). Python Bug Tracker.
  8. [**跳转^**](https://zh.wikipedia.org/wiki/GB_2312#cite_ref-9) [Encoding § Names and labels](https://www.w3.org/TR/encoding/#names-and-labels). W3C. [29 September 2016].

# 8.5 UTF-8码

1.[**跳转^**](https://zh.wikipedia.org/wiki/UTF-8#cite_ref-1) 参考RFC 2277 section 3.1

2.^ [跳转至：**2.0**](https://zh.wikipedia.org/wiki/UTF-8#cite_ref-faq_2-0) [**2.1**](https://zh.wikipedia.org/wiki/UTF-8#cite_ref-faq_2-1) [UTF-8 and Unicode FAQ for Unix/Linux by Markus Kuhn](http://www.cl.cam.ac.uk/~mgk25/unicode.html)

# 8.6 UTF-16码

* 1. [**跳转^**](https://zh.wikipedia.org/wiki/UTF-16#cite_ref-1) [Code in Apache Xalan 2.7.0 which can fail on surrogate pairs](http://www.google.com/codesearch/p?hl=en&sa=N&cd=3&ct=rc#WiHOcNHN_BU/xalan-j_2_7_0/src/org/apache/xalan/lib/ExsltStrings.java&q=file:ExsltStrings\.java%20align). Apache Foundation. The code wrongly assumes it is safe to use substring on the input
  2. [**跳转^**](https://zh.wikipedia.org/wiki/UTF-16#cite_ref-2) Python 2.6 decode of UTF16 does this on Linux, and it correctly handles surrogate pairs. All "CESU" decoders do it too, though they also mistranslate correct surrogate pairs into 2 characters