**Unicode编码**

**Unicode**（统一码、万国码、单一码）是一种在计算机上使用的字符编码。它为每种语言中的每个字符设定了统一并且唯一的[二进制](http://baike.baidu.com/view/18536.htm)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，1994年正式公布。随着计算机工作能力的增强，Unicode也在面世以来的十多年里得到普及。

## [一、](http://baike.baidu.com/view/40801.htm)简介

　　Unicode 是基于通用[字符集](http://baike.baidu.com/view/51987.htm)（Universal Character Set）的标准来发展，并且同时也以书本的形式（The Unicode Standard，目前第五版由Addison-Wesley Professional出版，ISBN-10: 0321480910）对外发表。

　　2006年7[月](http://baike.baidu.com/view/75273.htm)的最新版本的 Unicode 是5.0版本。 2005年[3月31日](http://baike.baidu.com/view/477558.htm)推出的Unicode 4.1.0 。另外，5.0 Beta于2005年12月12日推出，5.2版本（unicode standard）于2009年10月1日正式推出，以供各会员评价。

　　目前Unicode标准，6.1版已发布（2012年1月31日）。在unicode联盟网站上可以查看完整的6.1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。

## 二、Unicode 的编码和实现

　　大概来说，Unicode 编码系统可分为[编码方式](http://baike.baidu.com/view/1084820.htm)和实现方式两个层次。

### 2.1编码方式

　　Unicode是[国际组织](http://baike.baidu.com/view/10475.htm)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0-0x10FFFF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。[UTF-8](http://baike.baidu.com/view/25412.htm)、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（或称ISO/IEC 10646）标准所定义的标准字符集。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](http://baike.baidu.com/view/42488.htm)（ISO）和多语言软件制造商组成的统一码联盟。前者开发的 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](http://baike.baidu.com/view/1127103.htm)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），每行有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。代理区的目的用两个UTF-16字符表示BMP以外的字符。在介绍UTF-16编码时会介绍。

　　如前所述在Unicode 5.0.0版本中，238605-65534\*2-6400-2408=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。

### 2.2实现方式

　　在Unicode中：汉字“字”对应的数字是23383。在Unicode中，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](http://baike.baidu.com/view/497266.htm)、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。

### 2.3 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](http://baike.baidu.com/view/3001329.htm)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。

### 2.4 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](http://baike.baidu.com/view/18536.htm)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。

### 2.5 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

　　那么，怎么判断字节流的字节序呢？Unicode标准建议用BOM（Byte Order Mark）来区分字节序，即在传输字节流前，先传输被作为BOM的字符"零宽无中断空格"。这个字符的编码是FEFF，而反过来的FFFE（UTF-16）和FFFE0000（UTF-32）在Unicode中都是未定义的码位，不应该出现在实际传输中。下表是各种UTF编码的BOM：

　　UTF编码　║　Byte Order Mark

　　UTF-8　║　EF BB BF

　　UTF-16LE　║　FF FE

　　UTF-16BE　║　FE FF

　　UTF-32LE　║　FF FE 00 00

　　UTF-32BE　║　00 00 FE FF

UCS-2 编码范围

　　Unicode 到目前为止所定义的五个平面中,第0平面(BMP)最为重要. 其编码分布如下.

　　注：中文范围 4E00-9FBF：CJK 统一表意符号 (CJK Unified Ideographs)

　　0000-007F：C0控制符及基本拉丁文 (C0 Control and Basic Latin)

　　0080-00FF：C1控制符及拉丁文补充-1 (C1 Control and Latin 1 Supplement)

　　0100-017F：拉丁文扩展-A (Latin Extended-A)

　　0180-024F：拉丁文扩展-B (Latin Extended-B)

　　0250-02AF：国际音标扩展 (IPA Extensions)

　　02B0-02FF：空白修饰字母 (Spacing Modifiers)

　　0300-036F：结合用读音符号 (Combining Diacritics Marks)

　　0370-03FF：希腊文及科普特文 (Greek and Coptic)

0400-04FF：[西里尔字母](http://baike.baidu.com/view/701246.htm) (Cyrillic)

　　0500-052F：西里尔字母补充 (Cyrillic Supplement)

　　0530-058F：亚美尼亚语 (Armenian)

　　0590-05FF：希伯来文 (Hebrew)

　　0600-06FF：阿拉伯文 (Arabic)

　　0700-074F：[叙利亚](http://baike.baidu.com/view/7851.htm)文 (Syriac)

　　0750-077F：阿拉伯文补充 (Arabic Supplement)

　　0780-07BF：[马尔代夫](http://baike.baidu.com/view/4258.htm)语 (Thaana)

　　07C0-077F：[西非](http://baike.baidu.com/view/307601.htm)书面语言 (N'Ko)

　　0800-085F：阿维斯塔语及[巴列维语](http://baike.baidu.com/view/2091409.htm) (Avestan and Pahlavi)

　　0860-087F：Mandaic

　　0880-08AF：[撒马利亚](http://baike.baidu.com/view/72950.htm)语 (Samaritan)

　　0900-097F：[天城](http://baike.baidu.com/view/1815622.htm)文书 (Devanagari)

　　0980-09FF：孟加拉语 (Bengali)

　　0A00-0A7F：锡克教文 (Gurmukhi)

　　0A80-0AFF：[古吉拉特](http://baike.baidu.com/view/360869.htm)文 (Gujarati)

　　0B00-0B7F：奥里亚文 (Oriya)

　　0B80-0BFF：泰米尔文 (Tamil)

　　0C00-0C7F：泰卢固文 (Telugu)

　　0C80-0CFF：卡纳达文 (Kannada)

　　0D00-0D7F：德拉维族语 (Malayalam)

　　0D80-0DFF：僧伽罗语 (Sinhala)

　　0E00-0E7F：泰文 (Thai)

　　0E80-0EFF：[老挝](http://baike.baidu.com/view/7471.htm)文 (Lao)

　　0F00-0FFF：藏文 (Tibetan)

　　1000-109F：缅甸语 (Myanmar)

　　10A0-10FF：[格鲁吉亚语](http://baike.baidu.com/view/360506.htm) (Georgian)

　　1100-11FF：朝鲜文 (Hangul Jamo)

　　1200-137F：埃塞俄比亚语 (Ethiopic)

　　1380-139F：埃塞俄比亚语补充 (Ethiopic Supplement)

　　13A0-13FF：切罗基语 (Cherokee)

　　1400-167F：统一加拿大土著语音节 (Unified Canadian Aboriginal Syllabics)

　　1680-169F：欧甘字母 (Ogham)

　　16A0-16FF：如[尼文](http://baike.baidu.com/view/4150793.htm) (Runic)

　　1700-171F：塔加拉语 (Tagalog)

　　1720-173F：Hanunóo

　　1740-175F：Buhid

　　1760-177F：Tagbanwa

　　1780-17FF：高棉语 (Khmer)

　　1800-18AF：蒙古文 (Mongolian)

　　18B0-18FF：Cham

　　1900-194F：Limbu

　　1950-197F：[德宏](http://baike.baidu.com/view/186721.htm)泰语 (Tai Le)

　　1980-19DF：新傣仂语 (New Tai Lue)

　　19E0-19FF：高棉语记号 (Kmer Symbols)

　　1A00-1A1F：Buginese

　　1A20-1A5F：Batak

　　1A80-1AEF：Lanna

　　1B00-1B7F：[巴厘](http://baike.baidu.com/view/90943.htm)语 (Balinese)

　　1B80-1BB0：巽他语 (Sundanese)

　　1BC0-1BFF：Pahawh Hmong

　　1C00-1C4F：[雷布](http://baike.baidu.com/view/4846130.htm)查语(Lepcha)

　　1C50-1C7F：Ol Chiki

　　1C80-1CDF：[曼尼普尔语](http://baike.baidu.com/view/474489.htm) (Meithei/Manipuri)

　　1D00-1D7F：语音学扩展 (Phonetic Extensions)

　　1D80-1DBF：语音学扩展补充 (Phonetic Extensions Supplement)

　　1DC0-1DFF：结合用读音符号补充 (Combining Diacritics Marks Supplement)

　　1E00-1EFF：拉丁文扩充附加 (Latin Extended Additional)

　　1F00-1FFF：希腊语扩充 (Greek Extended)

　　2000-206F：常用[标点](http://baike.baidu.com/view/302346.htm) (General Punctuation)

　　2070-209F：上标及下标 (Superscripts and Subscripts)

　　20A0-20CF：货币符号 (Currency Symbols)

　　20D0-20FF：组合用记号 (Combining Diacritics Marks for Symbols)

　　2100-214F：字母式符号 (Letterlike Symbols)

　　2150-218F：数字形式 (Number Form)

　　2190-21FF：箭头 (Arrows)

　　2200-22FF：数学运算符 (Mathematical Operator)

　　2300-23FF：杂项工业符号 (Miscellaneous Technical)

　　2400-243F：控制图片 (Control Pictures)

　　2440-245F：光学识别符 (Optical Character Recognition)

　　2460-24FF：封闭式字母数字 (Enclosed Alphanumerics)

　　2500-257F：制表符 (Box Drawing)

　　2580-259F：方块元素 (Block Element)

　　25A0-25FF：几何图形 (Geometric Shapes)

　　2600-26FF：杂项符号 (Miscellaneous Symbols)

　　2700-27BF：印刷符号 (Dingbats)

　　27C0-27EF：杂项数学符号-A (Miscellaneous Mathematical Symbols-A)

　　27F0-27FF：追加箭头-A (Supplemental Arrows-A)

　　2800-28FF：盲文点字模型 (Braille Patterns)

　　2900-297F：追加箭头-B (Supplemental Arrows-B)

　　2980-29FF：杂项数学符号-B (Miscellaneous Mathematical Symbols-B)

　　2A00-2AFF：追加数学运算符 (Supplemental Mathematical Operator)

　　2B00-2BFF：杂项符号和箭头 (Miscellaneous Symbols and Arrows)

　　2C00-2C5F：[格拉哥里字母](http://baike.baidu.com/view/557723.htm) (Glagolitic)

　　2C60-2C7F：拉丁文扩展-C (Latin Extended-C)

　　2C80-2CFF：[古埃及](http://baike.baidu.com/view/8498.htm)语 (Coptic)

　　2D00-2D2F：格鲁吉亚语补充 (Georgian Supplement)

　　2D30-2D7F：提非纳文 (Tifinagh)

　　2D80-2DDF：埃塞俄比亚语扩展 (Ethiopic Extended)

　　2E00-2E7F：追加标点 (Supplemental Punctuation)

　　2E80-2EFF：CJK 部首补充 (CJK Radicals Supplement)

　　2F00-2FDF：康熙字典部首 (Kangxi Radicals)

　　2FF0-2FFF：表意文字描述符 (Ideographic Description Characters)

　　3000-303F：CJK 符号和标点 (CJK Symbols and Punctuation)

　　3040-309F：日文平假名 (Hiragana)

　　30A0-30FF：日文片假名 (Katakana)

　　3100-312F：注音字母 (Bopomofo)

　　3130-318F：朝鲜文兼容字母 (Hangul Compatibility Jamo)

　　3190-319F：象形字注释标志 (Kanbun)

　　31A0-31BF：注音字母扩展 (Bopomofo Extended)

　　31C0-31EF：CJK 笔画 (CJK Strokes)

　　31F0-31FF：日文片假名语音扩展 (Katakana Phonetic Extensions)

　　3200-32FF：封闭式 CJK 文字和月份 (Enclosed CJK Letters and Months)

　　3300-33FF：CJK 兼容 (CJK Compatibility)

　　3400-4DBF：CJK 统一表意符号扩展 A (CJK Unified Ideographs Extension A)

　　4DC0-4DFF：[易经六十四卦](http://baike.baidu.com/view/16312.htm)符号 (Yijing Hexagrams Symbols)

　　4E00-9FBF：CJK 统一表意符号 (CJK Unified Ideographs)

　　A000-A48F：彝文音节 (Yi Syllables)

　　A490-A4CF：彝文字根 (Yi Radicals)

　　A500-A61F：Vai

　　A660-A6FF：统一加拿大土著语音节补充 (Unified Canadian Aboriginal Syllabics Supplement)

　　A700-A71F：声调修饰字母 (Modifier Tone Letters)

　　A720-A7FF：拉丁文扩展-D (Latin Extended-D)

　　A800-A82F：Syloti Nagri

　　A840-A87F：八思巴字 (Phags-pa)

　　A880-A8DF：Saurashtra

　　A900-A97F：爪哇语 (Javanese)

　　A980-A9DF：Chakma

　　AA00-AA3F：Varang Kshiti

　　AA40-AA6F：Sorang Sompeng

　　AA80-AADF：Newari

　　AB00-AB5F：[越南](http://baike.baidu.com/view/4205.htm)傣语 (Vi?t Thái)

　　AB80-ABA0：Kayah Li

　　AC00-D7AF：朝鲜文音节 (Hangul Syllables)

　　D800-DBFF：High-half zone of UTF-16

　　DC00-DFFF：Low-half zone of UTF-16

　　E000-F8FF：自行使用区域 (Private Use Zone)

　　F900-FAFF：CJK 兼容象形文字 (CJK Compatibility Ideographs)

　　FB00-FB4F：字母表达形式 (Alphabetic Presentation Form)

　　FB50-FDFF：阿拉伯表达形式A ([Arabic Presentation Form-A](http://baike.baidu.com/view/521080.htm))

　　FE00-FE0F：变量选择符 (Variation Selector)

　　FE10-FE1F：竖排形式 (Vertical Forms)

　　FE20-FE2F：组合用半符号 (Combining Half Marks)

　　FE30-FE4F：CJK 兼容形式 (CJK Compatibility Forms)

　　FE50-FE6F：小型变体形式 (Small Form Variants)

　　FE70-FEFF：阿拉伯表达形式B ([Arabic Presentation Form-B](http://baike.baidu.com/view/521076.htm))

　　FF00-FFEF：半型及全型形式 (Halfwidth and Fullwidth Form)

　　FFF0-FFFF：特殊 (Specials)

## 三、非 Unicode 环境

　　在非 Unicode 环境下，由于不同国家和地区采用的字符集不一致，很可能出现无法正常显示所有字符的情况。[微软](http://baike.baidu.com/view/2353.htm)公司使用了代码页（[Codepage](http://baike.baidu.com/view/1428668.htm)）转换表的技术来过渡性的部分解决这一问题，即通过指定的转换表将非 Unicode 的字符编码转换为同一字符对应的系统内部使用的 Unicode 编码。可以在“语言与区域设置”中选择一个代码页作为非 Unicode 编码所采用的默认[编码方式](http://baike.baidu.com/view/1084820.htm)，如936为简体中文GBK，950为正体中文Big5（皆指PC上使用的）。在这种情况下，一些非英语的[欧洲](http://baike.baidu.com/view/3622.htm)语言编写的[软件](http://baike.baidu.com/view/37.htm)和文档很可能出现[乱码](http://baike.baidu.com/view/47.htm)。而将代码页设置为相应语言中文处理又会出现问题，这一情况无法避免。从根本上说，完全采用统一编码才是解决之道，但目前尚无法做到这一点。

　　代码页技术现在广泛为各种平台所采用。UTF-7 的[代码页](http://baike.baidu.com/view/684997.htm)是65000，UTF-8 的代码页是65001。

## 四、XML 和 Unicode

[XML](http://baike.baidu.com/view/63.htm)及其子集[HTML](http://baike.baidu.com/view/692.htm)采用UTF-8作为标准字集，理论上我们可以在各种支持XML标准的[浏览器](http://baike.baidu.com/view/7718.htm)上显示任何地区文字的[网页](http://baike.baidu.com/view/828.htm)，只要[电脑](http://baike.baidu.com/view/2358.htm)本身安装有合适的[字体](http://baike.baidu.com/view/285104.htm)即可。可以利用&#nnn;的格式显示特定的字符。nnn代表该字符的十进制 Unicode 代码。如果采用十六进制代码，在编码之前加上x字符即可。但部分旧版本的浏览器可能无法识别十六进制代码。

　　然而部分由于 Unicode 版本发展原因，很多浏览器只能显示 UCS-2 完整[字符集](http://baike.baidu.com/view/51987.htm)也即现在使用的 Unicode 版本中的一个小子集。下表可以检验您的浏览器怎样显示各种各样的 Unicode 代码：

　　代码 字符标准名称 (英语) 在浏览器上的显示

　　A&#大写拉丁字母"A" A

　　&#szlig; 小写[拉丁字母](http://baike.baidu.com/view/107325.htm)"Sharp S" &szlig;

　　&#thorn; 小写[拉丁](http://baike.baidu.com/view/38715.htm)[字母](http://baike.baidu.com/view/94782.htm)"Thorn" &thorn;

　　Δ大写[希腊](http://baike.baidu.com/view/6744.htm)字母"Delta" Δ

　　Й 大写[斯拉夫](http://baike.baidu.com/view/253815.htm)字母"Short I" Й

　　ק[希伯来](http://baike.baidu.com/view/176645.htm)字母"Qof" ק

　　م[阿拉伯](http://baike.baidu.com/view/96268.htm)字母 "Meem" م

　　๗[泰文](http://baike.baidu.com/view/1980705.htm)数字 7 ๗

　　ቐ[埃塞俄比亚](http://baike.baidu.com/view/11274.htm)[音节文字](http://baike.baidu.com/view/426727.htm)"Qha" ቐ

　　あ[日语](http://baike.baidu.com/view/20584.htm)[平假名](http://baike.baidu.com/view/489512.htm) "A" あ

　　ア日语[片假名](http://baike.baidu.com/view/62454.htm) "A" ア

　　叶简体[汉字](http://baike.baidu.com/view/1712.htm) "叶" 叶

　　叶 [繁体](http://baike.baidu.com/view/143053.htm)汉字 "叶" 叶

　　엽[韩国](http://baike.baidu.com/view/3299.htm)音节文字 " Yeob" 엽

## 五、输入Unicode

综述

　　除了[输入法](http://baike.baidu.com/view/38725.htm)外，[操作系统](http://baike.baidu.com/view/880.htm)会提供几种方法输入Unicode。像是[Windows 2000](http://baike.baidu.com/view/153378.htm)之后的Windows系统就提供一个可点击的表。例如在[Microsoft Word](http://baike.baidu.com/view/828743.htm)或者金山WPS之下，按下 [Alt](http://baike.baidu.com/view/65582.htm) 键不放，输入 0 和某个字符的 Unicode 编码（[十进制](http://baike.baidu.com/view/359301.htm)），再松开 Alt 键即可得到该字符，如Alt + 033865会得到Unicode字符“叶”（繁体）。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的四位 Unicode 编码进行互相转换。

　　Unicode 编码表反弹

　　0000-0FFF 8000-8FFF 10000-10FFF 20000-20FFF 28000-28FFF

　　1000-1FFF 9000-9FFF 21000-21FFF 29000-29FFF

　　2000-2FFF A000-AFFF 22000-22FFF 2A000-2AFFF

　　3000-3FFF B000-BFFF 23000-23FFF

　　4000-4FFF C000-CFFF 1D000-1DFFF 24000-24FFF 2F000-2FFFF

　　5000-5FFF D000-DFFF 25000-25FFF

　　6000-6FFF E000-EFFF 26000-26FFF

　　7000-7FFF F000-FFFF 27000-27FFF E0000-E0FFF

　　Unicode 目前已经有5.0版本。世界上有一大批计算机、语言学等科学家专门研究Unicode，到了现在Unicode标准已经不单是一个编码标准，还是记录人类语言文字资料的一个巨大的数据库，同时从事人类文化遗产的发掘和保护工作。

　　对于中文而言，Unicode 16编码里面已经包含了[GB18030](http://baike.baidu.com/view/889058.htm)里面的所有汉字（27484个字），目前Unicode标准准备把[康熙字典](http://baike.baidu.com/view/32121.htm)的所有汉字放入到Unicode 32bit编码中。

　　简单地说，Unicode扩展自ASCII字元集。在严格的ASCII中，每个字元用7位元表示，或者电脑上普遍使用的每字元有8位元宽；而Unicode使用全16位元字元集。这使得Unicode能够表示世界上所有的书写语言中可能用於电脑通讯的字元、象形文字和其他符号。Unicode最初打算作为ASCII的补充，可能的话，最终将代替它。考虑到ASCII是电脑中最具支配地位的标准，所以这的确是一个很高的目标。

　　Unicode影响到了电脑工业的每个部分，但也许会对作业系统和[程序设计语言](http://baike.baidu.com/view/128511.htm)的影响最大。从这方面来看，我们已经上路了。Windows NT从底层支持Unicode（不幸的是，Windows 98只是小部分支援Unicode）。先天即被ANSI束缚的[C程序设计语言](http://baike.baidu.com/view/611688.htm)通过对宽字元集的支持来支持Unicode。

输入方法

[中文输入法](http://baike.baidu.com/view/180389.htm)截至2009年3月，可以使用[微软拼音2003](http://baike.baidu.com/view/379977.htm)或2007版本海峰[五笔](http://baike.baidu.com/view/4350.htm)9.3版本，新注音输入法 和 VimIM 进行输入。

　　微软拼音 在输入法启动状态下，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相应十六进制值的方式输入相应文字。例如中文“胥”输入“5066”，朝鲜语文字“셅”输入“c145”（不需要在前面加0x或x）。海峰五笔 此输入法已经直接支持透过五笔码输入方式输入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（Unicode部分）“㗎”为“keks”，CJK扩展B区的“𣿱”为“iyho”和CJK扩展C区的“𫇛”为“muih”。新注音输入法 在输入法启动状态时，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“卡”，则是“`U5CE0”。VimIM 在 Vim 环境中，可以直接键入十进制或十六进制 Unicode 码。既不需要启动输入法，也不需要码表。

[日文输入法](http://baike.baidu.com/view/7230184.htm)使用Microsoft IME 2007，可以在IME PAD里找到UNICODE的点击表。点击字符即可输入。选择[字体](http://baike.baidu.com/view/285104.htm)可以预览字符效果。

　　其他除了输入法外，操作系统也会提供另外几种方法输入 Unicode。像是Windows 2000之后的 Windows 系统就提供一个可点击的字符映射表。又或者在Microsoft Word下，按下 Alt 键不放，输入 0 和某个字符的 Unicode 编码（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的四位 Unicode 编码进行互相转换。

　　[1-2]新建文本也能输入、右键

## 六、使用Unicode原因

基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造Unicode之前，有数百种指定这些数字的编码系统。

没有一个编码可以包含足够的字符：例如，单单欧州共同体就需要好几种不同的编码来包括所有的语言。即使是单一种语言，例如英语，也没有哪一个编码可以适用于所有的字母，[标点](http://baike.baidu.com/view/302346.htm)符号，和常用的技术符号。这些编码系统也会互相冲突。也就是说，两种编码可能使用相同的数字代表两个不同的字符，或使用不同的数字代表相同的字符。

任何一台特定的计算机（特别是服务器）都需要支持许多不同的编码，但是，不论什么时候数据通过不同的编码或平台之间，那些数据总会有损坏的危险。