Unicode

going down the rabbit hole

Unicode - going down the rabbit hole

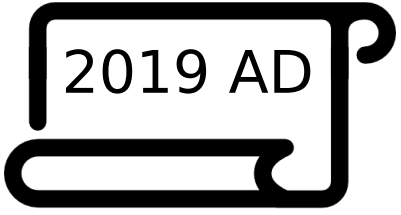
A history of the written word

A history of the computerized word

Unicode

C++ and Unicode

Future perfect



الْحُرُوف الْعَرَبِيَّة

Unicode - going down the rabbit hole

A history of the written word

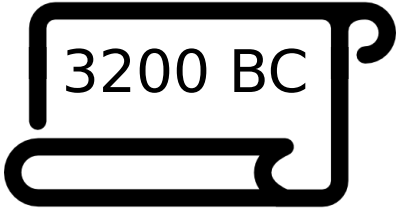
A history of the computerized word

Unicode

C++ and Unicode

Future perfect

الْحُرُوف الْعَرَبِيَّة



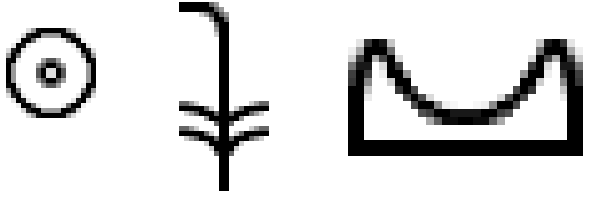
3200 BC

Source: George Hodan (Flickr)

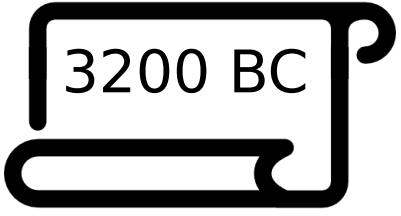
الْحُرُوف الْعَرَبِيَّة

Egyptian Hieroglyphs

- Used from 32nd century BC until 394 AD
- Three kinds of ways to use ‘letters’:
  - Literal
  - Symbolic
  - As a ‘letter’

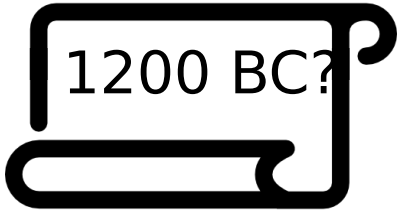


- Written on, or carved into, stone



الْحُرُوف الْعَرَبِيَّة

- 这本书正文有 500 页。
- 我一得到任何消息，就立刻给你发短信。
- 我们能在周二前看到你演讲的全文吗?
- 《尤利西斯》为考试指定的必读书。

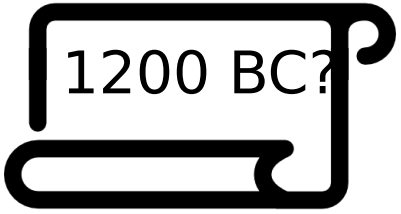


الْحُرُوف الْعَرَبِيَّة

Trick question

- Chinese
- Japanese
- Korean
- Taiwanese

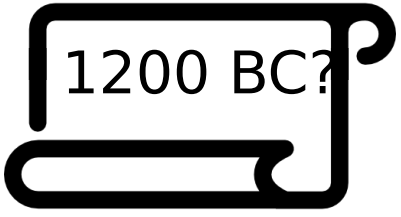
الْحُرُوف الْعَرَبِيَّة



Chinese (Hanzi)

- 漢字 – Han character(s)
- Initially every ‘letter’ was a word
  - Monosyllabic
- Evolved to polysyllabic words
  - Most letters are one syllables

日 木

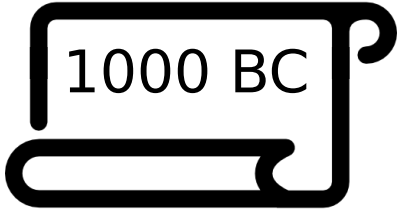


الْحُرُوف الْعَرَبِيَّة

Hebrew

- Stopped being in common use in 400AD
- Reappeared in late 1800s
- Language is read and written right-to-left

סעיף א. כל בני אדם נולדו בני חורין ושווים בערכם
ובזכויותיהם. כולם חוננו בתבונה ובמצפון, לפיכך חובה
עליהם לנהוג איש ברעהו ברוח של אחוה

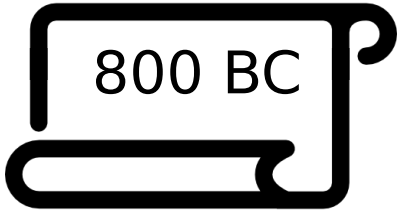


الْحُرُوف الْعَرَبِيَّة

Greek

- Used from 8th century BC on
- Origin of ‘alphabet’ - **alpha, beta**, gamma, …
- Adaptation of the Phoenician alphabet
- Evolved greatly during use

Όλοι οι άνθρωποι γεννιούνται ελεύθεροι

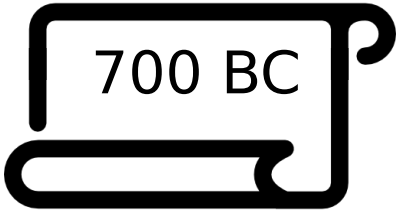


الْحُرُوف الْعَرَبِيَّة

Latin

- The script we all know and use every day
- Multiple letters form a syllable
- Mapping from letters to syllables **very** region and language dependent

الْحُرُوف الْعَرَبِيَّة

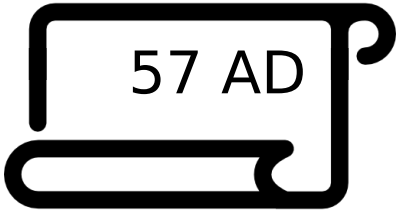


Japanese (Kanji)

- Same character set as Chinese Hanzi
  - Kanji ... also means "Han characters"

問話文動度県水安氏和政保

الْحُرُوف الْعَرَبِيَّة



義とか云つて何ても作者の經驗した悲にも係かる事を聊かも技巧を加へず、有の儘に、だら〳〵と、牛の涎のやうに書くのが流行るさうだ。好い事が流行る。私も矢張り其て行く。

て、題は「平凡」、書方は牛の涎。

さあ、是からが本文だが此處らて回を改めたが好からうと思ふ。

三

私は地方生れだ。戸籍を並べても仕方がないから、唯某縣の某市として置く。其處て生れて其處て育つたのだ。

子供の時分の事は最う大抵忘れて了つたが、不思議なもので覺えてゐる事だと、判然と昨日の事のやうに想はれる事もある。中にも是ばかりは一生目の底に染付いて忘れられまいと思ふのは十の時死別れた祖母の面だ。

今ても目を瞑ると、直ぐ顯然と目の前に浮ぶ。面長の老人だから無論皺は寄つてゐたが締つた口元で、段鼻て、なか〳〵上品な面相だつたが眼が大きな眼て、女には強過る程權が有つて、古屋の——これが私の家の姓だ——古屋の隱居の眼といつたら、隨分評判の眼だつたさうだ。成程然ういへば、何か氣に入らぬ事が有つて祖母が白眼てヂロリと睨むと、子供心にも何だか無氣味だつたやうな覺がまだ有る。

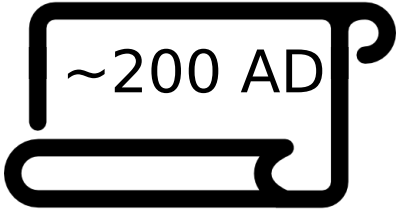
Arabic

Ligature-based language

- Written and read right-to-left

العربية الفصحى

ا ل ع ر ب ي ة ا ل ف ص ح ى

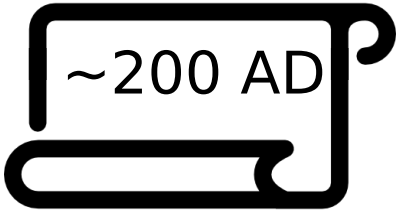


الْحُرُوف الْعَرَبِيَّة

Ligatures

- Shuffle
- High-flying
- Gone fishing
- Sheffield

- Shuffle
- High-flying
- Gone fishing
- Sheffield



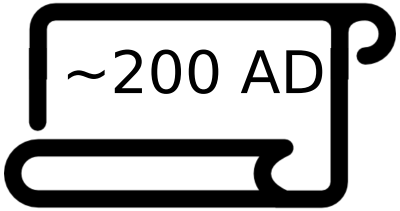
الْحُرُوف الْعَرَبِيَّة

ÃƒÆ'Ã¢â‚¬Å¡Ãƒâ€šÃ‚ ...ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

Ligatures

- ﬄ ffl
- ﬂ fl
- ﬁ fi
- ﬃ ffi

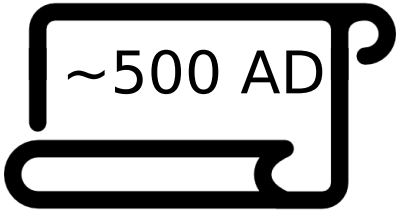
الْحُرُوف الْعَرَبِيَّة



Japanese (Hiragana)

- “Simplified” Kana (letters)
  - Only ~50 letters to learn
- Pretty close to an alphabet
- Old forms of Hiragana deprecated
  - Still used, name changed to Hentaigana

はっぴょうけっか　ちょっかん　ひっし

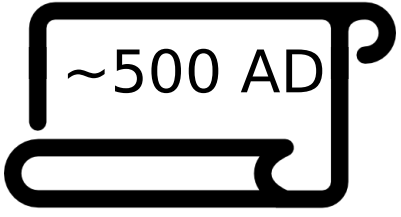


Japanese (Katakana)

- Used for transliterating foreign words

カタカナ – Ka-ta-ka-na

ペテル ビヌデルス　- Peter Bindels (approx.)



الْحُرُوف الْعَرَبِيَّة

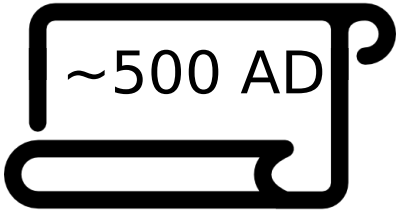
Japanese (Katakana)

- Used for transliterating foreign words

カタカナ – Ka-ta-ka-na

ペテ**ル** ビヌデ**ル**ス　- Pete**r** Binde**l**s (approx.)

الْحُرُوف الْعَرَبِيَّة



Ogham (᚛ᚑᚌᚐᚋ᚜)

- Irish written language
- Usually carved into the edge of stone
- Only language with non-empty space symbol

~500 AD

الْحُرُوف الْعَرَبِيَّة

Hangul (Korean)

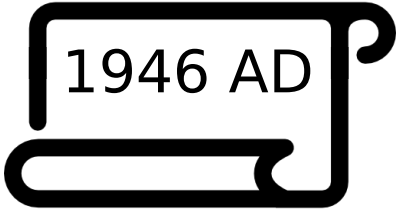
- Korea used Hanja (Chinese Hanzi) before
- Letters are not the units you think
- ㄲㅜㄹㅂㅓㄹ → 꿀벌

세계를 향한 대화 , 유니코드로 하십시오 .

الْحُرُوف الْعَرَبِيَّة

Japanese (Kanji)

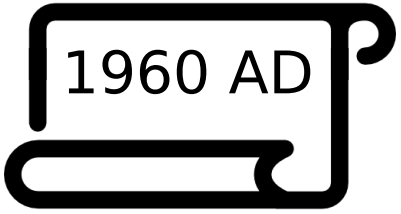
- Writing of common characters simplified after WWII
  - “Shinjitai”, as opposed to old “kyūjitai”



الْحُرُوف الْعَرَبِيَّة

Simplified Chinese (简化字 ; jiǎnhuàzì)

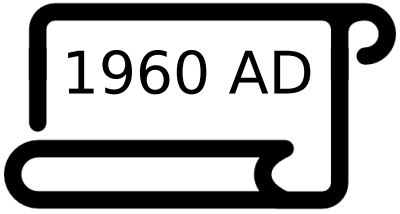
- People’s Republic of China created
- Similar to Japan’s simplifications, but different



الْحُرُوف الْعَرَبِيَّة

- 栈 – stack (Hanzi / kyūjitai Kanji)
- 栈 – stack (Shinjitai Kanji)
- 栈 – stack (Simplified Chinese)

الْحُرُوف الْعَرَبِيَّة



ÃƒÆ’Ã¢â‚¬Å¡Ãƒâ€šÂ ... sit amet, consectetur adipiscing elit, sed do eiusmod tem

Unicode - going down the rabbit hole

A history of the written word

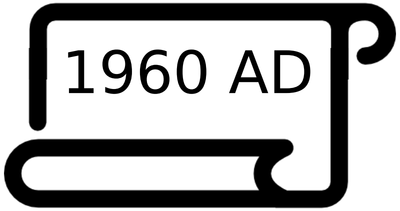
A history of the computerized word

Unicode

C++ and Unicode

Future perfect

الْحُرُوفُ الْعَرَبِيَّة



Unicode - going down the rabbit hole

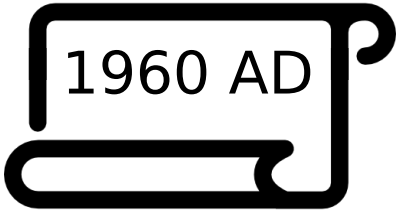
A history of the written word

A history of the computerized word

Unicode

C++ and Unicode

Future perfect

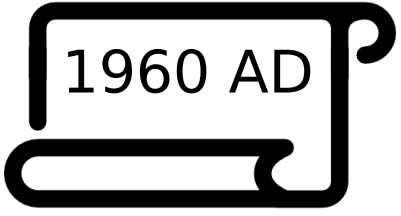


الْحُرُوف الْعَرَبِيَّة

Basic classification

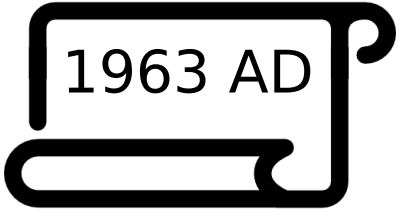
- Single-byte encodings
  - ASCII compatible
- Multi-byte encodings
- Variable-length encodings
  - Self-synchronizing
  - Non-self-synchronizing

الْحُرُوف الْعَرَبِيَّة

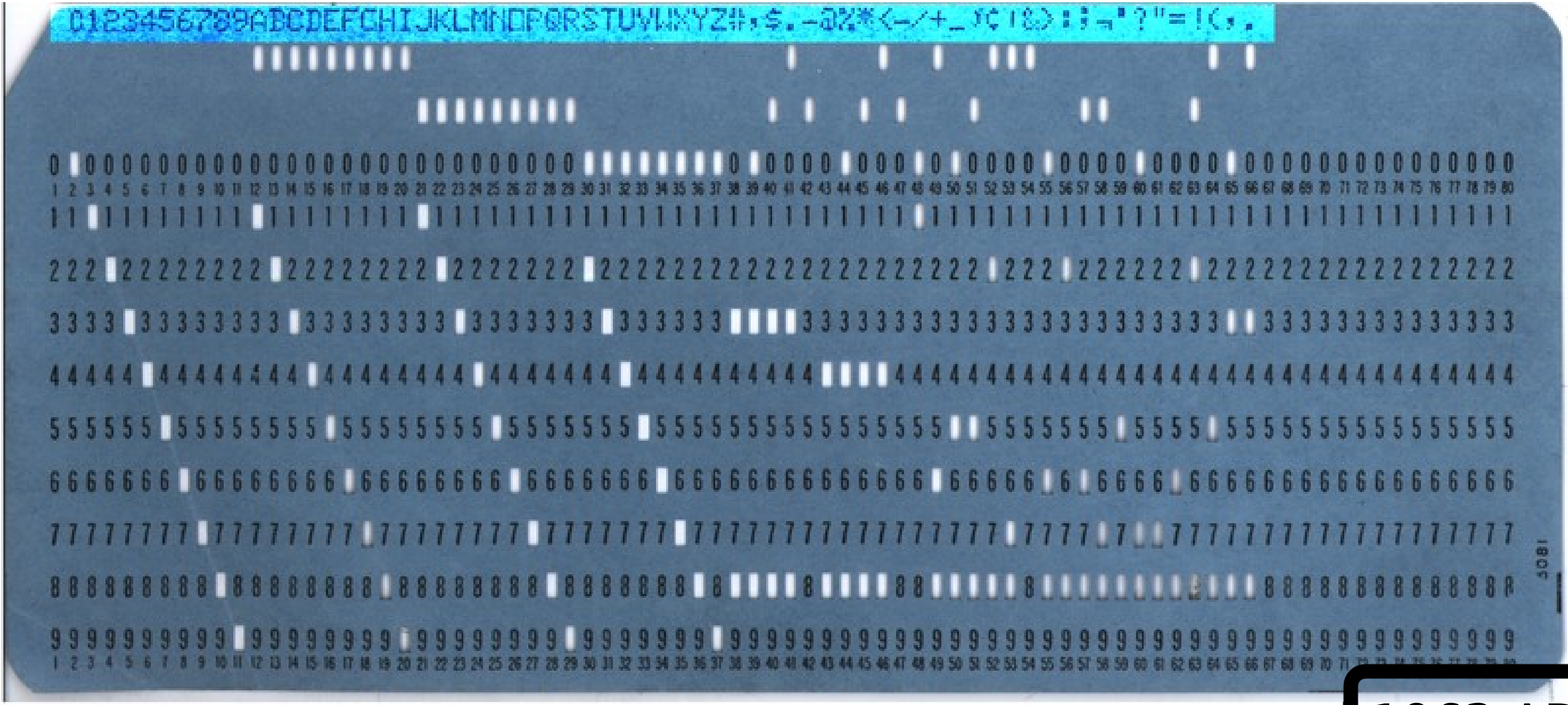


Simple Latin encodings

- ASCII
  - American Standard Code for Information Interchange
  - 7-bit, all characters in logical order
- EBCDIC
  - Extended from Binary-Coded Decimal (BCD) Interchange Code
  - 8-bit, compatible with punch cards



ÃƒÆ'Ã¢â‚¬Å¡Ãƒâ€šÂ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem



الْحُرُوفُ الْعَرَبِيَّة

1963 AD

ÃƒÆ’Ã¢â‚¬Å¡Ãƒâ€šÂ... ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

GB2312

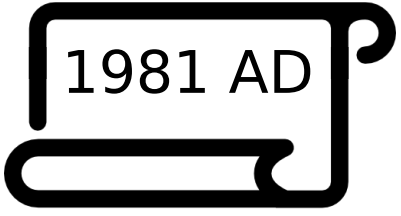
- 8-bit, bottom half is ASCII
- Variable length, multibyte character set
- Guojia Biaozhun (国家标准 , national standard)
- Chinese character set

الحروف العربية

1980 AD

ANSI

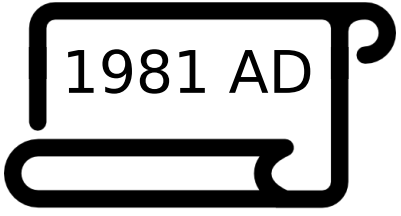
- ANSI didn’t have anything to do with it
- Also often called “Extended ASCII”
- Code Page 437 from DOS
- 8-bit, bottom half is ASCII
- Top half filled with line drawing and some math symbols



الْحُرُوف الْعَرَبِيَّة

DOS CP437

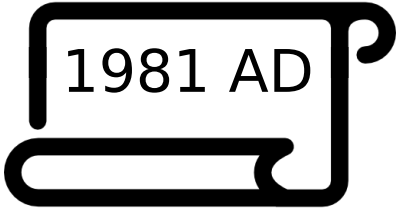
- ANSI didn’t have anything to do with it
- Also often called “Extended ASCII”
- Code Page 437 from DOS
- 8-bit, bottom half is ASCII
- Top half filled with line drawing and some math symbols



الْحُرُوف الْعَرَبِيَّة

DOS CP866

- 8-bit, bottom half is ASCII
- Top half filled with line drawing and cyrillic letters

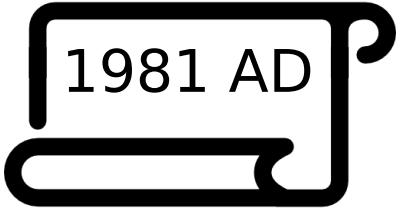


الْحُرُوف الْعَرَبِيَّة

DOS CP850

- 8-bit, bottom half is ASCII
- Top half filled with line drawing and accented letters

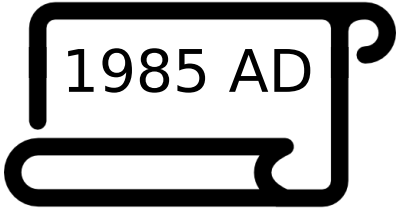
الْحُرُوف الْعَرَبِيَّة



Windows 1252

- 8-bit, bottom half is ASCII
- Top half filled with text processor symbols, accented letters and other things

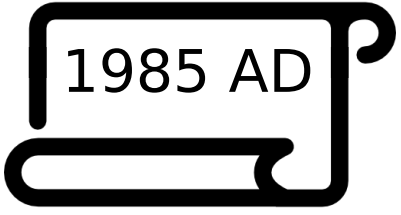
and yes, LaTeX calls this “newansi”.



الْحُرُوف الْعَرَبِيَّة

Windows 1255

- 8-bit, bottom half is ASCII
- Top half filled with Hebrew letters

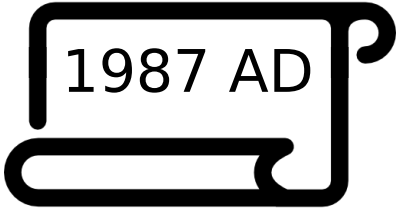


الْحُرُوف الْعَرَبِيَّة

ISO 8859

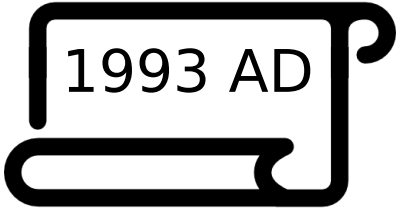
- Standardize 8-bit code pages
  - 15 defined
  - Cannot mix use

الْحُرُوف الْعَرَبِيَّة



GBK

- Extension of GB2312
- Just adds way more Chinese characters



الْحُرُوف الْعَرَبِيَّة

Unicode - going down the rabbit hole

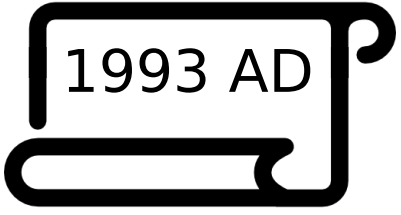
A history of the written word

A history of the computerized word

Unicode

C++ and Unicode

Future perfect



الْحُرُوف الْعَرَبِيَّة

Unicode - going down the rabbit hole

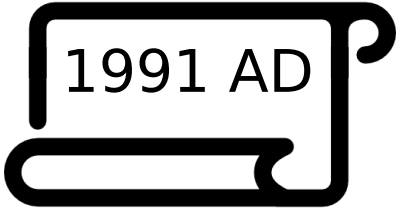
A history of the written word

A history of the computerized word

Unicode

C++ and Unicode

Future perfect



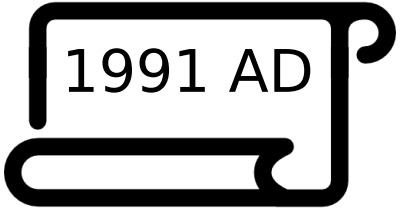
الْحُرُوف الْعَرَبِيَّة

Unicode

Unicode is intended to address the need for a workable, reliable world text encoding.

Unicode could be roughly described as "wide-body ASCII" that has been stretched to 16 bits to encompass the characters of all the world's living languages. In a properly engineered design, 16 bits per character are more than sufficient for this purpose.

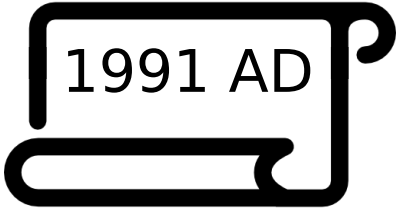
- Joe Becker



Unicode

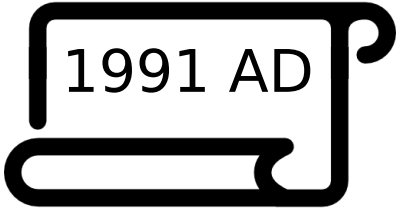
- encodes graphemes rather than glyphs
- "wide-body ASCII" stretched to 16 bits
- incorporated on 3 January 1991
- first volume published in October 1991

الْحُرُوف الْعَرَبِيَّة



UCS2

- 16-bit, bottom bit is ASCII
- Intended to contain all the world’s languages
- No more conversions
- Byte order dependent
  - Byte Order Mark (BOM) recommended at start

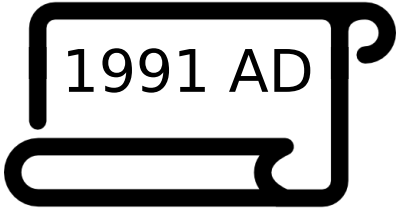


الْحُرُوف الْعَرَبِيَّة

UTF-7

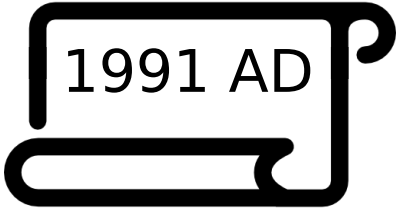
- Not ASCII compatible per se
- Intended to transport Unicode across (7-bit) ASCII-only communication channels

الْحُرُوف الْعَرَبِيَّة



UTF-8

- ASCII compatible
- Intended as a temporary encoding
  - C-style functions with 8-bit encoding support
  - More efficient than UCS2 for ASCII
  - Less efficient than UCS2 for all Eastern languages
  - Variable-length

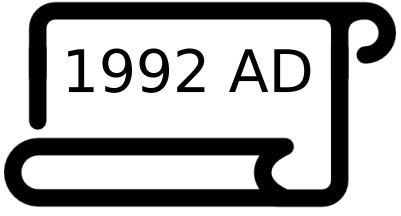


الْحُرُوف الْعَرَبِيَّة

Han unification

- 16-bit character set
- GBK has 21,886
- Hanja, Hanzi, Kanji and Simplified Chinese...

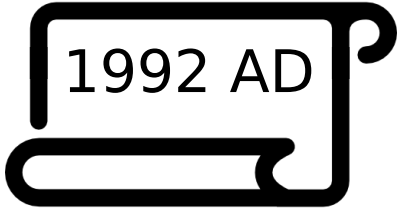
الْحُرُوف الْعَرَبِيَّة



Han unification

- Not all characters unified
- Most characters no issue
- Some have different glyphs, but the same grapheme now...

الْحُرُوف الْعَرَبِيَّة



Code point	Chinese (simplified) (zh-Hans)	Chinese (traditional) (zh-Hant)	Chinese (traditional, Hong Kong) (zh-Hant-HK)	Japanese (ja)	Korean (ko)	Vietnamese (vi-Hani)	English
U+4ECA	今	今	今	今	今	今	now
U+4EE4	令	令	令	令	令	令	cause/command
U+514D	免	免	免	免	免	免	exempt/spare
U+5165	入	入	入	入	入	入	enter
U+5168	全	全	全	全	全	全	all/total
U+5177	具	具	具	具	具	具	tool
U+5203	刃	刃	刃	刃	刃	刃	knife edge
U+5316	化	化	化	化	化	化	transform/change
U+5916	外	外	外	外	外	外	outside
U+60C5	情	情	情	情	情	情	feeling
U+624D	才	才	才	才	才	才	talent
U+62B5	抵	抵	抵	抵	抵	抵	arrive/resist
U+6B21	次	次	次	次	次	次	secondary/follow

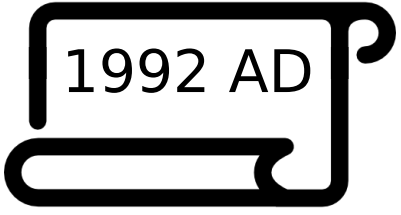
الْحُرُوف الْعَرَبِيَّة

1992 AD

What is a character

- ɑ or a ?
- a or a ?
- 丢 vs 丢 ?

الْحُرُوف الْعَرَبِيَّة



ÃƒÆ'Ã¢â‚¬Å¡Ãƒâ€šÂ... ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

What is a character

- ɑ or a ? → Same thing, different glyph
- a or a ? → Different thing, same glyph
- 丟 vs 丢 ? → Different thing

الحروف العربية

1992 AD

Wingdings

1990 AD

الْحُرُوف الْعَرَبِيَّة

Wingdings

1990 AD

الْحُرُوف الْعَرَبِيَّة

Emoji

- e (絵 , "picture") + moji (文字 , "character")
- Popular because of instant-messaging and texting
- 

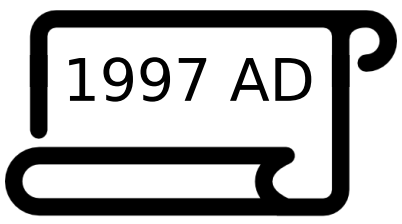
1997 AD

الْحُرُوف الْعَرَبِيَّة


2017 AD

Controversies around emoji

- 🔫 is mostly replaced with a water gun
- 💣, 🗡️ and 🔫 lend themselves really well to crime
- 🍑 and 🍆 used to represent anatomy instead of a fruit and a vegetable
- Lawsuits threatened for allowing use of 🖕



الْحُرُوف الْعَرَبِيَّة

ÃƒÆ’Ã¢â‚¬Å¡Ãƒâ€šÂ ... sum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

Complicated emoji

e + ´ = é

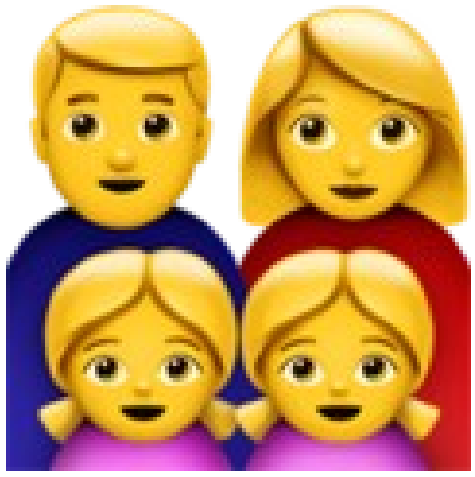
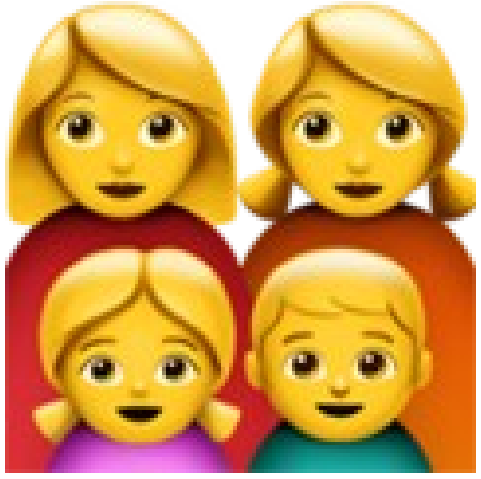
🌾 + 👩 = 👩‍🌾

🕴 + 🏾 + ♀ = 🕴🏾‍♀️

الحُرُوف العَرَبِيّة

1997 AD

Emoji

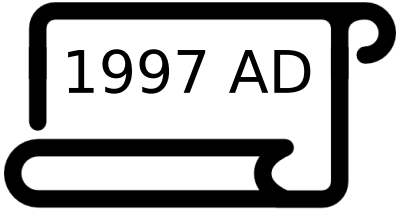




الْحُرُوف الْعَرَبِيَّة

1997 AD

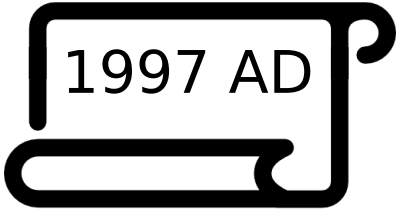
UTF-16

- 16-bit, bottom bit is ASCII
- Surrogate pairs create characters not in UCS2
- Byte order dependent
  - Byte Order Mark (BOM) recommended at start
- Variable length
- Cannot encode code points beyond 1’114’112



UTF-32 (UCS-4)

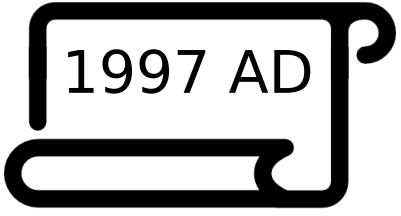
- 32-bit, bottom bit is ASCII
- Surrogate pair characters invalid
- Byte order dependent
  - Byte Order Mark (BOM) recommended at start
- Fixed length
- Very inefficient for any text
- May not contain code points inaccessible in UTF-16



UTF-8

- ASCII compatible
- Intended as a temporary encoding

- No byte order dependency
- Pretty efficient, only Asian languages are less efficient than UTF-16
- May not contain code points inaccessible in UTF-16

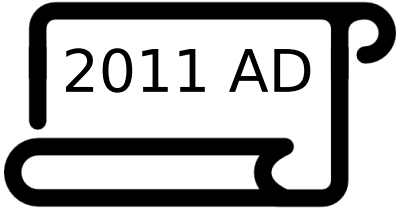


الحروف العربية

Encoding unicode

- Storing UTF-16 as UTF-8
  - CESU-8
- Storing UTF-16 with null bytes as UTF-8
  - Modified UTF-8

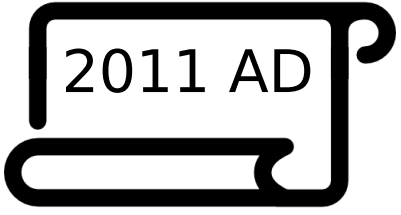
الْحُرُوف الْعَرَبِيَّة



Encoding unicode

- But what if my system uses EBCDIC?
  - UTF-EBCDIC
- We can’t just switch from GBK
  - GB18030

الْحُرُوف الْعَرَبِيَّة



ÃƒÆ’Ã¢â‚¬Å¡Ãƒâ€šÃ‚ ... ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

Mojibake

- The result of decoding text in one encoding, with a different encoding
  - Most common: UTF8 read as some 1-byte encoding
  - Billy Oâ€™Neill

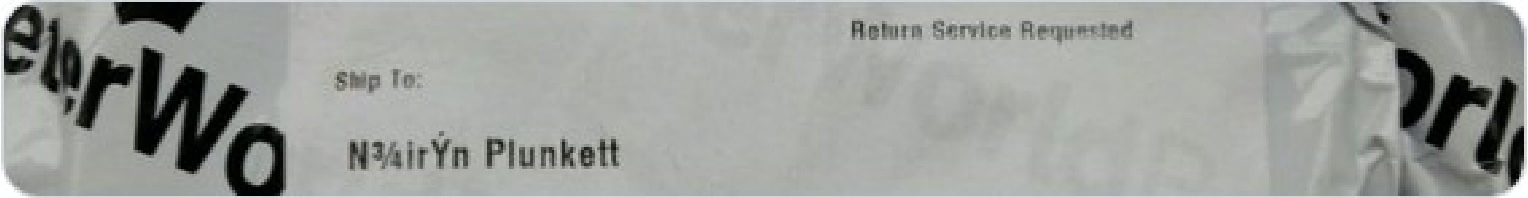
الْحُرُوفُ الْعَرَبِيَّة

2011 AD

ÃƒÆ’Ã¢â‚¬Å¡Ãƒ â€šÂ‚Â Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

Nóirín Plunkett
@noirinp

Didn't realise that in 2015, there were still ways to misencode my name that I hadn't seen before.

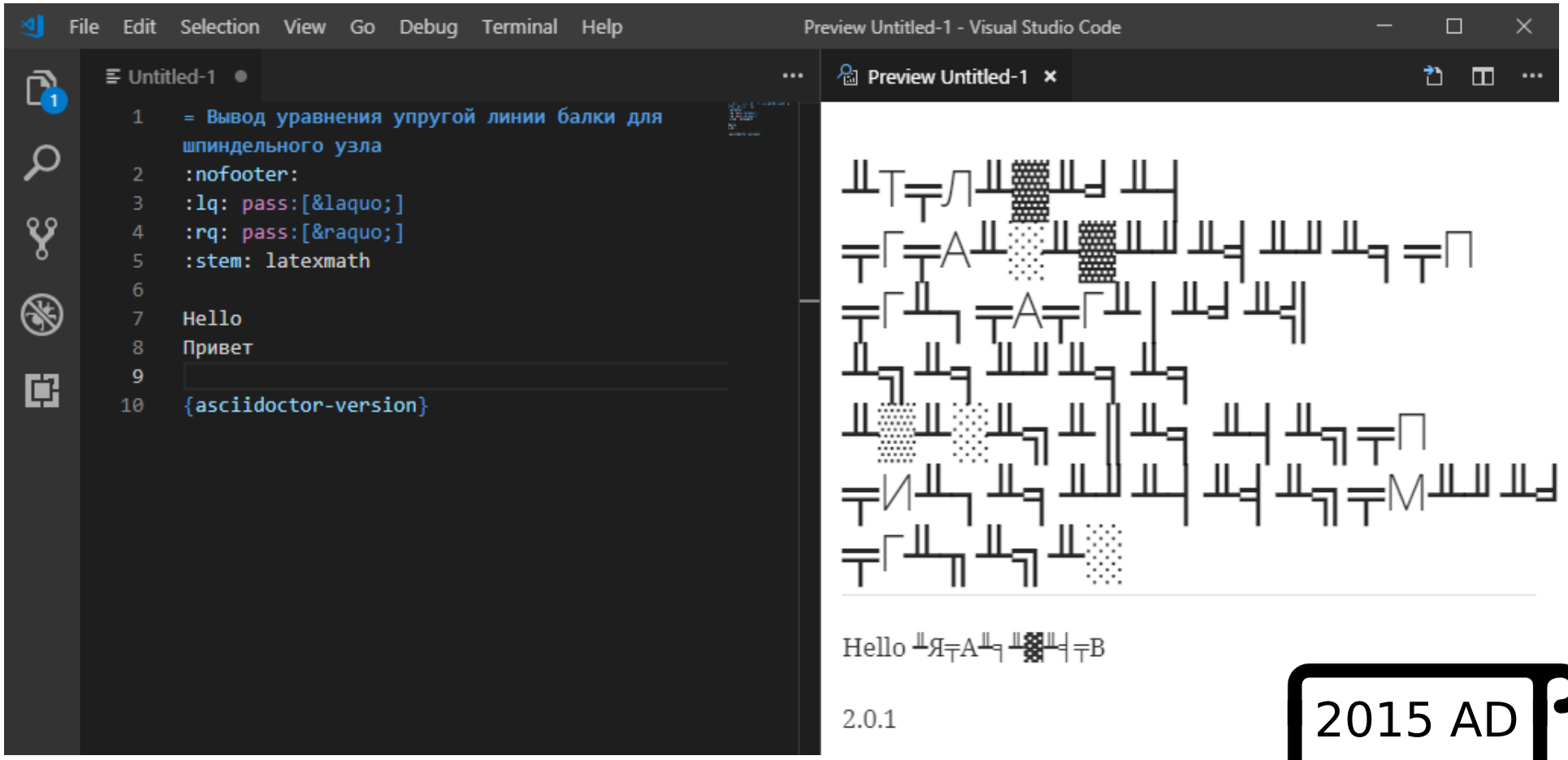


9:49 PM · May 19, 2015 · Twitter for Android

1 Retweet **3** Likes

الْحُرُوفُ الْعَرَبِيَّةُ

2015 AD



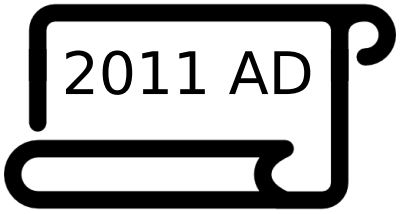
"This letter was sent to a Russian student by her French friend, who manually wrote the address that she received by e-mail." Mojibake diacritics translated to Cyrillic by the postal employees via The New Aesthetic

2015 AD

الْحُرُوف الْعَرَبِيَّة

Displaying unicode

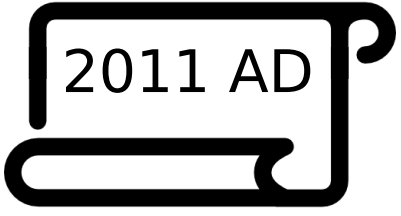
- Displaying most encodings is simple:
  - Map each character to a glyph
  - Calculate positions for each glyph
  - Render glyphs at positions



Displaying unicode

- Unicode is different
  - Combining diacritics
  - RTL languages – sometimes mixed!
  - Korean, Devanagari, … complex characters
  - Ligatures

الْحُرُوف الْعَرَبِيَّة



Displaying unicode

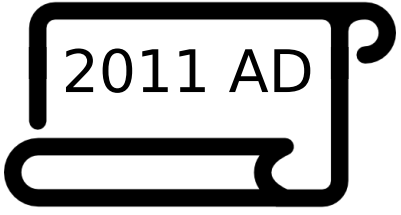
1)Decode to unicode code points

2)Normalize code points

3)Extract grapheme clusters

4)Shape to determine glyphs & positions

5)Render glyphs at appropriate positions



الْحُرُوف الْعَرَبِيَّة

ÃƒÆ’Ã¢â‚¬Å¡Ãƒâ€šÂ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

Unicode abuse

- Zalgo text
  - Take regular text (Lorem ipsum)
  - Add some combining diacritics (Łöřêm̀ íṗşųṁ)
  - Add tons more (Lorem ipsum)
  - And even more (Lorem ipsum)

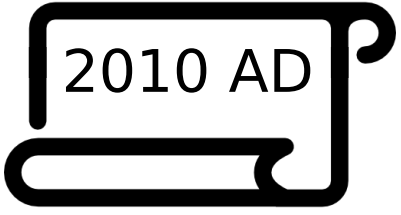
الْحُرُوف الْعَرَبِيَّة

2010 AD

Unicode abuse

- ¯_(ツ)_/¯
  - Starts with macron (U+00AF), backslash, underscore, open parentheses, Hiragana character, closing parentheses, underscore, slash, macron

الْحُرُوف الْعَرَبِيَّة



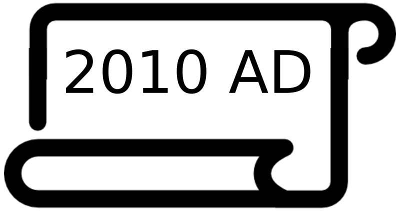
ÃƒÆ’Ã¢â‚¬Å¡Ãƒâ€šÂ,Â ... sum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem

Unicode abuse

Just take letters from other languages that look like Latin upside-down.

˙uʍop-ǝpᴉsdn uᴉʇɐ˥ ǝʞᴉl ʞool ʇɐɥʇ sǝƃɐnƃuɐl
ɹǝɥʇo ɯoɹɟ sɹǝʇʇǝl ǝʞɐʇ ʇsnſ

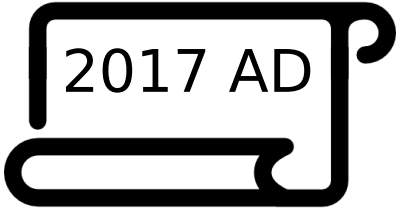
الحروف العربية



Unicode abuse

- Using imitation letters to resemble something else
- Often abused in domain names
- https://www.apple.com
- https://www.apple.com

الْحُرُوف الْعَرَبِيَّة



Unicode abuse

- Using imitation letters to resemble something else
- Often abused in domain names
- https://www.apple.com → Actually Apple
- https://www.apple.com → Cyrillic letters

الْحُرُوف الْعَرَبِيَّة

Unicode - going down the rabbit hole

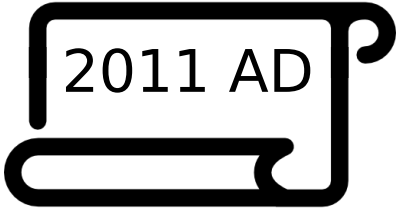
A history of the written word

A history of the computerized word

Unicode

C++ and Unicode

Future perfect



الْحُرُوف الْعَرَبِيَّة

Unicode - going down the rabbit hole

A history of the written word

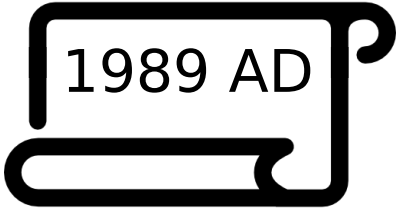
A history of the computerized word

Unicode

C++ and Unicode

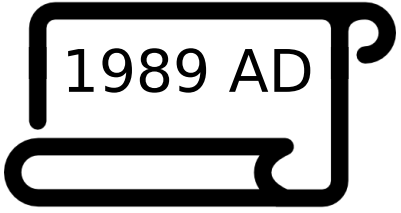
Future perfect

الْحُرُوف الْعَرَبِيَّة



C++ in 1989

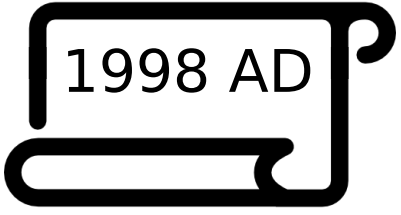
- No support
  - Remember, unicode doesn’t get created until 1991



الْحُرُوف الْعَرَبِيَّة

C++ in 1998

- std::string
  - Current locale with 8-bit characters
- std::wstring
  - Current wide locale, with sizeof(wchar_t) characters
    - 16 bit on Windows, 32 bit on Unix
  - wchar_t definition: big enough to hold any character in the encoding

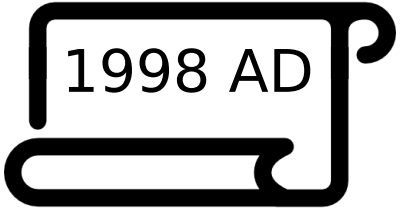


الْحُرُوف الْعَرَبِيَّة

C++ in 1998

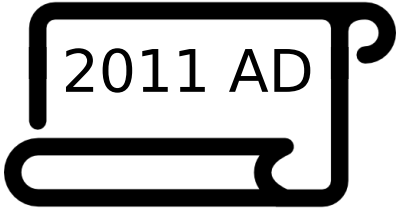
- Not much to handle unicode
  - C95 gave us what we have
- wcstombs
- mbstowcs

الْحُرُوف الْعَرَبِيَّة



C++ in 2011

- std::u16string, char16_t
  - UTF16 string
- std::u32wstring, char32_t
  - UTF32 string
- std::regex
  - No Unicode considerations
- No UTF8 string or character type

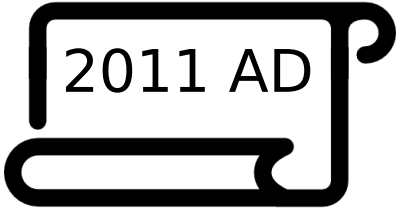


الْحُرُوف الْعَرَبِيَّة

C++ in 2011

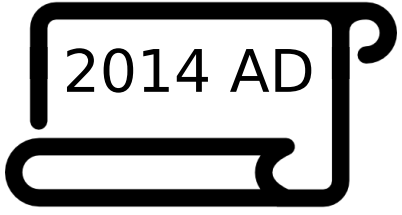
- uchar.h, cuchar
  - Multibyte to u16string, u32string and reverse

الْحُرُوف الْعَرَبِيَّة



C++ in 2014

- ...

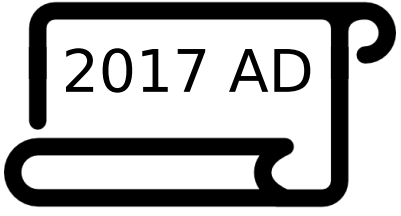


الْحُرُوف الْعَرَبِيَّة

C++ in 2017

- u8"", u"" and U"" string literals
  - resulting in char[], char16_t[] and char32_t[]

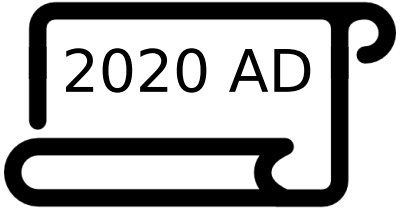
- SG16 formed to handle proper Unicode in C++
  - ISO C++ committee study group



الْحُرُوف الْعَرَبِيَّة

C++ in 2020

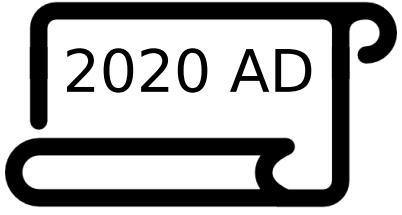
- std::u8string, char8_t
  - UTF8 string and character type
- u8”” changed to char8_t[]
  - Breaking change
- uchar.h, cuchar
  - Multibyte to u8string and reverse



C++ in 2020

- std::string<T>
  - No easy construction from mbs or wcs inputs
  - No logical output conversion for printf, std::cout, ...

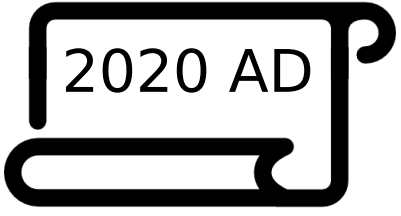
الْحُرُوف الْعَرَبِيَّة



C++ in 2020

```cpp
#include <string>
#include <iostream>

int main() {
    std::u16string s = u"Hello ";
    std::u32string s2 = U"CppCon ";
    std::u8string s3 = u8"World ";
    std::cout << s;
    std::cout << s2;
    std::cout << s3;
}
```

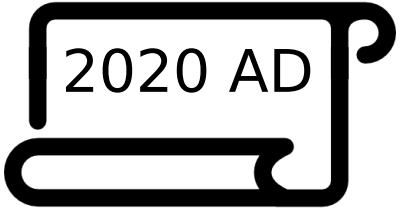


الْحُرُوف الْعَرَبِيَّة

C++ in 2020

```cpp
#include <string>
#include <iostream>

int main() {
    std::u16string s = u"Hello ";
    std::u32string s2 = U"CppCon ";
    std::u8string s3 = u8"World ";
    std::wcout << s;
    std::wcout << s2;
    std::wcout << s3;
}
```



الْحُرُوف الْعَرَبِيَّة

Displaying unicode

1)Decode – <mostly not in C++>

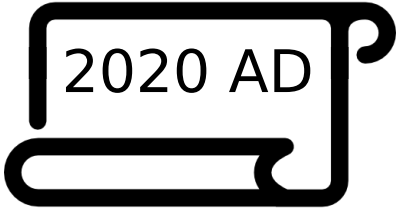
2)Normalize – <not in C++>

3)Extract graphemes – <not in C++>

4)Shape – Harfbuzz, or similar

5)Render – Freetype, or similar

الْحُرُوف الْعَرَبِيَّة



Unicode - going down the rabbit hole

A history of the written word

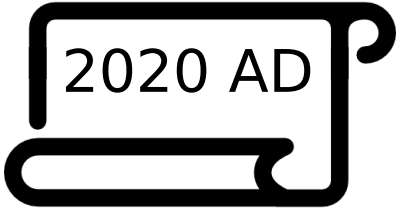
A history of the computerized word

Unicode

C++ and Unicode

Future perfect

الْحُرُوف الْعَرَبِيَّة



Unicode - going down the rabbit hole

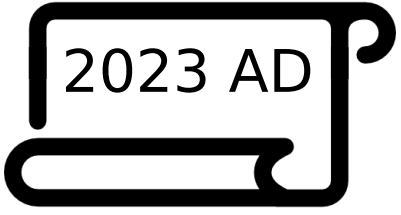
A history of the written word

A history of the computerized word

Unicode

C++ and Unicode

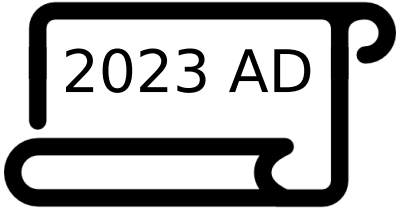
Future perfect



الْحُرُوف الْعَرَبِيَّة

SG16 plans

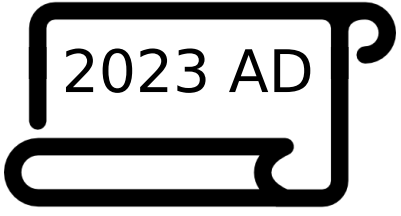
- C++20: Lay the groundwork
  - char8_t needs to be in
  - u8”” needs to produce char8_t[]
  - Lots of detail fixes
  - Get references to Unicode standards (ISO10646) in the C++ standard



الْحُرُوف الْعَرَبِيَّة

SG16 plans

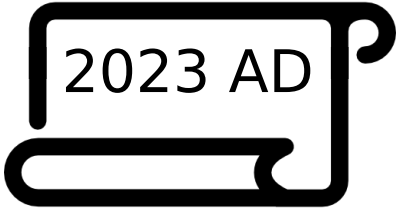
- C++23: Introduce text capable types
  - Text: a string that knows code points and graphemes
  - Enable unicode support for other parts of the standard
    - std::regex is unfixable
    - Hana Dusíková's CTRE library slated for C++23 with unicode support
- https://wg21.link/p1238 (Unicode Direction paper)



SG16 plans

- When to convert?
  - Current proposals: all internal logic is UTF8
  - Check your legacy encodings at the border
    - (convert whenever you receive data or send it out)
  - Preferably, normalize at the border too
    - But need normalizing during string edits too

الْحُرُوف الْعَرَبِيَّة



UTF-8 Everywhere

Manifesto

1 Purpose of this document

This document contains special characters. Without proper rendering support, you may see question marks, boxes, or other symbols.

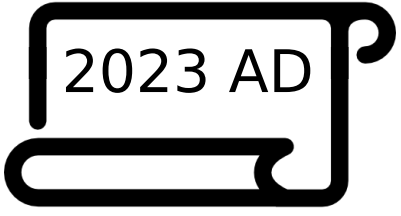
Our goal is to promote usage and support of the UTF-8 encoding and to convince that it should be the default choice of encoding for storing text strings in memory or on disk, for communication and all other uses. We believe that our approach improves performance, reduces complexity of software and helps prevent many Unicode-related bugs. We suggest that other encodings of Unicode (or text, in general) belong to rare edge-cases of optimization and should be avoided by mainstream users.

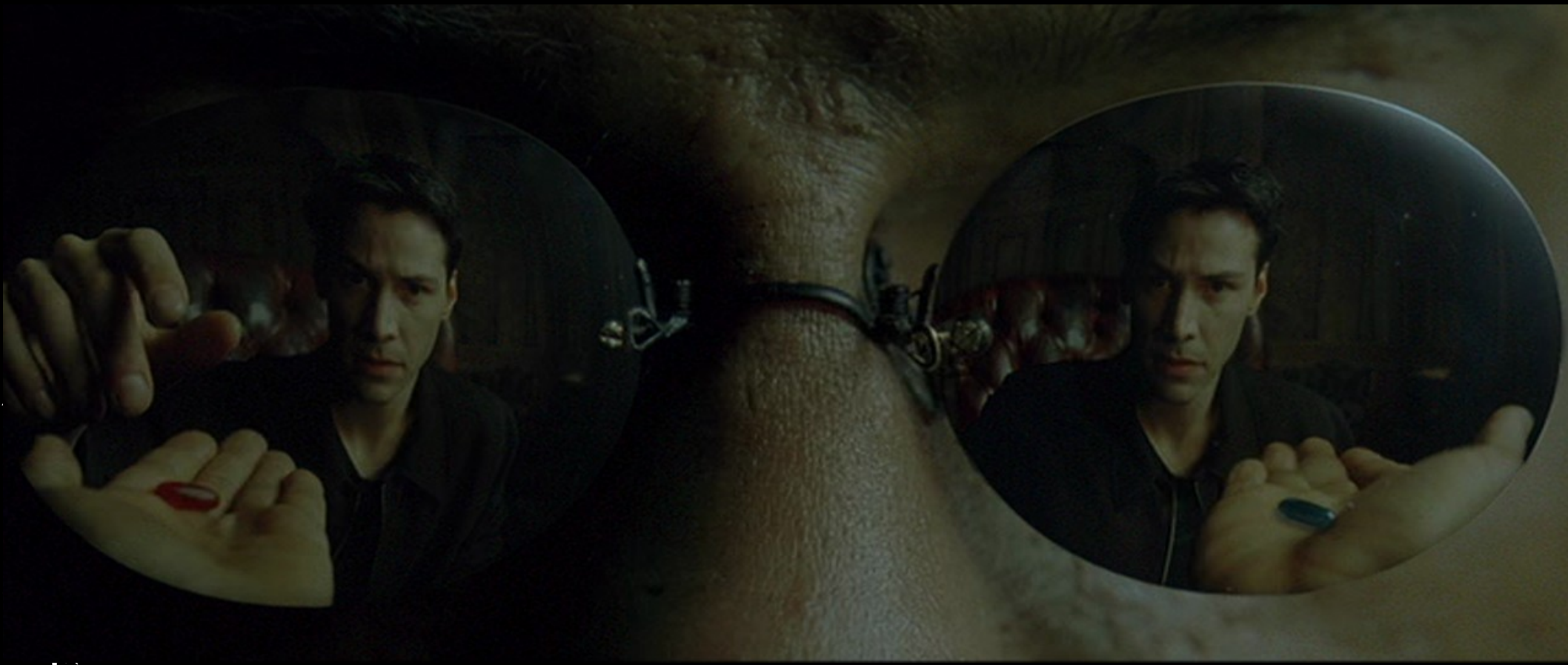
In particular, we believe that the very popular UTF-16 encoding (often mistakenly referred to as 'widechar' or simply 'Unicode' in the Windows world) has no place in library APIs except for specialized text processing libraries, e.g.

SG16

- For any questions about SG16’s current trajectory, ask on our mailing list
  - unicode@isocpp.open-std.org
- Organisational stuff
  - Ask Tom Honermann, chair of SG16
  - <tom@honermann.net>

الْحُرُوف الْعَرَبِيَّة





This was

Unicode

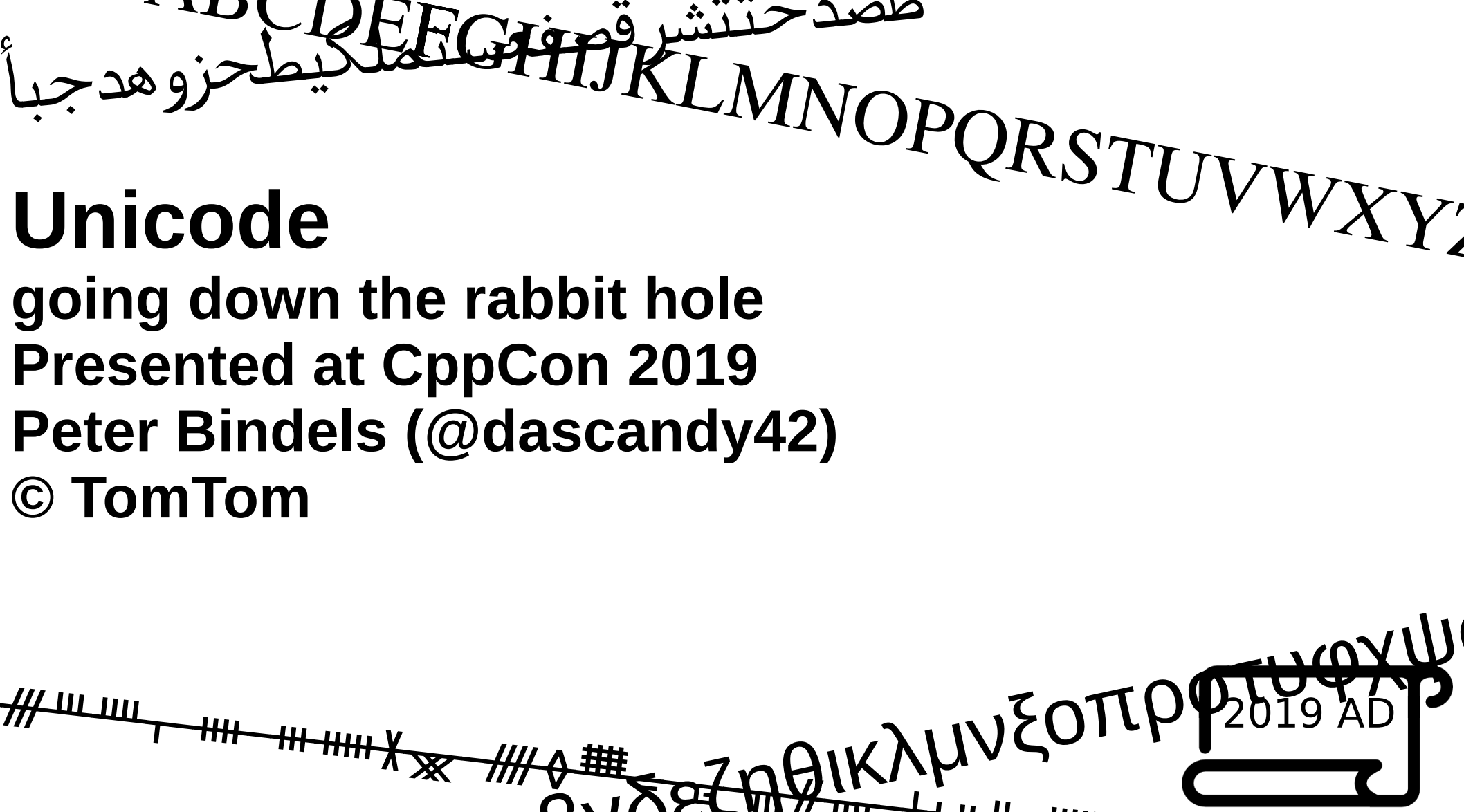
going down the rabbit hole

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ضصدحتشرقصفسثمكيطحزوهدجبأ

2019 AD


2019 AD



Unicode

going down the rabbit hole

Presented at CppCon 2019
Peter Bindels (@dascandy42)