Unicode **Character Code**

A **character** is the **smallest possible component of a text** (e.g., ‘A’, ‘B’, ‘È’ and ‘Í’) that has semantic value.

Even the extended (8 bit) version of *ASCII* is not enough for international use.

The Unicode standard (http://www.unicode.org/) describes **how characters are represented** by unique **code points**. A code point is an **integer value**, usually denoted in base 16. Values range from **0** through **0x10FFFF** (1,114,111 decimal).

The notation **U+**12CA is used to **denote** the character with **value** 0x12ca (4,810 decimal).

The Unicode standard contains tables listing **characters** and their corresponding **code points**:

```
0061    'a'; LATIN SMALL LETTER A
0062    'b'; LATIN SMALL LETTER B
0063    'c'; LATIN SMALL LETTER C
...
007B    '{'; LEFT CURLY BRACKET
```

Unicode was designed to be an ***ASCII*-super set**: the first 256 characters in the *Unicode character* set are identical to those in the extended *ASCII* code.

http://www.unicode.org/charts/

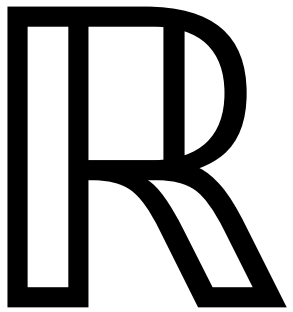
Unicode 7.0 Character Code Charts

SCRIPTS | SYMBOLS | NOTES

Find chart by hex code: Go Related links: Name index Help & links

Scripts

European Scripts	African Scripts	South Asian Scripts
Armenian	Bamum	Bengali and Assamese
Armenian Ligatures	Bamum Supplement	Brahmi
Caucasian Albanian	Bassa Vah	Chakma
Cypriot Syllabary	Coptic	Devanagari
Cyrillic	Coptic in Greek block	Devanagari Extended
Cyrillic Supplement	Coptic Epact Numbers	Grantha
Cyrillic Extended-A	Egyptian Hieroglyphs (1MB)	Gujarati
Cyrillic Extended-B	Ethiopic	Gurmukhi
Elbasan	Ethiopic Supplement	Kaithi
Georgian	Ethiopic Extended	Kannada
Georgian Supplement	Ethiopic Extended-A	Kharoshthi
Glagolitic	Mende Kikakui	Khojki
Gothic	Meroitic	Khudawadi
Greek	Meroitic Cursive	Lepcha
Greek Extended	Meroitic Hieroglyphs	Limbu
Latin	N'Ko	Mahajani
Basic Latin (ASCII)	Osmanya	Malayalam
Latin-1 Supplement	Tifinagh	Meetei Mayek
Latin Extended-A	Vai	Meetei Mayek Extensions
Latin Extended-B	**Middle Eastern Scripts**	Modi
Latin Extended-C	Arabic	Mro
Latin Extended-D	Arabic Supplement	Ol Chiki
Latin Extended-E	Arabic Extended-A	Oriya
Latin Extended Additional	Arabic Presentation Forms-A	Saurashtra
Latin Ligatures	Arabic Presentation Forms-B	Sharada
Fullwidth Latin Letters	Aramaic, Imperial	Siddham
Linear A	Avestan	Sinhala
Linear B	Carian	Sinhala Archaic Numbers
Linear B Syllabary	Cuneiform (1MB)	Sora Sompeng
Linear B Ideograms	Cuneiform Numbers and Punctuation	Syloti Nagri
Ogham	Old Persian	Takri
Old Italic	Ugaritic	Tamil
Old Permic	Hebrew	Telugu
Phaistos Disc	Hebrew Presentation Forms	Thaana
Runic	Lycian	Tirhuta
Shavian	Lydian	Vedic Extensions
Phonetic & Shorthand Symbols	Mandaic	Warang Citi
Duployan	Nabataean	**Southeast Asian Scripts**
Shorthand Format Controls	Old North Arabian	Cham
IPA Extensions	Old South Arabian	Kayah Li



U+211D

Unicode Data	
Name	DOUBLE-STRUCK CAPITAL R
Block	Letterlike Symbols
Category	Letter, Uppercase [Lu]
Combine	0
BIDI	Left-to-Right [L]
Decomposition	<font> LATIN CAPITAL LETTER R (U+0052)
Mirror	N
Old name	DOUBLE-STRUCK R
Index entries	numbers, real R, DOUBLE-STRUCK CAPITAL real numbers set of real numbers, the
Comments	the set of real numbers
Version	Unicode 1.1.0 (June, 1993)

in Python(3):

```
>>> print("\N{DOUBLE-STRUCK CAPITAL R}")
ℝ
>>> print("\u211D")
ℝ

>>> ord("\u211D")
8477
>>> chr(8477)
'ℝ'
```

Unicode code points

```
>>> ord('€')
8364

>>> hex(ord('€'))
'0x20ac'

>>> chr(8364)
'€'

>>> import unicodedata
>>> unicodedata.name('€')
'EURO SIGN'

>>> unicodedata.lookup('EURO SIGN')
'€'

>>> unicodedata.category('€')   # http://www.fileformat.info/info/unicode/category/index.htm
'Sc'                            # [S]ymbol [c]urrency
```

Archaic letters

0370 Ͱ GREEK CAPITAL LETTER HETA
→ 2C75 Ⱶ latin capital letter half h

0371 ͱ GREEK SMALL LETTER HETA
→ 2C76 ⱶ latin small letter half h

0372 Ͳ GREEK CAPITAL LETTER ARCHAIC SAMPI

0373 ͳ GREEK SMALL LETTER ARCHAIC SAMPI

Numeral signs

0374 ʹ GREEK NUMERAL SIGN
= dexia keraia
• indicates numeric use of letters
→ 02CA ˊ modifier letter acute accent
≡ 02B9 ʹ modifier letter prime

0375 ͵ GREEK LOWER NUMERAL SIGN
= aristeri keraia
• indicates numeric use of letters
→ 02CF ˏ modifier letter low acute accent

Archaic letters

0376 Ͷ GREEK CAPITAL LETTER PAMPHYLIAN DIGAMMA

0377 ͷ GREEK SMALL LETTER PAMPHYLIAN DIGAMMA

Iota subscript

037A ͺ GREEK YPOGEGRAMMENI
= iota subscript
→ 0345 ◌ͅ combining greek ypogegrammeni
≈ 0020 SP 0345 ◌ͅ

Lowercase of editorial symbols

037B ͻ GREEK SMALL REVERSED LUNATE SIGMA SYMBOL

037C ͼ GREEK SMALL DOTTED LUNATE SIGMA SYMBOL

037D ͽ GREEK SMALL REVERSED DOTTED LUNATE SIGMA SYMBOL

Punctuation

037E ; GREEK QUESTION MARK
= erotimatiko
• sentence-final punctuation
• 003B ; is the preferred character
→ 003F ? question mark
≡ 003B ; semicolon

Additional letter

037F Ϳ GREEK CAPITAL LETTER YOT
• lowercase is 03F3 ϳ

Spacing accent marks

0384 ΄ GREEK TONOS
→ 00B4 ´ acute accent
→ 030D ◌̍ combining vertical line above
≈ 0020 SP 0301 ◌́

0385 ΅ GREEK DIALYTIKA TONOS
≡ 00A8 ¨ 0301 ◌́

Letter

0386 Ά GREEK CAPITAL LETTER ALPHA WITH TONOS
≡ 0391 Α 0301 ◌́

Punctuation

0387 · GREEK ANO TELEIA
• functions in Greek like a semicolon
• 00B7 · is the preferred character
≡ 00B7 · middle dot

Letters

0388 Έ GREEK CAPITAL LETTER EPSILON WITH TONOS
≡ 0395 Ε 0301 ◌́

0389 Ή GREEK CAPITAL LETTER ETA WITH TONOS
≡ 0397 Η 0301 ◌́

038A Ί GREEK CAPITAL LETTER IOTA WITH TONOS
≡ 0399 Ι 0301 ◌́

038B <reserved>

038C Ό GREEK CAPITAL LETTER OMICRON WITH TONOS
≡ 039F Ο 0301 ◌́

038D <reserved>

038E Ύ GREEK CAPITAL LETTER UPSILON WITH TONOS
≡ 03A5 Υ 0301 ◌́

038F Ώ GREEK CAPITAL LETTER OMEGA WITH TONOS
≡ 03A9 Ω 0301 ◌́

0390 ΐ GREEK SMALL LETTER IOTA WITH DIALYTIKA AND TONOS
≡ 03CA ϊ 0301 ◌́

0391 Α GREEK CAPITAL LETTER ALPHA

0392 Β GREEK CAPITAL LETTER BETA

0393 Γ GREEK CAPITAL LETTER GAMMA
= gamma function
→ 213E ℾ double-struck capital gamma

0394 Δ GREEK CAPITAL LETTER DELTA
→ 2206 ∆ increment

0395 Ε GREEK CAPITAL LETTER EPSILON

0396 Ζ GREEK CAPITAL LETTER ZETA

0397 Η GREEK CAPITAL LETTER ETA

0398 Θ GREEK CAPITAL LETTER THETA

0399 Ι GREEK CAPITAL LETTER IOTA
= iota adscript

039A Κ GREEK CAPITAL LETTER KAPPA

039B Λ GREEK CAPITAL LETTER LAMDA

039C Μ GREEK CAPITAL LETTER MU

039D Ν GREEK CAPITAL LETTER NU

039E Ξ GREEK CAPITAL LETTER XI

039F Ο GREEK CAPITAL LETTER OMICRON

03A0 Π GREEK CAPITAL LETTER PI
→ 213F ℿ double-struck capital pi
→ 220F ∏ n-ary product

03A1 Ρ GREEK CAPITAL LETTER RHO

03A2 <reserved>

03A3 Σ GREEK CAPITAL LETTER SIGMA
→ 01A9 Σ latin capital letter esh
→ 2211 ∑ n-ary summation

03A4 Τ GREEK CAPITAL LETTER TAU

03A5 Υ GREEK CAPITAL LETTER UPSILON

03A6 Φ GREEK CAPITAL LETTER PHI

03A7 Χ GREEK CAPITAL LETTER CHI

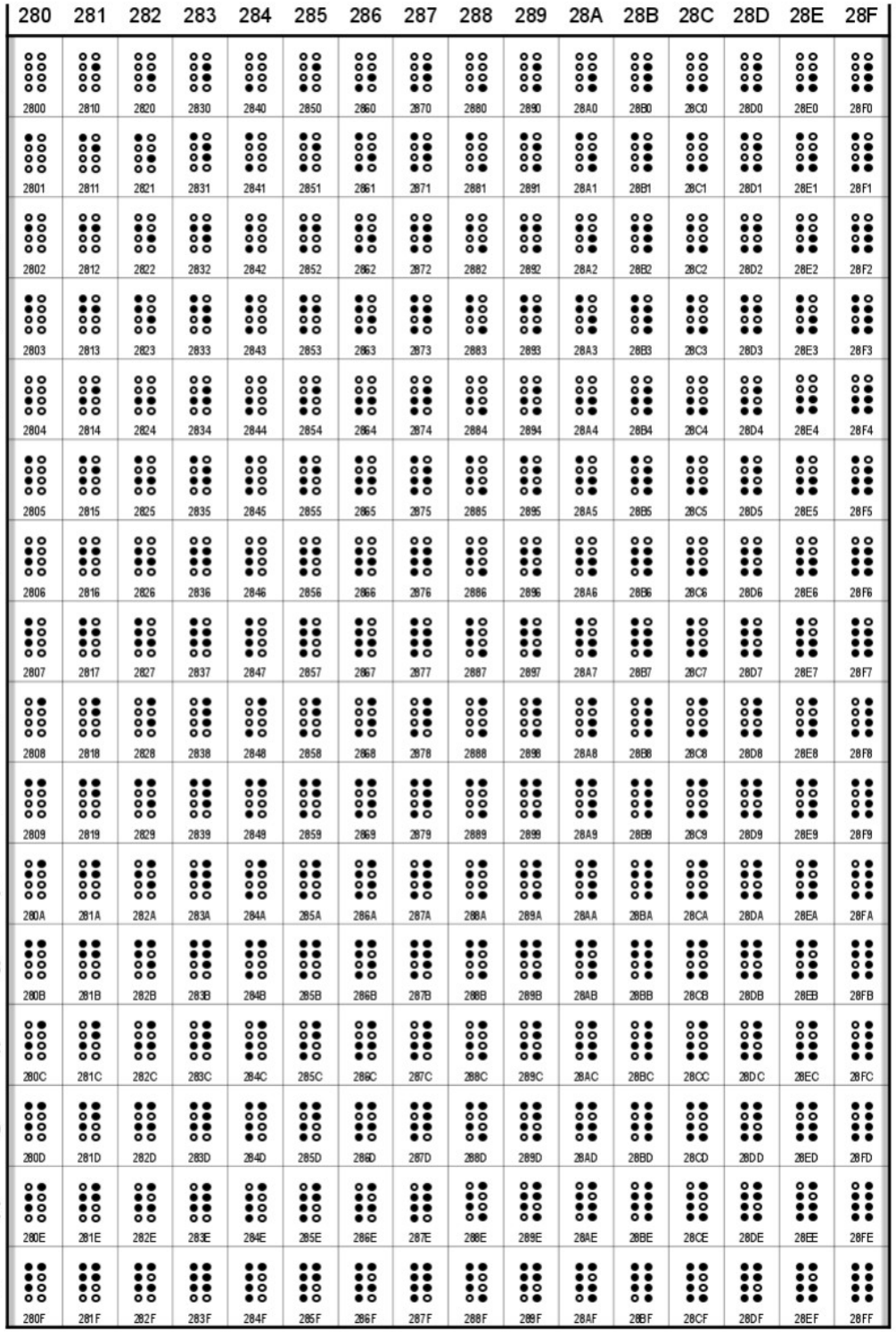
03A8 Ψ GREEK CAPITAL LETTER PSI

03A9 Ω GREEK CAPITAL LETTER OMEGA
→ 2126 Ω ohm sign
→ 2127 ℧ inverted ohm sign

03AA Ϊ GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
≡ 0399 Ι 0308 ◌̈

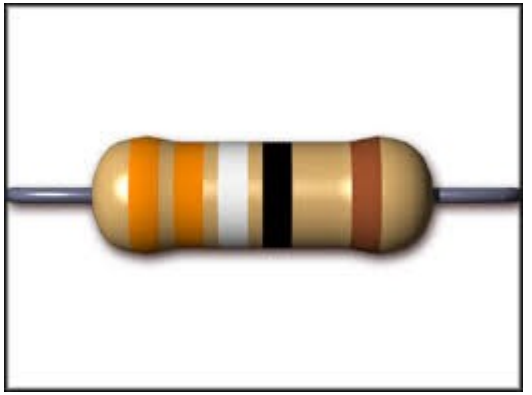
03AB Ϋ GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
≡ 03A5 Υ 0308 ◌̈

03AC ά GREEK SMALL LETTER ALPHA WITH TONOS
≡ 03B1 α 0301 ◌́


280 281 282 283 284 285 286 287 288 289 28A 28B 28C 28D 28E 28F
0 2800 2810 2820 2830 2840 2850 2860 2870 2880 2890 28A0 28B0 28C0 28D0 28E0 28F0
1 2801 2811 2821 2831 2841 2851 2861 2871 2881 2891 28A1 28B1 28C1 28D1 28E1 28F1
2 2802 2812 2822 2832 2842 2852 2862 2872 2882 2892 28A2 28B2 28C2 28D2 28E2 28F2
3 2803 2813 2823 2833 2843 2853 2863 2873 2883 2893 28A3 28B3 28C3 28D3 28E3 28F3
4 2804 2814 2824 2834 2844 2854 2864 2874 2884 2894 28A4 28B4 28C4 28D4 28E4 28F4
5 2805 2815 2825 2835 2845 2855 2865 2875 2885 2895 28A5 28B5 28C5 28D5 28E5 28F5
6 2806 2816 2826 2836 2846 2856 2866 2876 2886 2896 28A6 28B6 28C6 28D6 28E6 28F6
7 2807 2817 2827 2837 2847 2857 2867 2877 2887 2897 28A7 28B7 28C7 28D7 28E7 28F7
8 2808 2818 2828 2838 2848 2858 2868 2878 2888 2898 28A8 28B8 28C8 28D8 28E8 28F8
9 2809 2819 2829 2839 2849 2859 2869 2879 2889 2899 28A9 28B9 28C9 28D9 28E9 28F9
A 280A 281A 282A 283A 284A 285A 286A 287A 288A 289A 28AA 28BA 28CA 28DA 28EA 28FA
B 280B 281B 282B 283B 284B 285B 286B 287B 288B 289B 28AB 28BB 28CB 28DB 28EB 28FB
C 280C 281C 282C 283C 284C 285C 286C 287C 288C 289C 28AC 28BC 28CC 28DC 28EC 28FC
D 280D 281D 282D 283D 284D 285D 286D 287D 288D 289D 28AD 28BD 28CD 28DD 28ED 28FD
E 280E 281E 282E 283E 284E 285E 286E 287E 288E 289E 28AE 28BE 28CE 28DE 28EE 28FE
F 280F 281F 282F 283F 284F 285F 286F 287F 288F 289F 28AF 28BF 28CF 28DF 28EF 28FF


130E 130F 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 131A 131B
0 130E0 130F0 13100 13110 13120 13130 13140 13150 13160 13170 13180 13190 131A0 131B0
1 130E1 130F1 13101 13111 13121 13131 13141 13151 13161 13171 13181 13191 131A1 131B1
2 130E2 130F2 13102 13112 13122 13132 13142 13152 13162 13172 13182 13192 131A2 131B2
3 130E3 130F3 13103 13113 13123 13133 13143 13153 13163 13173 13183 13193 131A3 131B3
4 130E4 130F4 13104 13114 13124 13134 13144 13154 13164 13174 13184 13194 131A4 131B4
5 130E5 130F5 13105 13115 13125 13135 13145 13155 13165 13175 13185 13195 131A5 131B5
6 130E6 130F6 13106 13116 13126 13136 13146 13156 13166 13176 13186 13196 131A6 131B6
7 130E7 130F7 13107 13117 13127 13137 13147 13157 13167 13177 13187 13197 131A7 131B7
8 130E8 130F8 13108 13118 13128 13138 13148 13158 13168 13178 13188 13198 131A8 131B8
9 130E9 130F9 13109 13119 13129 13139 13149 13159 13169 13179 13189 13199 131A9 131B9
A 130EA 130FA 1310A 1311A 1312A 1313A 1314A 1315A 1316A 1317A 1318A 1319A 131AA 131BA
B 130EB 130FB 1310B 1311B 1312B 1313B 1314B 1315B 1316B 1317B 1318B 1319B 131AB 131BB
C 130EC 130FC 1310C 1311C 1312C 1313C 1314C 1315C 1316C 1317C 1318C 1319C 131AC 131BC
D 130ED 130FD 1310D 1311D 1312D 1313D 1314D 1315D 1316D 1317D 1318D 1319D 131AD 131BD
E 130EE 130FE 1310E 1311E 1312E 1313E 1314E 1315E 1316E 1317E 1318E 1319E 131AE 131BE
F 130EF 130FF 1310F 1311F 1312F 1313F 1314F 1315F 1316F 1317F 1318F 1319F 131AF 131BF

A **Unicode code point** represents a **character**



Characters are defined by their **meaning** in a **language**,
Glyphs are defined by their **appearance**.

A text-to-speech reader should pronounce “a 339 **Ω** resistor”
“a three hundred and thirty nine Ohm resistor” and not
“a three hundred and thirty nine uppercase omega resistor”

The glyph **Ω** is represented by unicode character
U+03A9 when it represents the Greek letter omega
U+2126 when it represents Ohms, the unit of electrical resistance.

The glyph **M** is represented by unicode character
U+004D when it represents a Latin letter
U+216F when it represents the Roman numeral for 1,000.

Glyphs are handled by **font renderers**

typeface vs. font

Back in the good old days of analog printing, every page was laboriously set out in frames with metal letters. That was rolled in ink, and then it was pressed down onto a clean piece of paper. That was a page layout. Printers needed thousands of physical metal blocks, each with the character it was meant to represent set out in **relief** (the **type face**). If you wanted to print Garamond, for example, you needed different blocks for every different size (10 point, 12 point, 14 point, and so on) and weight (bold, light, medium).

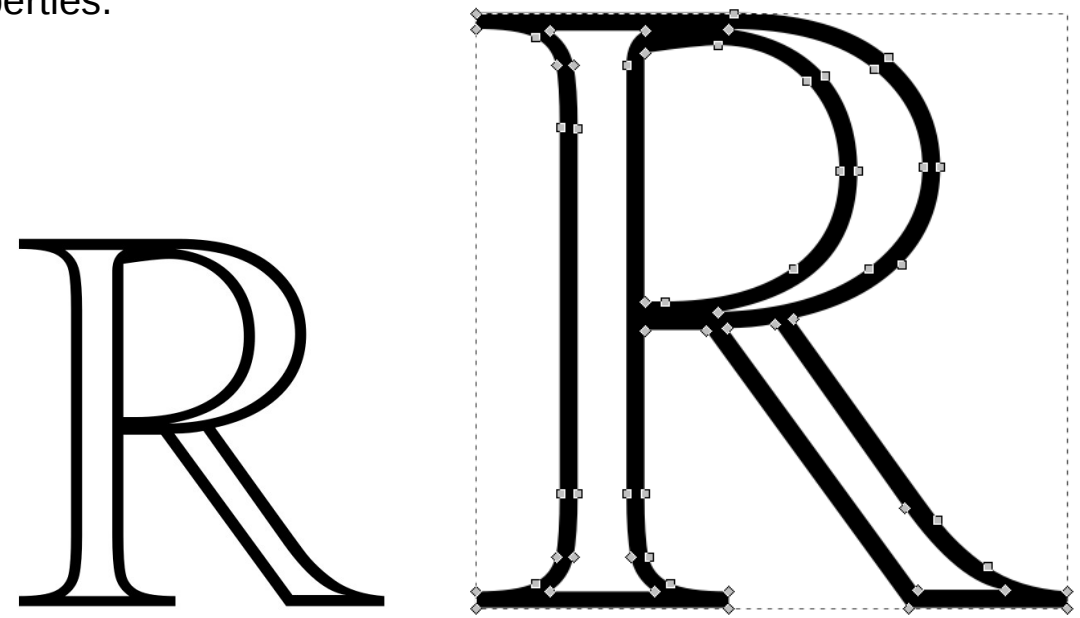
A **typeface** (also known as **font family**) is a set of one or more fonts each composed of **glyphs** that **share common design features**. Each font of a typeface has a specific weight, style, condensation, width, slant, italicization, ornamentation, and designer or foundry (and formerly size, in metal fonts).

A **font** described a **subset of blocks** in a **typeface**—but each font embodied a particular **size** and **weight**. For example, bolded Garamond in 12 point was considered a different font than normal Garamond in 8 point, and italicized Times New Roman at 24 point would be considered a different font than italicized Times New Roman at 28 point.

Scalable font vs. Bit-mapped font

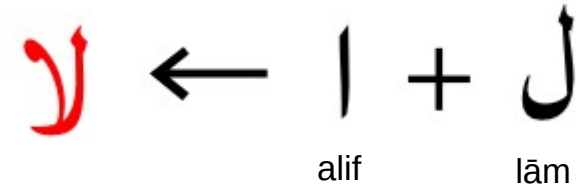
A **scalable** font is a font that is created in the required point size **when needed** for **display** or **printing**.
The **dot patterns (bitmaps)** are generated from a set of outline fonts, or base fonts,
which contain a mathematical representation of the typeface.
The two major scalable fonts are Adobe's Type 1 PostScript and Apple/Microsoft's TrueType.

A **bitmapped** font that is designed from scratch for a particular font size. It always looks the best.
Scalable fonts however eliminate storing hundreds of different sizes of fonts on disk.
In most cases, only the trained eye can tell the difference.
Scaling does not always retain all properties.



Character vs. Glyph ligatures

character combo	ligature	example
ff	ff	coffee
fi	fi	fiscal
ffi	ffi	office
fl	fl	flavor
Th	Th	The



A **ligature glyph** is the **joining** together of **one or more** glyphs into **one continuous** glyph.
The ligature for aesthetically combining fi is **one glyph**, but **two characters**.

A **ligature character** (unicode standard):
"The existing ligatures exist basically for compatibility and round-tripping with non-Unicode character sets. Their use is discouraged."

bloomingdale's

Unicode string **encodings**

A Unicode **string** is a **sequence of code points** (each representing a character).

This sequence needs to be **represented** as a set of **bytes** (unsigned integer values from 0 through 255) in memory. The rules for translating a Unicode string into a sequence of bytes are called an **encoding**.

Encodings don't have to handle every possible Unicode character, and most encodings don't.

ASCII encoding:

If a code point is < 128, each byte is the same as the value of the code point.
If a code point is >= 128, the Unicode string can not be represented in this encoding.

```
>>> ord('a'.encode('ASCII'))
97

>>> '€'.encode('ASCII')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
UnicodeEncodeError: 'ascii' codec can't encode character '\u20ac' in position 0: ordinal not in range(128)
```

Latin-1, also known as **ISO-8859-1** encoding**:**

Unicode code points 0–255 are identical to the Latin-1 values,
so converting to this encoding simply requires converting code points to byte values;
if a code point larger than 255 is encountered, the string can't be encoded into Latin-1.

```
>>> ord('a'.encode('Latin-1'))
97

>>> '€'.encode('Latin-1')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
UnicodeEncodeError: 'latin-1' codec can't encode character '\u20ac' in position 0: ordinal not in range(256)
```

Unicode string **encodings**

UTF-8 is one of the most commonly used encodings. UTF stands for "**Unicode Transformation Format**", and the '8' means that (one to four) 8-bit numbers are used in the encoding (i.e., a "**variable length** encoding").

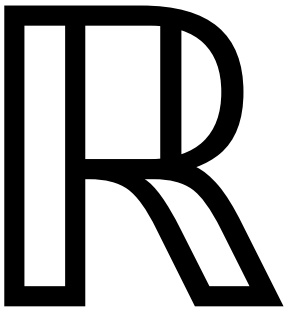
1st Byte	2nd Byte	3rd Byte	4th Byte	Number of Free Bits	Maximum Expressible Unicode Value
0xxxxxxx				7	007F hex (127)
110xxxxx	10xxxxxx			(5+6)=11	07FF hex (2047)
1110xxxx	10xxxxxx	10xxxxxx		(4+6+6)=16	FFFF hex (65535)
11110xxx	10xxxxxx	10xxxxxx	10xxxxxx	(3+6+6+6)=21	10FFFF hex (1,114,111)

UTF-8 has several convenient properties:

- It can handle **any** Unicode code point.
- A Unicode string is turned into a string of bytes containing **no embedded zero bytes**. Hence,UTF-8 strings can be processed by C functions such as strcpy() and sent through (e.g., network) protocols that can't handle zero bytes.
- A string of **ASCII text** is also valid UTF-8 text.
- UTF-8 is fairly **compact:** most commonly used characters can be represented with one or two bytes.
- If bytes are corrupted or lost, it's possible to determine the **start of the next** UTF-8-encoded code point and resynchronize. It's also unlikely that random 8-bit data will look like valid UTF-8.

Unicode string (en/de)coding

```
>>> ord('a'.encode('UTF-8'))
97

>>> '€'.encode('UTF-8')
b'\xe2\x82\xac'

>>> '€'.encode('UTF-16')
b'\xff\xfe\xac '

>>> '€'.encode('UTF-32')
b'\xff\xfe\x00\x00\xac \x00\x00'



>>> b'\xE2\x82\xAC'.decode('UTF-8')
'€'

>>> b'\xff\xfe\xac '.decode('UTF-16')
'€'

>>> b'\xff\xfe\x00\x00\xac \x00\x00'.decode('UTF-32')
'€'
```



Unicode Data	
Name	DOUBLE-STRUCK CAPITAL R
Block	Letterlike Symbols
Category	Letter, Uppercase [Lu]
Combine	0
BIDI	Left-to-Right [L]
Decomposition	<font> LATIN CAPITAL LETTER R (U+0052)
Mirror	N
Old name	DOUBLE-STRUCK R
Index entries	numbers, real R, DOUBLE-STRUCK CAPITAL real numbers set of real numbers, the
Comments	the set of real numbers
Version	Unicode 1.1.0 (June, 1993)

Encodings	
HTML Entity (decimal)	ℝ
HTML Entity (hex)	ℝ
How to type in Microsoft Windows	Alt +211D
UTF-8 (hex)	0xE2 0x84 0x9D (e2849d)
UTF-8 (binary)	11100010:10000100:10011101
UTF-16 (hex)	0x211D (211d)
UTF-16 (decimal)	8,477
UTF-32 (hex)	0x0000211D (211d)
UTF-32 (decimal)	8,477
C/C++/Java source code	"\u211D"
Python source code	u"\u211D"
More...	

In-browser UTF-8 test: http://www.fileformat.info/info/unicode/utf8test.htm
UTF-8 format description: http://www.fileformat.info/info/unicode/utf8.htm