

# 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. Possible values range from **0** through **0x10FFFF** (1,114,111 decimal). Only 143,859 are currently (Unicode 13.0) in use.

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

Unicode 7.0 Character Code Charts

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
```

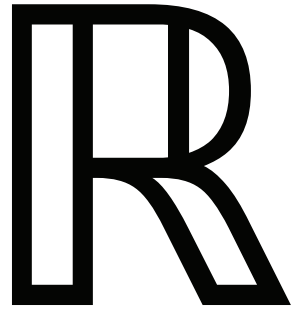
*Unicode* was designed to be an **ASCII-super set**: the first 256 characters in the *Unicode character set* are identical to those in the extended *ASCII* code.

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

Find chart by hex code:   Related links: [Name index](#) [Help & links](#)

## Scripts

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



U+211D

in Python(3):

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

℞

```
>>> print("\u211D")
```

℞

```
>>> ord("\u211D")
```

8477

```
>>> chr(8477)
```

'℞'

Unicode Data	
Name	DOUBLE-STRUCK CAPITAL R
Block	<a href="#">Letterlike Symbols</a>
Category	<a href="#">Letter, Uppercase [Lu]</a>
Combine	0
BIDI	Left-to-Right [L]
Decomposition	<font> <a href="#">LATIN CAPITAL LETTER R (U+0052)</a>
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	<a href="#">Unicode 1.1.0 (June, 1993)</a>

# 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
```

<https://decodeunicode.org/>

Renders all 137,374 Unicode 11.0.0 code points:

It produces the corresponding typographical character (more general: “glyph”)

	037	038	039	03A	03B	03C	03D	03E	03F
0	Ϝ 0370		ι 0390	Π 03A0	ύ 03B0	π 03C0	ϛ 03D0	ϝ 03E0	κ 03F0
1	ϝ 0371		Α 0391	Ρ 03A1	α 03B1	ρ 03C1	Ϟ 03D1	ϟ 03E1	ϙ 03F1
2	Τ 0372		Β 0392		β 03B2	ς 03C2	Υ 03D2	Ω 03E2	Ϙ 03F2
3	Τ 0373		Γ 0393	Σ 03A3	γ 03B3	σ 03C3	Υ 03D3	ω 03E3	ϙ 03F3
4	ϛ 0374	ϛ 0384	Δ 0394	Τ 03A4	δ 03B4	τ 03C4	Ϛ 03D4	ϣ 03E4	Θ 03F4
5	ϛ 0375	ϛ 0385	Ε 0395	Υ 03A5	ε 03B5	υ 03C5	ϕ 03D5	ϣ 03E5	€ 03F5
6	И 0376	Α 0386	Ζ 0396	Φ 03A6	ζ 03B6	φ 03C6	ϖ 03D6	ϛ 03E6	ϛ 03F6
7	и 0377	· 0387	Η 0397	Χ 03A7	η 03B7	χ 03C7	ϙ 03D7	ϛ 03E7	ϛ 03F7
8		Ε 0388	Θ 0398	Ψ 03A8	θ 03B8	ψ 03C8	ϙ 03D8	ϛ 03E8	ϛ 03F8
9		Η 0389	Ι 0399	Ω 03A9	ι 03B9	ω 03C9	ϙ 03D9	ϛ 03E9	ϛ 03F9
A	· 037A	Ι 038A	Κ 039A	Ϊ 03AA	κ 03BA	ϊ 03CA	ϙ 03DA	ϛ 03EA	Μ 03FA
B	ϙ 037B		Λ 039B	Ϛ 03AB	λ 03BB	ϛ 03CB	ϙ 03DB	ϛ 03EB	ϛ 03FB
C	ϙ 037C	Ο 038C	Μ 039C	ά 03AC	μ 03BC	ό 03CC	Ϝ 03DC	ϛ 03EC	ϙ 03FC
D	ϙ 037D		Ν 039D	έ 03AD	ν 03BD	ύ 03CD	Ϝ 03DD	ϛ 03ED	ϙ 03FD
E	· 037E	Υ 038E	Ξ 039E	ή 03AE	ξ 03BE	ώ 03CE	ϙ 03DE	ϛ 03EE	ϙ 03FE
F	ϙ 037F	Ω 038F	Ο 039F	ί 03AF	ο 03BF	ϙ 03CF	ϙ 03DF	ϛ 03EF	ϙ 03FF

#### Archaic letters

- 0370 Ϝ GREEK CAPITAL LETTER HETA  
→ 2C75 Ϝ latin capital letter half h
- 0371 ϝ GREEK SMALL LETTER HETA  
→ 2C76 ϝ latin small letter half h
- 0372 Ϟ GREEK CAPITAL LETTER ARCHAIC SAMPI
- 0373 ϟ GREEK SMALL LETTER ARCHAIC SAMPI

#### Numeral signs

- 0374 ϛ GREEK NUMERAL SIGN  
= dexia kerala  
• indicates numeric use of letters  
→ 02CA ϛ modifier letter acute accent  
≡ 02B9 ϛ modifier letter prime
- 0375 ϙ GREEK LOWER NUMERAL SIGN  
= aristeri kerala  
• indicates numeric use of letters  
→ 02CF ϙ modifier letter low acute accent

#### Archaic letters

- 0376 Ϟ GREEK CAPITAL LETTER PAMPHYLIAN DIGAMMA
- 0377 ϟ GREEK SMALL LETTER PAMPHYLIAN DIGAMMA

#### Iota subscript

- 037A ϛ GREEK YPOGEGRAMMENI  
= iota subscript  
→ 0345 ϛ combining greek ypogegrammeni  
≈ 0020 ϛ 0345 ϛ

#### Lowercase of editorial symbols

- 037B ϙ GREEK SMALL REVERSED LUNATE SIGMA SYMBOL
- 037C ϙ GREEK SMALL DOTTED LUNATE SIGMA SYMBOL
- 037D ϙ GREEK SMALL REVERSED DOTTED LUNATE SIGMA SYMBOL

#### Punctuation

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

#### Additional letter

- 037F ϙ GREEK CAPITAL LETTER YOT  
• lowercase is 03F3 ϙ

#### Spacing accent marks

- 0384 ϛ GREEK TONOS  
→ 00B4 ϛ acute accent  
→ 030D ϛ combining vertical line above  
≈ 0020 ϛ 0301 ϛ
- 0385 ϛ GREEK DIALYTIKA TONOS  
≡ 00A8 ϛ 0301 ϛ

#### Letter

- 0386 Α GREEK CAPITAL LETTER ALPHA WITH TONOS  
≡ 0391 Α 0301 ϛ

#### Punctuation

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

#### Letters

- 0388 Ε GREEK CAPITAL LETTER EPSILON WITH TONOS  
≡ 0395 Ε 0301 ϛ
- 0389 Η GREEK CAPITAL LETTER ETA WITH TONOS  
≡ 0397 Η 0301 ϛ
- 038A Ι GREEK CAPITAL LETTER IOTA WITH TONOS  
≡ 0399 Ι 0301 ϛ
- 038B ϙ <reserved>
- 038C Ο GREEK CAPITAL LETTER OMICRON WITH TONOS  
≡ 039F Ο 0301 ϛ
- 038D ϙ <reserved>
- 038E Υ GREEK CAPITAL LETTER UPSILON WITH TONOS  
≡ 03A5 Υ 0301 ϛ
- 038F Ω GREEK CAPITAL LETTER OMEGA WITH TONOS  
≡ 03A9 Ω 0301 ϛ
- 0390 ι GREEK SMALL LETTER IOTA WITH DIALYTIKA AND TONOS  
≡ 03CA ι 0301 ϛ
- 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 Δ 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 Π n-ary product
- 03A1 Ρ GREEK CAPITAL LETTER RHO
- 03A2 ϙ <reserved>
- 03A3 Σ GREEK CAPITAL LETTER SIGMA  
→ 01A9 Σ latin capital letter esh  
→ 2211 Σ 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 Ω inverted ohm sign
- 03AA Ι GREEK CAPITAL LETTER IOTA WITH DIALYTIKA  
≡ 0399 Ι 0308 ϛ
- 03AB Υ GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA  
≡ 03A5 Υ 0308 ϛ
- 03AC α GREEK SMALL LETTER ALPHA WITH TONOS  
≡ 03B1 α 0301 ϛ

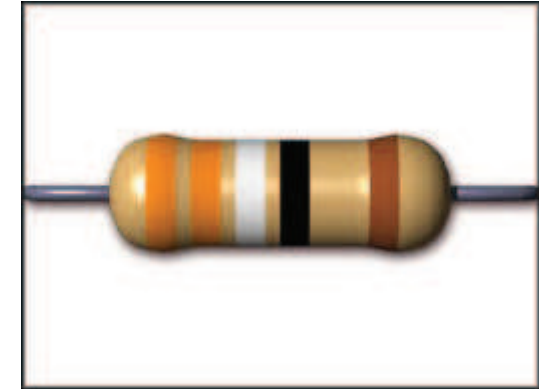
	280	281	282	283	284	285	286	287	288	289	28A	28B	28C	28D	28E	28F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

	130E	130F	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	131A	131B
0														
1														
2														
3														
4														
5														
6														
7														
8														
9														
A														
B														
C														
D														
E														
F														

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  $\Omega$  resistor”  
“a three hundred and thirty nine Ohm resistor” and not  
“a three hundred and thirty nine uppercase omega resistor”

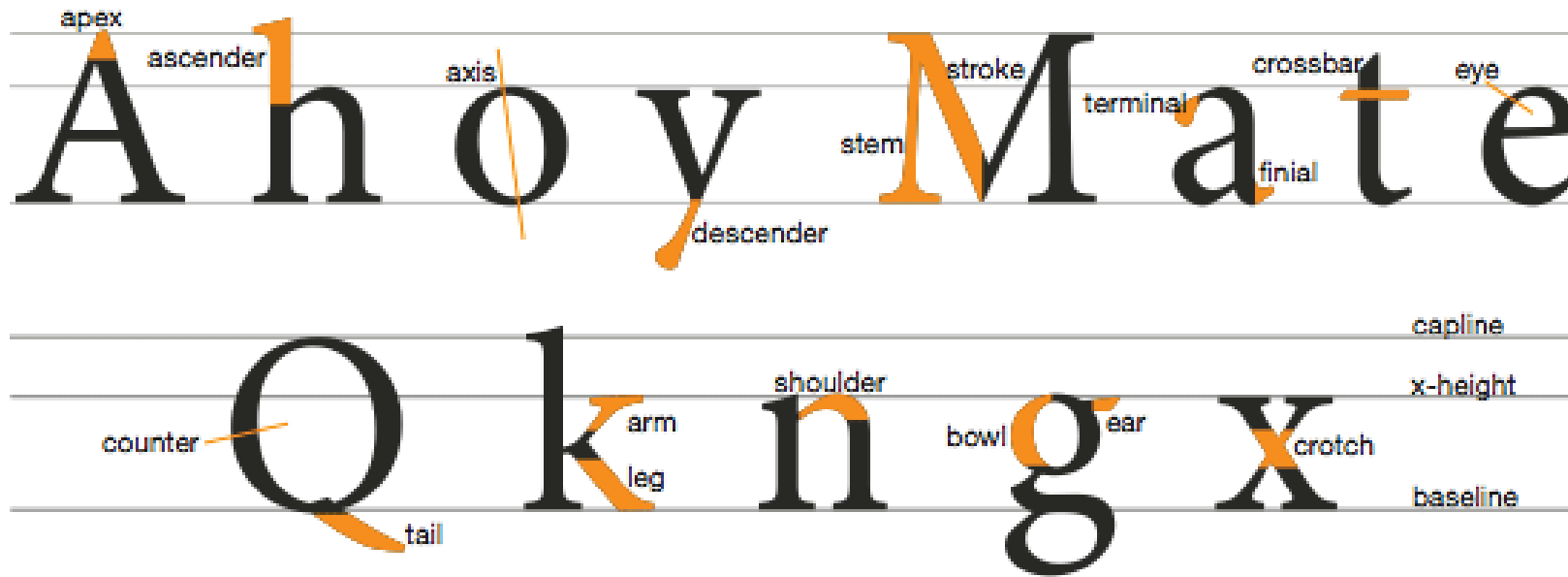


The glyph  $\Omega$  is represented by unicode character  
U+03A9 when it represents the Greek letter omega  
U+2126 when it represents Ohms, the unit of electrical resistance.

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**

# anatomy of a glyph



<http://designrfix.com/fonts/arial-helvetica>



Arial  
Geneva  
Helvetica  
Tahoma  
Verdana

# 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.



# computer representation: scalable vs. bitmap font

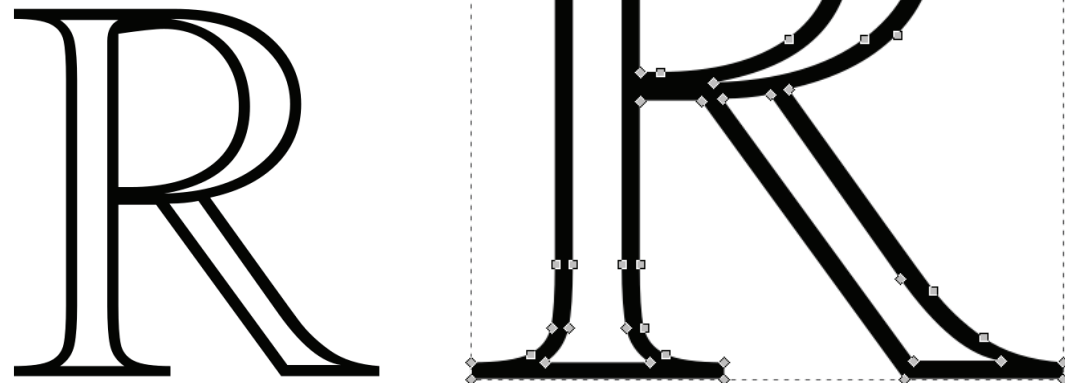


<http://www.computerhope.com>

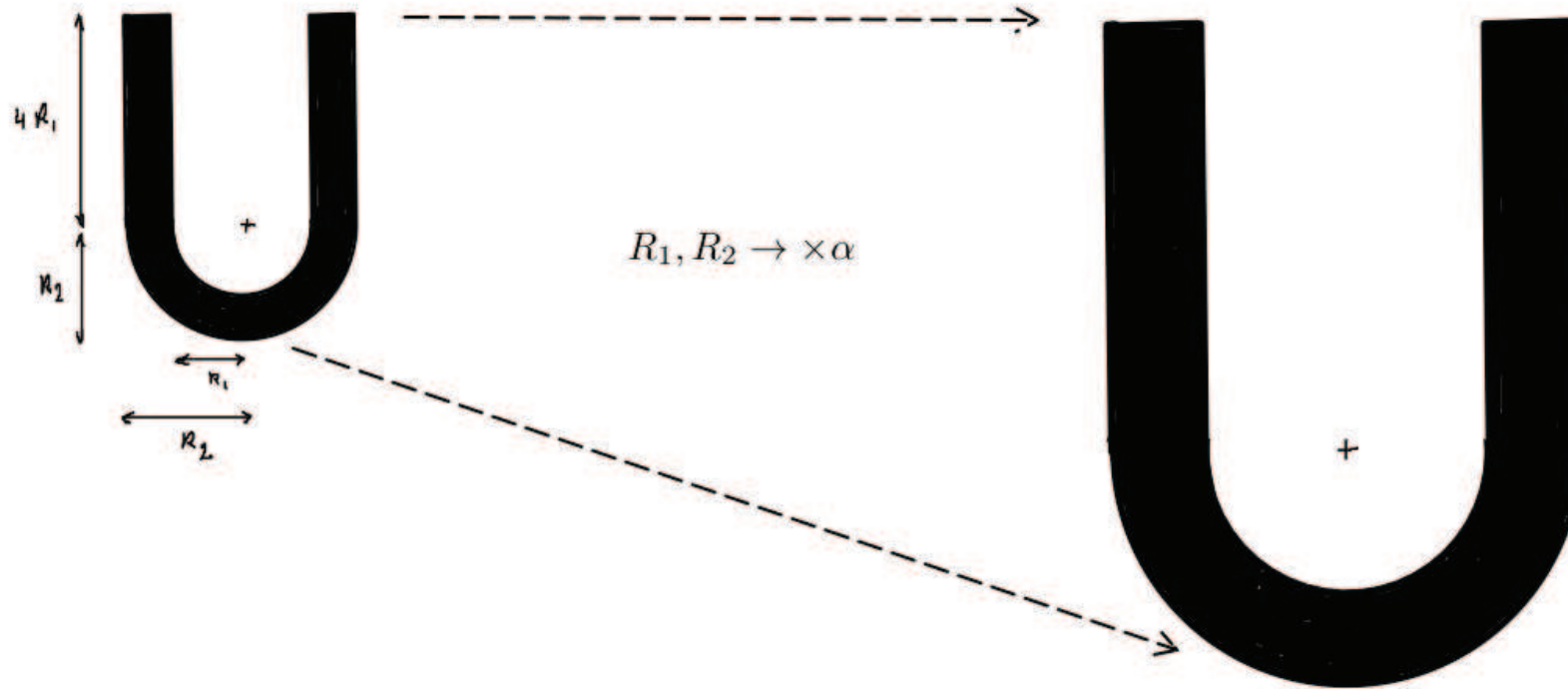
A **scalable** font is a font that is created in the required point size **when needed** for **display** or **printing**. The **dot patterns (bitmaps)** are generated from a set of **outline** fonts, or **base** fonts, which contain a **mathematical representation** of the typeface. The two major scalable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

A **bitmapped** font designed from scratch **for a particular font size** 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 visual properties.



# scalable font may not preserve all visual properties



preserved property under scaling: ratio of black/total area

$$s(R_1, R_2) = \frac{8R_1(R_2 - R_1) + \pi \frac{R_2^2 - R_1^2}{2}}{2R_2(4R_1 + R_2)} = s(\alpha R_1, \alpha R_2) = \frac{8R_1(R_2 - R_1) + \pi \frac{R_2^2 - R_1^2}{2}}{2R_2(4R_1 + R_2)}$$

non-preserved property under scaling: ratio of black area/sum of width and height

$$r(R_1, R_2) = \frac{8R_1(R_2 - R_1) + \pi \frac{R_2^2 - R_1^2}{2}}{4R_1 + 3R_2} \neq r(\alpha R_1, \alpha R_2) = \frac{\alpha(8R_1(R_2 - R_1) + \pi \frac{R_2^2 - R_1^2}{2})}{4R_1 + 3R_2}$$

# Proportional vs. monospace font

- displaying code with **indentation** (not using <TAB>)
- static vs. dynamic (animated) display of **changing** values

# character vs. glyph ligatures

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

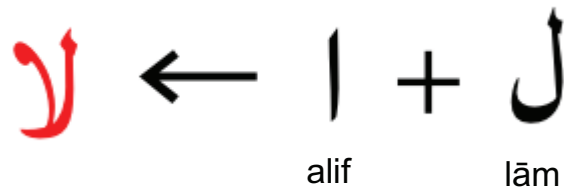
A **ligature glyph** is the **joining** together of **one or more** glyphs into **one continuous** glyph.

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."



Que di<sup>ct</sup>es vous de mon appel,  
Garnier? Fis je sens ou folie?

Toute be<sup>st</sup>e garde sa pel

Qui la contraint, e<sup>ff</sup>orce ou lie

S'elle peut, elle se deslie

Hoefler Text





# 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)

2<sup>21</sup>?

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                                     # < 127 → fits in 7 bit, compatible with ASCII

>>> 'a'.encode("UTF-8")
b'a' == b'\x61'                       # compatible with ASCII
                                     # b'...' means byte literal, not string

>>> '€'.encode('UTF-8')
b'\xe2\x82\xac'                       # variable length: 8, 16, 24, or 32 bit
                                     # write in binary and see patterns on previous slide
```

---

```
>>> '€'.encode('UTF-16-LE')
b'\xac '                               # variable length: 16 or 32 bit

>>> 'a'.encode("UTF-16-LE")
b'a\x00'                               # not compatible with ASCII, embedded \x00
```

---

```
>>> '€'.encode('UTF-32')
b'\xff\xfe\x00\x00\xac \x00\x00'     # 32 bit, not compatible with ASCII, embedded \x00
                                     # 64 bit? ... 4 byte Byte Order Mark (BOM)

>>> '€'.encode('UTF-32-LE')
b'\xac \x00\x00'                       # LE = Low Endian
                                     # 32 bit
```

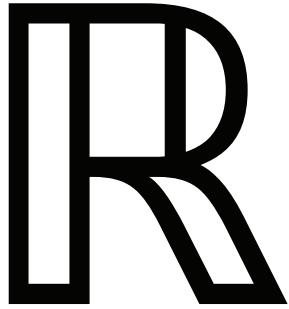
---

```
>>> 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	<a href="#">Letterlike Symbols</a>
Category	<a href="#">Letter, Uppercase [Lu]</a>
Combine	0
BIDI	Left-to-Right [L]
Decomposition	<font> <a href="#">LATIN CAPITAL LETTER R (U+0052)</a>
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	<a href="#">Unicode 1.1.0 (June, 1993)</a>

Encodings	
HTML Entity (decimal)	<a href="#">&amp;#8477;</a>
HTML Entity (hex)	<a href="#">&amp;#x211d;</a>
<a href="#">How to type in Microsoft Windows</a>	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	<a href="#">"�211D"</a>
Python source code	u"�211D"
<a href="#">More...</a>	

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>

# “ the big picture” (++ string of chars)

