15.4.14 Maximum unaccented height (*ASCENT*)

This measurement, on the em square, is from the baseline to the highest point reached by any glyph, excluding any accents or diacritical marks.

15.4.15 Maximum unaccented depth (*DESCENT*)

This measurement, on the em square, is from the baseline to the lowest point reached by any glyph, excluding any accents or diacritical marks.

15.4.18 Top Baseline

This gives the position in the em square of the top baseline. The top baseline is used by Sanscrit-derived scripts for alignment, just as the bottom baseline is used for Latin, Greek, and Cyrillic scripts.

Typeface anatomy

Typographers have developed a comprehensive vocabulary for describing the many aspects of typefaces and typography. Some vocabulary applies only to a subset of all scripts. *Serifs*, for example, are a purely decorative characteristic of typefaces used for European scripts, whereas the glyphs used in Arabic or East Asian scripts have characteristics (such as stroke width) that may be similar in some respects but cannot reasonably be called serifs and may not be purely decorative.

[edit] Serifs

Typefaces can be divided into two main categories: **serif** and **sans-serif**. Serifs comprise the small features at the end of strokes within letters. The printing industry refers to typeface without serifs as **sans-serif** (from French *sans*: "without"), or as *grotesque* (or, in German, *grotesk*).

Sans-Serif font

Serif font

Serif font (serifs highlighted in red)

Great variety exists among both serif and sans-serif typefaces. Both groups contain faces designed for setting large amounts of body text, and others intended primarily as decorative. The presence or absence of serifs forms only one of many factors to consider when choosing a typeface.

Typefaces with serifs are often considered easier to read in long passages than those without. Studies on the matter are ambiguous, suggesting that most of this effect is due to the greater familiarity of serif typefaces. As a general rule, printed works such as newspapers and books almost always use serif typefaces, at least for the text body. Web sites do not have to specify a font and can simply respect the browser settings of the user. But of those web sites that do specify a font, most use modern sans-serif fonts, because it is commonly believed that, in contrast to the case for printed material, sans-serif fonts are easier than serif fonts to read on the low-resolution computer screen.

[edit] Proportion

A **proportional** typeface displays glyphs using varying widths, while a **non-proportional** or **fixed-width** or **monospaced** typeface uses fixed glyph widths.

Most people generally find proportional typefaces nicer-looking and easier to read; and thus they appear more commonly in professionally published printed material. For the same reason, GUI computer applications (such as word processors and web browsers) typically use proportional fonts. However, many proportional fonts contain fixed-width figures so that columns of numbers stay aligned.

Monospaced typefaces function better for some purposes because their glyphs line up in neat, regular columns. Most manually-operated typewriters and text-only computer displays use monospaced fonts. Most computer programs which have a text-based interface (terminal emulators, for example) use only monospace fonts in their configuration. Most computer programmers prefer to use monospace fonts while editing source code.

ASCII art usually requires a monospace font for proper viewing. In a web page, the `<tt> </tt>` or `<pre> </pre>` HTML tag most commonly specifies non-proportional fonts. In LaTeX, the *verbatim* environment uses non-proportional fonts.

Any two lines of text with the same number of characters in each line in a monospace typeface should display as equal in width, while the same two lines in a proportional typeface may have radically different widths. This occurs because wide glyphs (like those for the letters W, Q, Z, M, D, O, H, and U) use more space, and narrow glyphs (like those for the letters i, t, l, and 1) use less space than the average-width glyph when using a proportional font.

In the publishing industry, editors read manuscripts in fixed-width fonts for ease of editing. The publishing industry considers it discourteous to submit a manuscript in a proportional font.

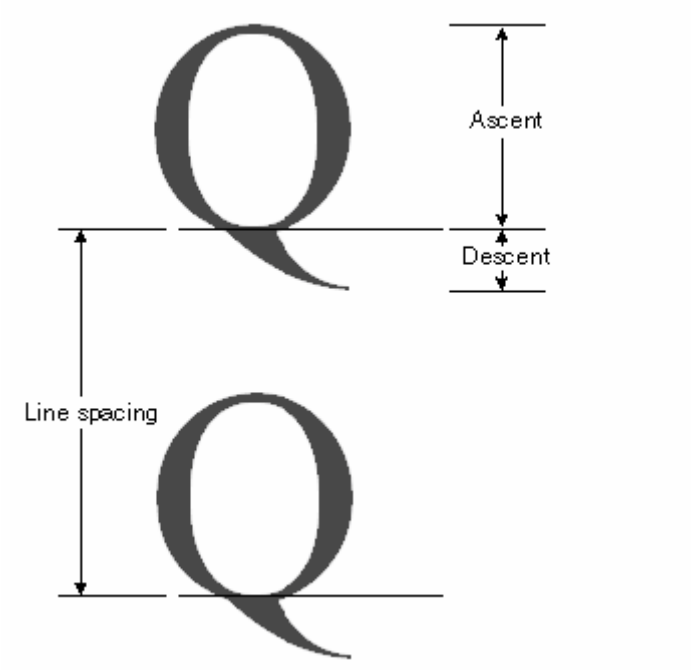
Font metrics

Most scripts share the notion of a baseline: an imaginary horizontal line on which characters rest. In some scripts, parts of glyphs lie below the baseline. The *descent* spans the distance between the baseline and the lowest descending glyph in a typeface, and the part of a glyph that descends below the baseline has the name "descender". Conversely, the *ascent* spans the distance between the baseline and the top of the glyph that reaches farthest from the baseline. The ascent and descent may or may not include distance added by accents or diacritical marks.

In the Latin, Greek and Cyrillic (sometimes collectively referred to as LGC) scripts, one can refer to the distance from the baseline to the top of regular lowercase glyphs (mean line) as the *x-height*, and the part of a glyph rising above the x-height as the "ascender". The distance from the baseline to the top of the ascent or a regular uppercase glyphs (cap line) is also known as the cap height.[1] The height of the ascender can have a dramatic effect on the readability and appearance of a font. The ratio between the x-height and the ascent or cap height often serves to characterise typefaces.

[MSDN]

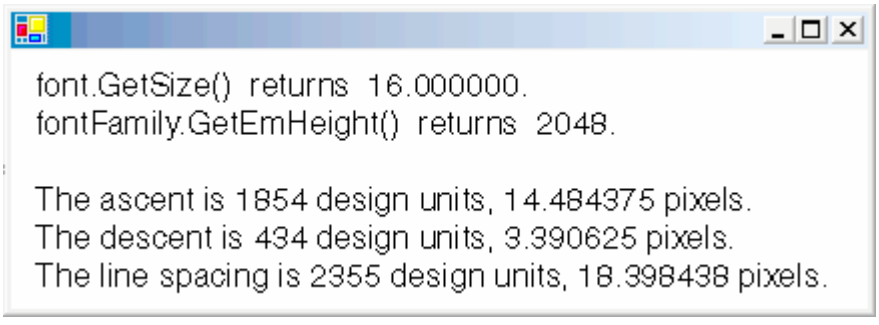
The following illustration shows the various metrics.



Example

The following example displays the metrics for the regular style of the Arial font family. The code also creates a **Font** object (based on the Arial family) with size 16 pixels and displays the metrics (in pixels) for that particular **Font** object.

The following illustration shows the output of the example code.



Note the first two lines of output in the preceding illustration. The **Font** object returns a size of 16, and the **FontFamily** object returns an em height of 2,048. These two numbers (16 and 2,048) are the key to converting between font design units and the units (in this case pixels) of the **Font** object.

For example, you can convert the ascent from design units to pixels as follows:

$$\frac{1854 \text{ design units}}{1} \times \frac{16 \text{ pixels}}{2048 \text{ design units}} = 14.484375 \text{ pixels}$$