JAMES FELICI
FOREWORD BY FRANK ROMANO

The Complete Manual *of* Typography

SECOND EDITION

A GUIDE TO SETTING PERFECT TYPE

"The ultimate typographic tool: a concise, beautiful book that pulls together everything you need to produce great typography."

FRANK ROMANO
ROCHESTER INSTITUTE OF TECHNOLOGY, SCHOOL OF PRINT MEDIA

"Dangerously good book on typography. 'Dangerous' because there is enough well-presented information in this volume to set you on the path to typography snobbery. This book is an excellent read and reference volume for any designer, print or web."

—NORA BROWN, Nora Brown Design

"Felici elegantly and painstakingly sets out to demonstrate how to set type 'perfectly' in a digital age. This is the book that answers all the questions you wanted to ask, but also demonstrates all the steps you need to pursue to achieve a kind of typographic perfection."

—MARGARET RICHARDSON, FontShop

"Buy this book, read it cover-to-cover, then keep it handy. You'll be surprised at what a difference it can make in the appearance of your work, both print and web."

—PETER BAUER, *Photoshop User*

"*The Complete Manual of Typography*, by James Felici, condenses timeless wisdom and timely technology into one complete guide. It explains everything about type designs and usage. If you had only one book on typography, this should be it."

—JAY NELSON, *Design Tools Monthly*

"Reading this book is like sitting down with a longtime typesetter and going over the details of a complex job. Most people will use it as a reference—which it is—but reading any section straight through is rewarding. The writing is clear and straightforward, and Felici has obviously thought long and hard about everything he deals with here."

—JOHN D. BERRY, CreativePro.com

"This excellent book discusses how type should look and how to set type like a professional."

—LINDA BUSHYAGER, *HiTech Review*

"What Felici's book does is show the importance to the reading experience of type that is well set on the page. It is copiously illustrated and elegant in design, and, I confess, I savored each of its 300 pages."

—DAN BARNETT, Musable Blog

"This is a superb reference book and should be often consulted by those who take pride in typography."

—PHILLIP PARR, Cider Press

"James Felici deserves a special place on every computer user's desk because with the power to put words on paper there comes a responsibility to do it well. For the ultimate guide to setting perfect type, you'll need *The Complete Manual of Typography*."

—FRED SHOWKER, DTG Magazine

"If nothing else, this book will make interesting reading for people who love to read books and think about the written word. For me, I wouldn't be without it, no matter the cost. This is one of my better reference books, and I Love Type."

—GEORGE ENGEL, Foxwood Estates Computer Club

"While Felici has abundant experience setting type in almost every format used in the twentieth century, he takes the capabilities and possibilities of the computer as a starting point for a very lucid and practical discussion of how to get the best possible type from software. The book contains one of the few really clear explanations of hyphenation and justification settings and how best to use them, as well as very practical and contemporary advice on issues such as line length and text color."

—FONTS ANON

"It covers all aspects of type design and applications of them in print and screen. This is like a master course in the finer points of typography. For a book that covers the historical tradition as well as digital innovations, this is a remarkable achievement."

—ROY JOHNSON, Mantex.co.uk

The Complete Manual *of* Typography, Second Edition

The Complete Manual *of*

SECOND EDITION

Typography

A GUIDE TO SETTING PERFECT TYPE

JAMES FELICI

The Complete Manual of Typography: A Guide to Setting Perfect Type, Second Edition
James Felici

This Adobe Press book is published by Peachpit.
For information on Adobe Press books, contact:

Peachpit
1249 Eighth Street
Berkeley, CA 94710
510/524-2178
510/524-2221 (fax)

For the latest on Adobe Press books, go to www.adobepress.com
Peachpit is a division of Pearson Education.
To report errors, please send a note to: errata@peachpit.com

Editor Rebecca Gulick
Production Editor and Compositor David Van Ness
Cover and Interior Designer Frances Baca with Mimi Heft
Copy Editor Karen Seriguchi
Proofreader Patricia Pane
Indexer Jack Lewis

This book is set in Monotype Perpetua and Linotype Syntax, both from Adobe Systems. Perpetua is a trademark of the Monotype Corporation registered in the U.S. Patent and Trademark Office and may be registered in certain other jurisdictions. Syntax is a registered trademark of Linotype-Hell AG and/or its subsidiaries.

ISBN-13: 978-0-321-77326-5
ISBN-10: 0-321-77326-8

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

for Jennifer

Foreword

Typography is what communication looks like.

But it is almost impossible to look and read at the same time because they are different perceptions. There is beauty in the language and beauty in the way it is presented. It all started about two millennia ago.

In AD 111, there was one typeface. It was the inscriptional seriffed lettering on the Trajan Column. In AD 2011, there are almost 200,000 fonts (most of them based on Garamond).

The Romans chiseled type into granite and made it monumental. Later, Jenson engraved type into metal and made it elemental. He went from columns of stone to columns of type.

There is a difference between type and typography. Typography was born because Gutenberg wanted to make his Bible appear handwritten. It was the first major publishing scam—pages reproduced as printed type at hand-calligraphy prices.

Typography is the use of type to advocate, communicate, celebrate, educate, elaborate, illuminate, and disseminate. Along the way, the words and pages become art. Type and typography fostered books, magazines, catalogues, newspapers, forms, and a plethora of promotional materials.

Type and typography—what you do and how you do it—are both science and art. There are rules, most of which are ignored. There are tools, most of which are unknown. But now you have the ultimate typographic tool: Jim Felici's knowledge at your fingertips.

As we edge toward 600 years of linear text and phase from paper to screen, the principles of good typography have not changed even as the technology of typography continues to change.

Type and typesetting went from metal to wood to film to dots. We set individual letters, lines of letters, and then pages of letters. We went from mechanical machine to mainframe to mini to desktop computer. We went from bitmaps to programmed curves and splines. We went from PostScript to TrueType to OpenType.

In only a few years we wiped out the entire typesetting industry, and typesetting became the province of the creative originator. The most demanding

type buyers became less demanding. We saw typewriter inch marks instead of real quotes and two hyphens substituting for em dashes. Forget about en dashes and real small caps and good H&J. Eventually, the industry did give us professional font sets, and programs automated many typographic processes.

There was a time when Courier, a monospaced typewriter face, was the most used typeface on the planet. Today that distinction belongs to a combination of Times and Helvetica. The most used faces are still the classics.

The letter and the numeral and the symbol begat the glyph, and the number of glyphs in a font multiplied—real small caps, old-style figures, gobs of diacriticals, and dingbats galore.

Those who work with type have to catch up with both what is old and what is new. Fortunately, you have the solution in your hands: a concise, beautiful book that puts together in one place everything you need to produce great typography. Thanks, Jim.

—FRANK ROMANO
Professor Emeritus
Rochester Institute of Technology
School of Print Media

Contents at a Glance

This page intentionally left blank

Table of Contents

Introduction

This book is about how type should look and how to make it look that way. It primarily covers type in print, which is where the art of typography reaches its highest form of expression. But people who read on computers, e-book readers, or any other electronic device need all the typographic help they can get, so setting type for screen display also gets its due.

The book is organized so that you can approach it in two ways: as a textbook to read from cover to cover, or as a reference guide to jump into at any point as need dictates. It has a wonderful index.

This is not a style guide, but an execution guide. It doesn't explain why you might choose to use the typeface Bembo over Garamond, but rather, having made that choice, how you can set Bembo in the best possible way.

The rules of typography are centuries old, and although the technologies have changed, the goal has always remained the same: a beautiful setting in the service of a pleasant and fruitful reading experience. So while this book explains in very practical terms how to use today's computerized tools, I've written it to outlast them. It's been completely updated since the first edition appeared in 2003, and as in the original, references to specific programs have been kept to a minimum (although the capabilities of all the major programs have been taken into account). Programs change, but the lessons in this book will be just as applicable to version 20.0 of your software as they are to the version you use today.

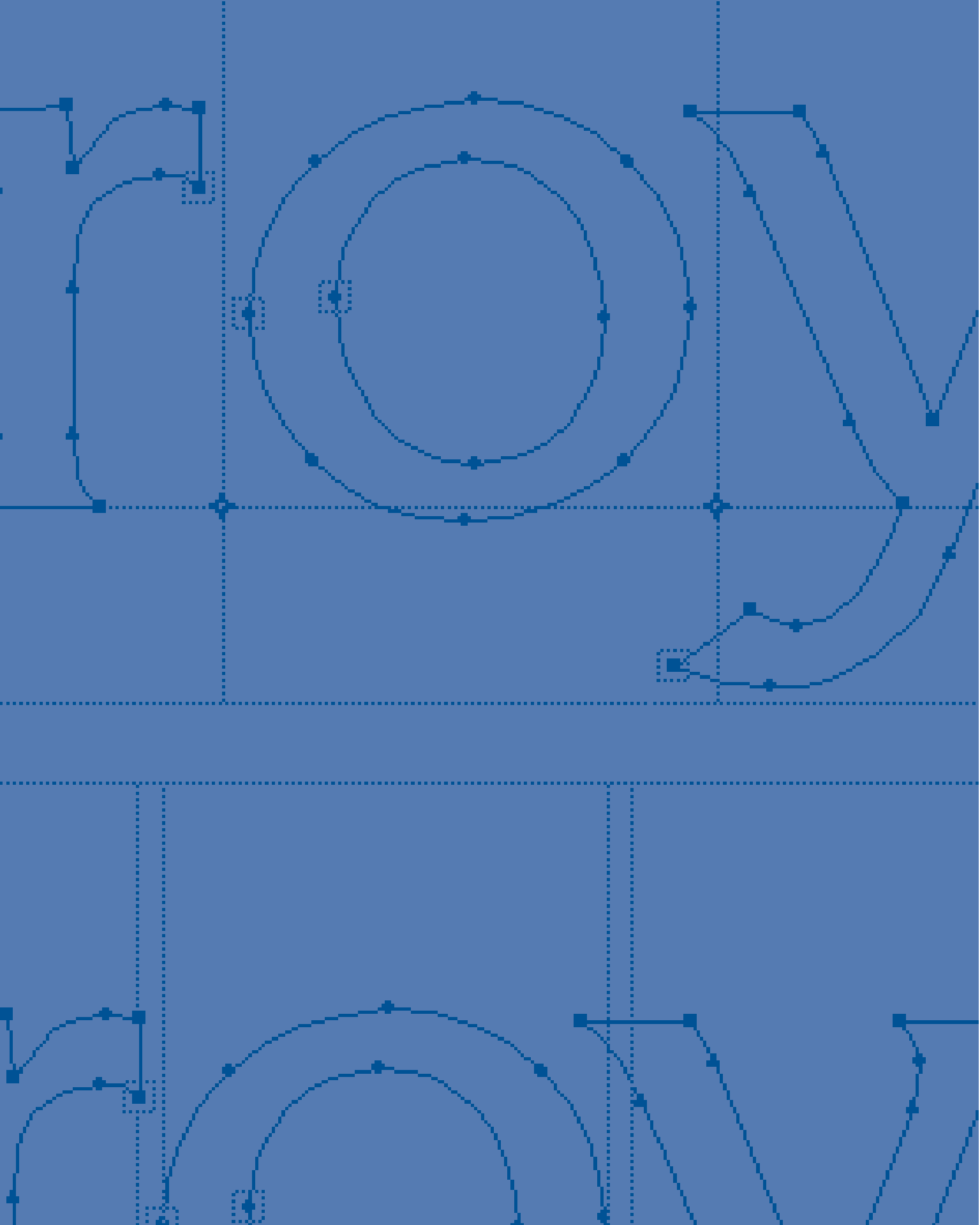
Not all of the capabilities discussed in this book exist in every program or system. But none of them are fantasies—they all exist somewhere. Every typographer and typesetter has to hope that they all will converge in one program as soon as possible. In the meantime, I've included scores of workarounds to wring good type out of uncooperative programs.

Beautiful type comes from attention to myriad tiny details. It's built up a fraction of an em at a time, through hundreds of decisions whose geometry belies their gravity. It requires, as a colleague once wrote, a heart hardened against accusations of being too fussy.

—JAMES FELICI

This page intentionally left blank

PART ONE Typographic Basics



CHAPTER 1 The State of the Art and How We Got Here

The way we set type today—and the language we use to talk about the process—comprise a hodgepodge of technologies, techniques, and influences that have accumulated for more than 500 years, ever since Johannes Gutenberg developed his printing system based on the notion of movable type. A surprising number of his innovations and refinements are still at work behind the curtains of today's slick computerized typesetting systems (as well as the most modest of word processors). You can't talk about type without using words that have roots in German, French, and Italian.

And despite typesetting's evolution over the years, most of its fundamental concepts can be understood best by looking at how Gutenberg figured out how to set type. This chapter explains the essentials of typesetting in the historical settings that spawned them.

The Building Blocks of Type

The basic idea of movable type is that every letter of the alphabet, every punctuation mark, and every numeral and symbol is molded in high relief on its own metal block. These blocks are set in rows to form lines of text, and the raised

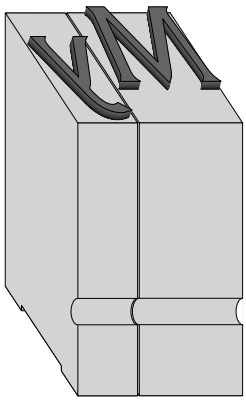


FIGURE 1.1 Gutenberg's system of movable type was based on the casting of each character on its own metal block. The widths of the blocks varied to accommodate each character. Word spaces were created with shorter blocks that didn't come up to the level at which the blocks were inked. The "nick" on the front of each block is an orientation aid, helping the hand typesetter to quickly distinguish, for example, between a *d* and an upside-down *p*.

shapes of the letters get inked and pressed onto paper. The common rubber stamp uses the same principle, but the type isn't movable.

Credit for inventing movable type probably belongs to the Koreans, who, centuries before Gutenberg, were printing with reusable type blocks made from ceramic. Gutenberg's genius was to work the concept into a complete typesetting and printing system (combining a press design borrowed from winemakers with metal-casting techniques borrowed from jewelers). The fundamental element was standardizing the dimensions and manufacture of those little printing blocks (Figure 1.1).

To set type using Gutenberg's technique, the blocks—cast from a lead alloy—were arranged in rows, as shown schematically in Figure 1.2. (To be printed correctly, the images of the characters were reversed and had to be set in lines reading from right to left—one aspect of the old system that has fortunately been left behind.) Those rows of letter blocks—lines of type—were stacked one below the other to form a page, and then the whole rectangular mass was locked in place by a frame that pressed in on all sides. To add space between the lines, thin strips of metal were inserted between the rows, a process called *adding lead.* Today the term *leading* is still used to describe the distance between one line of type and the one above or below it.

Just as important as the printing blocks in this scheme were the nonprinting blocks, whose job it was to secure the position of the others. These "blanks" were used to form indents and to fill out incomplete lines like those at the end of a paragraph.

Although this system of setting type was later automated and eventually computerized—replacing the metal printing blocks altogether—the concepts of letter blocks and spacing blocks persist in digital typesetting. They are, in fact, the keystones of the entire system.

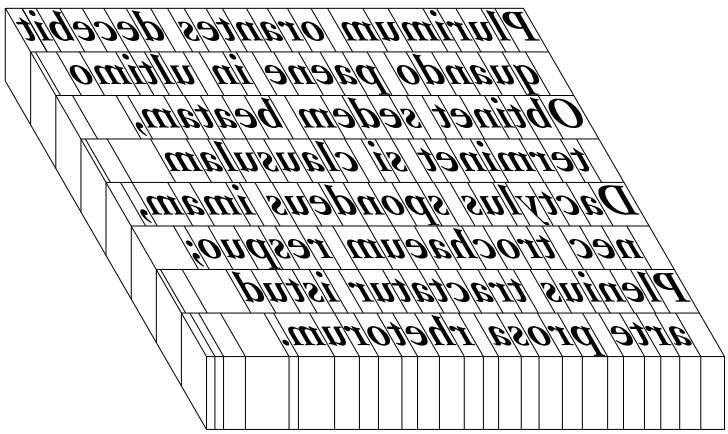


FIGURE 1.2 This schematic view of a body of handset metal type makes visible what's going on behind the scenes in a modern computerized typesetting system. Rows of characters are stacked as lines of text, and the ends of the lines (as well as the indents) are filled out with spaces. In handset type, the spaces are necessary for the type to be "locked up," held in place in a tight rectangle so that the block can be moved as a unit and printed without the individual characters moving around.

Bounding Boxes and Spaces

Within a computerized typesetting system, every character and symbol is conceived of as existing in a box whose dimensions are analogous to the surface of those old metal printing blocks (Figure 1.3). Just as in Gutenberg's day, this *bounding box* defines the space each letter takes up on a printed page. The space between the character image itself and the edge of the bounding box—called the *side bearing*—defines how far that character's image will be from the image of a character set next to it. Likewise, the upper and lower limits of the bounding box define where the lines of type above and below it will be set.

The big difference between metal type and digital type, of course, is that bounding boxes—being virtual boundaries, not physical ones like the edges of a piece of metal type—can now be manipulated so that they overlap (Figure 1.4). Typesetters in the digital age can position every character with infinite freedom and control. This is much easier than hacking at tiny metal blocks with a file to change their shape and alter their spacing relative to their neighbors.

Like handset type, digital type must also take into account spaces within and between lines, such as paragraph indents or the extra space between a headline and the text below it. Although those areas appear blank on the printed page, it's important to think of them as filled with space rather than as simple voids.

If there is an essential truism in typesetting, it is that a page contains no voids, only spaces between printed elements. The essence of typesetting is regulating the size of those spaces to control the balance and rhythm between black and white. This is the key to a graphically harmonious page—one with good *type color*—as well as to text that is pleasing and easy to read.

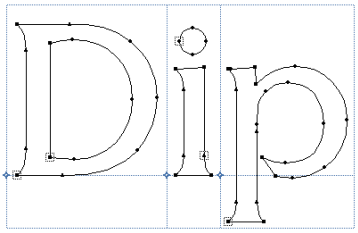


FIGURE 1.3 Seen from inside a font-editing program, character outlines appear in individual bounding boxes. The points along the outlines indicate where curves with differing shapes and straight line segments meet.

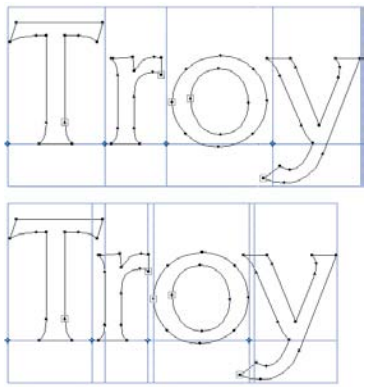


FIGURE 1.4 Drawn from within a font-editing program, the top illustration shows how characters fit together when they're typed without adjustments to the spacing. When the spaces between characters have been adjusted to compensate for the shapes they present to their neighbors (bottom), you can see how the bounding boxes of the characters overlap.

In the top sample, you can see that the *y* in this typeface is a *kerning character;* that is, parts of it extend beyond its bounding box. Kerning characters are created this way to minimize unsightly gaps between them and the letters next to them.

Type Design as a Function of Size

Gutenberg may have created a system for the reproduction of printed pages, but the visual system he was imitating—calligraphy—was already highly evolved. To meet the expectations of readers, Gutenberg and the printers that followed him were obliged to follow type design, book design, and calligraphic conventions very closely.

Calligraphers already knew, for example, that characters rendered in different sizes should be proportioned differently. Type in small sizes is more legible if the characters are somewhat wider and somewhat heavier. In large sizes, letters corresponding to the same design can be more finely modeled: thinner, lighter, and more nuanced (Figure 1.5). When various sizes are used together, such subtle design variations are not apparent and the design of the characters seems consistent throughout.

FIGURE 1.5 These three samples represent the same typeface, but they've been created using different master designs. The one at the top is based on a design for small type—footnotes, for example—and of the three it clearly has the heaviest bulk and greatest width, features that make it more legible despite its small stature on the page. The middle sample is drawn from a master designed for text-sized type. The bottom sample is from a display-sized master, and its narrower width and lighter weight are specifically tailored for large sizes.

Because metal type foundries needed to create separate sets of printing blocks for every type size that printers wanted to use, it was easy enough for them to follow this calligraphic tradition. The practice continued into the latter half of the twentieth century, when ironically, technological progress threatened this highly sophisticated system. First by means of photographic lenses and later through mathematical scaling, typesetting technologists discovered techniques by which a single type design could be shrunk or enlarged to a range of sizes. A single *master* image, then, could generate type for footnotes as well as newspaper headlines (Figure 1.6).

Vendors of typesetting systems were happy they didn't have to design type over and over for every size they wanted to produce, but it was a setback for the quality of printed material. When all sizes of type were produced from a single master (usually a size designed for standard book-text use), small type became pinched looking and hard to read, and large type looked bulky and sprawling. As a result, wholly separate typesetting systems came to be used for text-sized type and for larger *display* type. Technological solutions to these problems are only starting to appear (see "Multiple Master Fonts," in Chapter 3).

Evolution and Automation

The ink wasn't dry on Gutenberg's Bible before imitators came up with similar typesetting and printing systems. Type styles proliferated and popular styles were mercilessly plagiarized (another typographical tradition still honored today). There were no universal standards for the proportions or dimensions of the type blocks used to set type.

For almost 400 years, though, the processes of typesetting and printing very much resembled those practiced by Gutenberg. Type was set letter by letter, and after a print run, the type was dumped in a pile to be sorted and redistributed into its cases by a host of sorry apprentices. Those poor souls had to unfailingly separate *1*s from *l*s and *0*s from *O*s, differentiating them by their reversed images on tiny lead blocks whose faces were typically no larger than a match head. Each cabinet where the type was stored had multiple drawers, or cases; capital letters—*majuscules*—were put in the *upper case,* and the small, *minuscule,* letters were put in the *lower case.* Woe betide the poor worker who sorted more slowly than his boss set type, leaving his master angry and *out of sorts*.

The latter half of the nineteenth century saw three major developments that changed the shape of how type was set. Within a decade the Linotype and Monotype machines had sounded the death knell for large-scale commercially handset type.

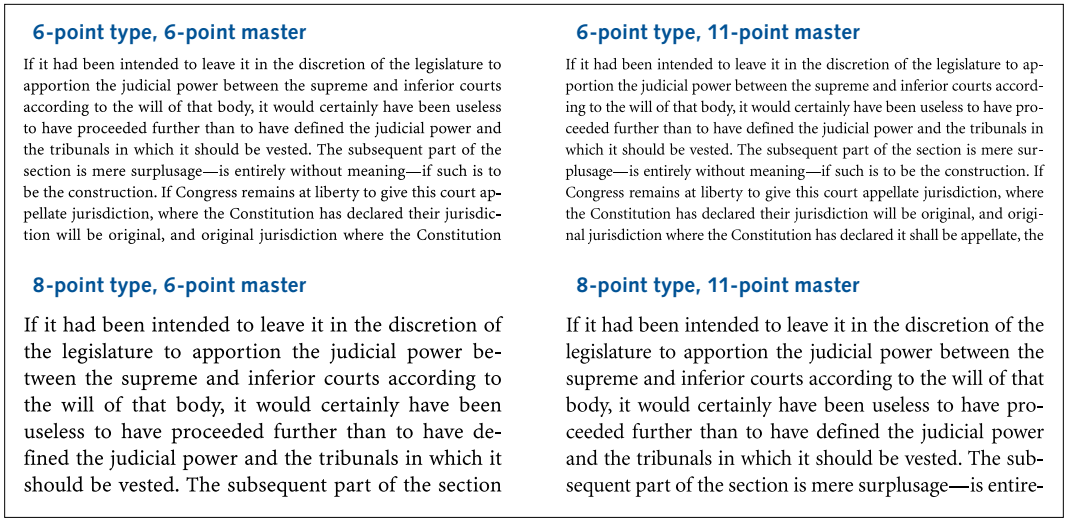


FIGURE 1.6 The left-hand samples above show type generated from a master design for small sizes. Compared with the samples on the right (based on a design for normal text-sized type), the small-master type is darker, it looks somewhat larger, and it sets wider, which you can see by comparing the line endings. The darker color of the small-master type will also blend better with that of other type on the page. Ultimately, type generated from the appropriate master designs will be easier to read than type—like that on the right—scaled out of its optimal size range.

The Typewriter: The First Desktop Publishing Tool

The first of those innovations was the Remington typewriter. It incorporated several concepts that are key to modern typesetting systems, and it paved the way for what could be called "office typesetting," which evolved into computerized word processing.

ESCAPEMENT

A primary feature of the typewriter is the movement of the printing element relative to the paper it's printing on. In most manual typewriters the carriage bearing the paper moves from side to side, and the printing element is stationary. In electronic typewriters, it's usually the printing head that moves.

The principle of such movement—called *escapement*—is applied in virtually every typesetting technology. After a manual typewriter key is struck, the paper moves—escapes—relative to the print mechanism. This puts the paper in position to receive the next character. When you type on a computer, the same thing occurs: After the character you type appears onscreen, the cursor moves to the right a distance equivalent to the width of that character.

Although it's no longer common practice in desktop typesetting programs, dedicated electronic typesetting systems routinely suspended the escapement motion to allow one character to be superimposed on another. Accented characters, for example, were commonly set by imaging one letter but suspending its normal escapement until another character—the accent—had been imaged on top of it. Then the escapement due the letter would be applied and normal typesetting would continue.

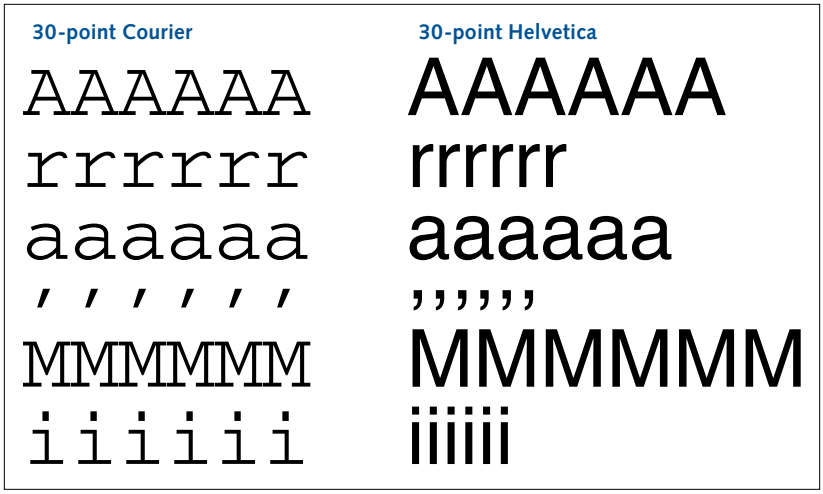


FIGURE 1.7 In monospaced typefaces such as Courier, all the characters take up the same amount of horizontal space on the line. Monospacing can be achieved through exaggerated side bearings (the white spaces flanking each character), as it is in the commas shown here, or by distortions in the character shapes themselves to make them fit the one-size-fits-all scheme. Proportionally spaced typefaces, such as Helvetica, allow characters to take on their "natural" proportions and widths.

MONOSPACED TYPE

The letters of the Latin alphabet (on which most European languages are based) have varying widths. The diverse shapes and widths of these letters, in fact, are a major reason why the practice of typography is so complex. These shapes have evolved over millennia to become part of the enormously subtle and sophisticated visual system of reading. They were not designed with typesetting in mind, and efforts to modernize the alphabet to make its character shapes more mechanically apt or more stylistically consistent have generally been resounding flops.

Typewriters, though, are crude devices (especially that first 1879 model), and it was impractical to design one that could have a unique escapement for every letter of the alphabet. So instead of adapting the machine to the alphabet, typewriter manufacturers adapted the alphabet to the typewriter, and *monospaced* type was born.

In a monospaced typeface, all the characters have the same width, so the typesetting machine (whatever it is) accords them all the same escapement. No matter which letter you type, the typewriter carriage moves the same distance. The common computer typeface Courier is in fact a typewriter typeface, and all its characters have the same width (Figure 1.7). In these monospaced types, where it was impractical to make the printed letterforms the same width, narrower ones—such as *l, i,* and punctuation—were given exaggerated features and side bearings so that the escapement was appropriate. Normally wider characters—*M, W, O*—had to be squeezed onto that same Procrustean bed.

Although monospaced fonts distort the natural forms of characters, such typefaces do have practical roles. Monospaced types continue to be used in applications where it's desirable to have the characters in each line align in neat vertical rows, as in some computer program displays. In addition, the

numbers in most typefaces have the same character width—you could call them a monospaced subset of the larger character set—so that numbers will align neatly in financial and numerical tables (Figure 1.8). When a computer typesetting system or printer can't image a particular typeface for some reason, it will often substitute the monospaced Courier, simply because Courier looks so different and so wrong that the error can't be overlooked. (In theory, anyway. If you start looking—particularly at ads—you'll often see Courier appearing in what are quite clearly erroneous but unnoticed substitutions.)

PROPORTIONAL TYPE

Monospacing was a dead end, but the idea of giving characters a limited number of specified numerical widths was appealing. Machine manufacturers still had to deal with the issue of escapement, but allowing every character to have a unique width would make for a very precise and complicated machine. Inventors of early typesetting systems tried dividing the characters in a typeface into a fixed number of categories, sorted by width. The narrowest ones would go in one group, the widest in another, and the rest in steps in between. If there were five such categories, the widths of the characters could be expressed as being from one to five units. In this scheme, all the characters would be designed to fit one of five widths, and the typesetting machine would translate those widths into one of five corresponding escapements. But to shoehorn all the characters of a typeface into such a small number of widths still meant distorting their design. To make better *proportionally spaced type*—that is, type that copied historical, handset models—the number of width categories had to be increased. This was what the Monotype and Linotype systems did.

Monotype: Counting Character Widths

The Monotype machine was an automated type foundry, which from a cauldron of hot metal cast individual letter blocks one at a time (at surprising speed) and spit them out into composed lines, just as if they had been set by hand. It could compose whole pages this way—starting from the bottom up—stacking one line upon another, eliminating huge amounts of handwork. When the job was done, the type was melted down and recycled—no more sorting.

A principal Monotype innovation was to allow someone working on a keyboard to record keystrokes and formatting commands—line length, indents, etc.—on a punched paper tape. The tape was then used to drive the typecasting machine, the way a punched paper scroll drives a player piano.

The information recorded on the paper tape was essentially the same as that recorded by today's computer programs as you type into a word processor

1,711,093,655
935,101,394
722,620
48,825,903

1,711,093,655
935,101,094
722,621
48,825,903

FIGURE 1.8 Most typefaces include *lining figures,* as shown at the top here. These numerals all have the same height and, most important, character width. They are, in effect, a monospaced subset of the characters within a typeface. This monospacing allows the numbers in tables to align neatly in columns. As in all monospaced systems, there are some spacing problems, and the *1* usually appears to be set a little too loose.

The sample at the bottom is set with old-style figures (sometimes called *lowercase figures*), which in this example have unique character widths, making them less desirable for financial reports and tables, where the lack of vertical alignment can create a disorganized impression.

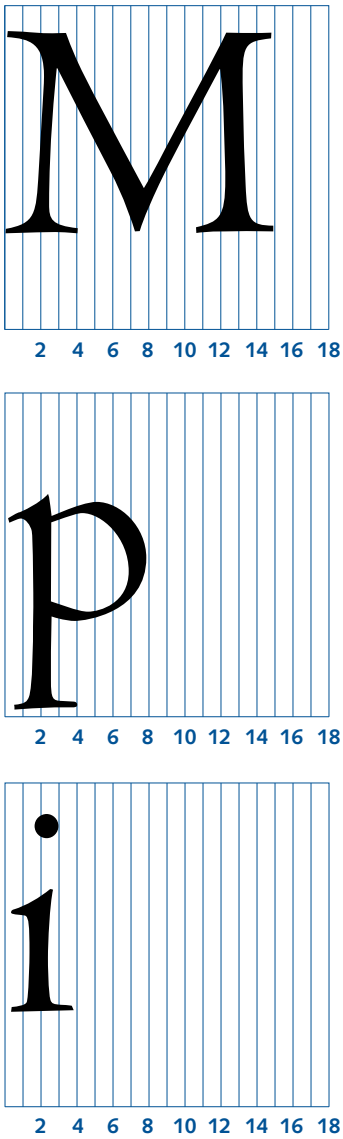


FIGURE 1.9 In the 18-unit character-width scheme used by Monotype casting systems, every character had to fit within one of the prescribed width categories. In this representation, the *M* on the top is 15 units wide, the *p* is 8 units wide, and the *i* only 4 units wide.

or page layout program. First, it recorded the width of every letter typed—the escapement, as it were—adding those widths together until their sum approached the specified length of the line being set. At the end of each line, the operator had to decide what to do with any small amount of space left over: leave it at the end of the line (as on a typewriter) or distribute it evenly between the words of the line, spreading the line to its maximum allowable length. (This process, called *justification,* is discussed in detail in Chapter 10.)

To accomplish this separation of input and output, every typeface for the Monotype system had to be designed with a standard way of expressing the widths of its characters. These widths were expressed in units, with the widest characters—often the *M* and *W*—being 18 units wide, and narrower characters correspondingly fewer, as seen in Figures 1.9 and 1.10.

The Monotype tape, like a modern computer disk, was recording two things as the typist worked: what keystrokes and spaces were called for and how wide they should be. The principal difference in this regard between now and then is that the widths of modern typefaces are measured not in one of 18 possible widths, but one of a thousand or more. Contemporary typeset characters can thus adhere even more closely to their natural, historical proportions. Typeface designers no longer have to worry about aesthetic restrictions imposed by the limitations of a typesetting system.

The Linotype machine, which appeared at about the same time, used a similar counting system but took a different tack. Instead of casting letters one by one, it assembled the molds for a line's worth of characters and cast them all at once in one piece—a line o' type. This *linecasting* machine was a "direct-input machine": Keystrokes were translated directly into machine action, not recorded in advance for output later.

The Changing Definition of *Font*

In the days of handset type, a *font* (a term that comes from an early French word meaning "molding" or "casting") comprised one or more drawers full of type blocks in a single size. (The difference between the current definitions of *font* and *typeface* are discussed in Chapter 3.) With the advent of the Monotype and Linotype machines, a font then became a set of molds (or *matrices*) from which type could be cast as it was needed, on the fly.

All of this type was destined for a specific kind of printing press, the *letterpress.* On a letterpress the printed impression is created by inking a raised surface (which can be line art or a halftone photograph as well as type) whose image is transferred under pressure to the paper. Recessed areas—those below *type-high*—receive no ink, do not come into contact with the paper, and so create the "blank" areas of the page.

FIGURE 1.10 This schematic layout shows how a Monotype font (or matrix) arranged characters according to their width, indicated on the scale at the left. Most of them are 9 or 10 units wide. Although the character set in this illustration is correct, the widths of the characters shown (drawn from a modern digital font) are only approximate.

But by the middle of the twentieth century, offset lithography was becoming an increasingly popular printing medium because of its radically lower costs. Offset litho is essentially a photographic process in which the image of a page is projected onto a thin, flexible printing plate covered with a photographic emulsion. When this plate is "developed" like film, the printing areas take on a quality that repels water but allows ink to adhere. When the plate is mounted on the rotating drums of a press, it is first wet with water and then smeared with ink. The oily ink is repelled by the wet parts of the plate (the blank parts of the page), but it adheres to the parts that form the printed image. The image is "offset"—printed, in effect—onto an intermediate roller, and that roller transfers the image in ink onto paper.

This double printing process may seem inefficient, but it has a very important consequence: It allows the image on the printing plate to be *right-reading,* just like the final page. On a letterpress, where ink is transferred directly to paper, the printing surface has to be a mirror image of the final page, just like a rubber stamp. That makes page composition much more complicated.

The development of offset lithography meant that printers no longer needed to stamp type physically to image it; all they needed was a photographic image of the type that could be transferred to the printing plate. Metal type was pushed out of the picture. Phototypesetting was born.

FIGURE 1.11 This is a life-size representation of a photographic film font used on a Mergenthaler VIP phototypesetter from the 1970s. Along with several other fonts, it was mounted on a drum that spun at high speed. A beam of light inside the drum was flashed through the appropriate character images as they whirled into position. The images were exposed onto photographic film.

PHOTOGRAPHIC FONTS

Early phototypesetting machines looked very much like Linotype and Monotype machines. But in place of the array of molds into which hot metal was poured for casting, these newer machines substituted small photographic negatives bearing the images of characters. A light shining through these negatives—one character after another—cast their images on photographic paper that, as on a typewriter, was advanced horizontally to build lines and then vertically to build pages or columns of type.

From a typesetting standpoint, the brilliant thing about photo type (quite apart from the absence of that cauldron of molten lead at your side all day) was that the type could be scaled to various sizes through a series of lenses. A range of type sizes could be generated from a single set of master images, a single font. Refinements in technology eventually reduced the size of these film fonts to a single negative of about 2 by 3 inches (Figure 1.11). But the type could still be enlarged only so far before it lost sharpness. Headline type, for example, had to be set on a machine equipped with much larger fonts.

ELECTRONIC FONTS

Efforts to improve the scaling of photographic fonts led in the 1950s to experiments with cathode-ray tubes (like those in TVs and computer monitors) to sharpen the images of type. Although that effort fizzled, by the 1960s a variety of typesetting machines appeared that could image type directly from a CRT onto photographic film. Images of the characters were not generated by photographs of letters; instead, mathematical formulas electronically generated the images on the screen. These were the first electronic fonts.

The most successful approach was to describe the outlines of characters, which were then filled in onscreen. But in those early days, most of the outlines were described as large numbers of straight lines and the simplest of curves.

Setting large type was still a problem, as parts of letters were marred by noticeable flat areas and facets. This scalable outline technology, though, has been refined, and it is now the standard on all typesetting systems (Figure 1.12).

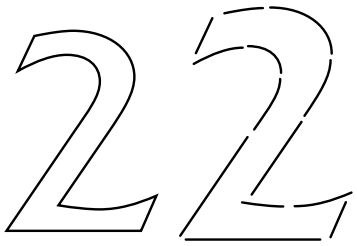


FIGURE 1.12 The outline of a character in a digital font (represented on the left) is composed of a relatively small number of straight-line and curved paths, or vectors, as shown in the disassembled version of the outline on the right. The entire outline can be scaled to any size while preserving its original proportions.

Desktop Publishing Alters the Rules

In the early 1980s a sudden series of technological changes revolutionized typesetting. First, desktop computers appeared that had enough memory and computing power to do the same work as those running dedicated typesetting systems, but at a fraction of the price. (A typical minicomputer for a dedicated typesetting system had only 768 kilobytes of memory.) The first of those dedicated systems was "ported" to a desktop computer in 1985.

Second, laser printers appeared that could function as low-cost desktop typesetters. Similar laser technology was soon applied to high-resolution phototypesetters, replacing the CRT-based generation of machines. At this point—when the laser began imaging—these machines stopped being mere typesetting machines and became *imagesetters,* as adept at rendering artwork as type.

Third, desktop computer operating systems inspired by concepts created at the Xerox Palo Alto Research Center (PARC) started using the screen to give an accurate preview of the printed page. This feature, called WYSIWYG (what you see is what you get), meant among other things that someone could set type without having to learn a huge vocabulary of arcane formatting commands. If you could make it look pretty on the screen, it would look pretty on the page.

Fourth, and perhaps most important, PostScript arrived, a computer language designed to describe any printed "event" on a page. Developed by Adobe Systems, PostScript was one of many *page description languages* (PDLs), but it was the most complete and the most promising, and—significantly—it had won commercial contracts with Apple Computer and Mergenthaler-Linotype, a leading manufacturer of high-end typesetting systems.

The PostScript Model

Probably the most notable aspect of PostScript—quite apart from the technology behind the language and all the things it could do—was that it was not linked to any specific computer or printer; nor was it linked to a particular operating system or other software. It was totally *device independent.* In theory, a page described in PostScript code (which could be written in universally readable ASCII—text-only—format) could be created on any computer and imaged on any printer, any monitor screen, or any unforeseeable other imaging device. Any computer program could express itself in PostScript code.

FIGURE 1.13 In a raster device such as an imagesetter or computer monitor, characters are drawn by a beam sweeping across the "page," blinking on and off as it goes. In this illustration, the beam has already swept horizontally across the page many times, drawing a slice of the character during each successive pass.

PostScript described everything it could as a scalable, *vector-based* object. That is, it defined objects as outlines or paths—vectors—that users could resize at will and image at any resolution. Prior to PostScript, most computer graphics—as well as type—were *bitmaps,* images drawn dot for dot (each one represented by one *bit* of computer data) for a particular resolution. Under the new regime, type became just another kind of scalable graphic.

RASTER IMAGE PROCESSING

To convert a page description written in PostScript into a printed page, those PostScript commands had to be interpreted and expressed for a specific output device. Because the computing process was so complex, the interpreter typically took the form of a "black box," called a *raster image processor,* or RIP. This was a separate computer dedicated solely to translating PostScript commands into directions that told an output device—a laser printer, for example—how to image a page.

The device's name is based on old television-industry jargon. It refers to the way an image is created on a CRT screen, one horizontal line at a time, by a narrow beam that scans from side to side in so-called raster lines (*raster* comes from a Latin word meaning "rake"), as shown in Figure 1.13. These horizontal lines are drawn from top to bottom on the screen at very high speed.

The beam—a stream of electrons in a CRT, a laser beam in a laser printer or imagesetter—blinks on and off to create the dark and light spots you see onscreen or on a printed page. The fineness of the beam and the rate at which it can blink on and off as it travels determine the *resolution* of the device, that is, the number of apparent dots it can create per inch or centimeter. Inkjet printers accomplish the same thing using an interruptible stream of ink instead of a beam of light.

The number of dots whose position a RIP has to calculate on each page is formidable. A page from a desktop laser printer with a resolution of 600 dots per inch (*dpi*) contains nearly 4 million dots. Even at the low end of common imagesetter resolutions, that number soars to 16 million dots per page.

Imaging on both the screen and the pages printed on desktop printers is usually handled by a software-only RIP built into a desktop computer. Dedicated high-resolution RIPs are still the norm for imagesetters.

DEVICE INDEPENDENCE

Device independence of the kind PostScript offered was revolutionary. In the centuries after Gutenberg, any type you bought had to match the press equipment you were using. Until the late nineteenth century, there weren't even any commonly accepted standards for the basic units of typographic measurement. When the rise of photocomposition led to an explosion in the

number of companies selling typesetting systems, each system vendor had its own proprietary font technology. Once you bought a system, you were obliged to buy that vendor's fonts, and they cost a fortune, often nearly as much as the hardware they worked with.

In addition, once a job was typeset on one system, it was wedded to that system forever, unless someone was willing to retype and reformat the entire thing. With very few exceptions, jobs created on one brand or model of typesetting system could not be transferred to another. It was no wonder that many publishers rushed to adopt a desktop publishing system—even with its more limited capabilities—just to be out from under the yoke of a single vendor and the cost of being locked in to a single proprietary technology. PostScript's promise of device-independent fonts was an irresistible siren call.

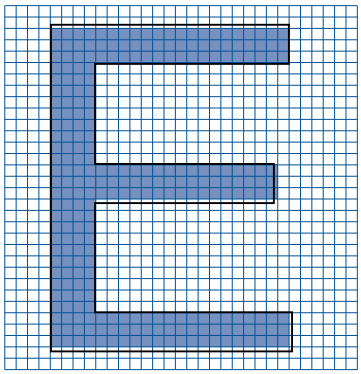


FIGURE 1.14 In this simplified illustration, a raster image processor lays the outline of a character over a grid representing all the possible pixels on a page and "colors in" those pixels whose centers fall within the outline. At this point, the imaged character is said to have been *rasterized*.

POSTSCRIPT FONTS

To make its device independence work, PostScript had to create a complete floating world of its own, free of proprietary technologies that kept existing systems incompatible with one another. Fonts were a key part of the equation; indeed, they proved to be one of the most complicated parts of the puzzle. Although its founders hadn't foreseen this development, Adobe soon became one of the leading font companies in the world.

PostScript fonts, like those in other formats that followed, stored the images of characters as outline drawings. The outlines are in turn built out of straight-line and curved segments. The curves—called *Bézier curves,* after the French automotive mathematician who discovered this compact way of describing complex curves—are the same as those used in popular drawing programs. When type of a certain size needs to be created, outlines for the characters are copied from the font, scaled to size by the RIP, and "colored in" (Figure 1.14) by the output device—usually a computer monitor, printer, or imagesetter—according to the resolution of the device.

IMAGING POSTSCRIPT FONTS

Determining which dots, or *pixels,* become part of this coloring-in is complicated. The RIP looks at the whole page as a grid of pixels. The process by which character outlines are superimposed on that grid, and decisions made about which pixels are to be imaged, is called *grid fitting.*

At its simplest, grid fitting consists of laying the outline of a character over the grid at its appointed location (the one specified by the typesetting commands that describe the page) and coloring in those pixels whose centers fall on or within the outline. As shown in Figure 1.15, though, this doesn't always create perfect visual results, especially at lower resolutions (as on a computer monitor screen),

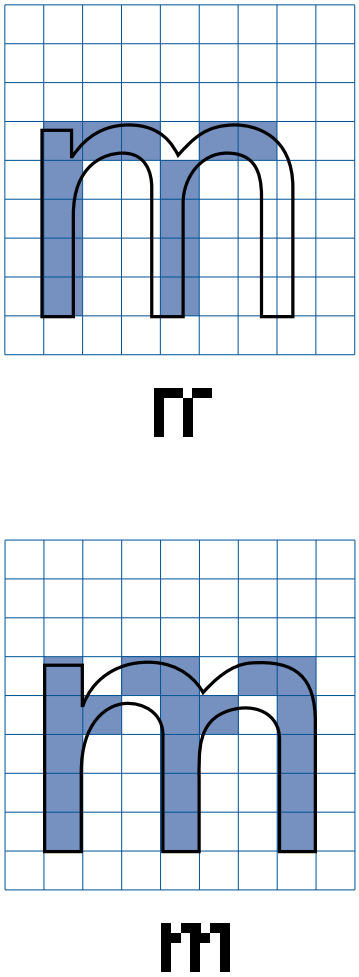


FIGURE 1.15 Character outlines rarely align as neatly on the grid as they do in Figure 1.14. Particularly at low resolutions (like the computer-screen resolution represented here), the outlines often fall at inconvenient locations. In the top grid, so few pixel centers fall within the outline that the character (shown as it would be imaged) is only half formed. For this reason, font engineers add instructions, or *hints,* to font coding that cause the outlines to change shape, assuring that the proper pixels are turned on, as shown in the lower grid.

where the eye can make out individual pixels and where even a single misplaced one can be distracting. In some cases, pixels may drop out, causing gaps in the bitmapped image. Sometimes subtly adjusting the position of the outline on the grid will create a more felicitous alignment of character form and pixels.

More often, the outline itself has to be reshaped to assure that the appropriate pixels are turned on and the character shape is reproduced with maximum fidelity. To accomplish this, programming instructions popularly called *hints* are added to fonts. These improve the clarity and form of characters rendered with relatively small numbers of pixels, in situations where they're being set at small sizes or at modest resolutions.

From the outset, Adobe Systems had its own system of hinting for PostScript fonts, but it would not give that system away for free, as it did with other details of how to create PostScript fonts. Other font vendors could create PostScript-compatible fonts, but those fonts wouldn't look as good as Adobe's except at very high resolutions. Some elements of the computing and publishing communities felt abused by this arrangement, and eventually Apple and Microsoft collaborated to create a new outline font format, TrueType.

PostScript, TrueType, and the later fusion font format combining the two—OpenType—are discussed and compared in detail in Chapter 4.

Output Resolution and Type Quality

Images of contemporary digital type are composed of dots. On a computer monitor screen, with a typical resolution of little more than 100 dpi, individual dots are big enough to see clearly. In this environment, it's very difficult to identify a particular typeface onscreen at common text sizes. There just aren't enough pixels available to draw a sufficiently detailed picture (Figure 1.16). To see type onscreen as clearly as it appears on the cheapest inkjet printer means zooming in to at least 300 percent magnification, at which point you don't get a good view of the page anymore.

As resolution increases, pixels become smaller, and the details and nuances of character shapes can be more clearly and accurately rendered. Equally important, at higher resolutions the crucial spaces between characters can be more rigorously controlled, resulting in far more legible and readable type. It's only above 1,000 dpi that you enter the realm of what's often called *typeset quality.*

But how the pixels are imaged is as important as their size. Photographic film is still the preferred substrate for commercial typesetting because the pixels imaged on it are exactingly formed and crisply defined. The grains of the silver compounds that change color when the film is exposed are minute, so the images of the pixels are sharply focused and hard-edged. Systems that create page images directly on offset printing plates use a variety of related