

A look at complaints about *The T_EXbook*

Udo Wermuth

Abstract

This article looks at the first volume of Donald E. Knuth’s *Computers & Typesetting* book series, titled *The T_EXbook*. It does not review this book in the classical sense, but rather lists aspects of this book’s style and contents that people dislike and criticize. Next, the paper discusses them one by one.

1 Introduction

Most, if not all, experienced T_EX users familiar with books teaching T_EX acknowledge that *The T_EXbook* [18] by Donald E. Knuth is a comprehensive exposition of all parameters and commands in T_EX and in its accompanying *format plain*. And all will agree that someone who wants to master the darker corners of T_EX should own it. Or as Alan Hoenig [15, p. 550] states: “*One of life’s main reference works, it should be beside any author’s computer.*”

In contrast to this outstanding status, critical statements exist too. Andrew Donald Booth wrote in a review about a book by Leslie Lamport [5]: “*One of the enjoyable things about T_EX is Knuth’s book which, although disorganized, is full of wit and, given the effort, is complete and accurate.*” Boris Veytsman recommends [37]: “*If you want to study T_EX (as opposed to L^AT_EX, ConT_EXt or other formats), then a book to read is T_EX by Topic by Victor Eijkhout. . . . After you read and understand Victor, return to The T_EXbook and reread it.*” He seems to indicate that *The T_EXbook* is quite difficult. (*T_EX by Topic* is not a book for beginners; see section 11.) But the comments are too unspecific to be discussed seriously. In section 2 I briefly look at the word “disorganized” and then I present a list of concrete complaints against *The T_EXbook*.

I use *The T_EXbook* a lot in my studies; I know it well and I like it. To be honest, I do not support any of the discussed complaints.

Contents. Section 2 states general remarks about *The T_EXbook* and other books about T_EX. Sections 3–9 discuss critical comments about *The T_EXbook* that are listed near the end of section 2. Section 10 addresses the misconception that it isn’t a book for beginners, but mentions issues too. Section 11 looks at other books. Section 12 contains final remarks.

2 Books about T_EX

For the first version of T_EX, Knuth also wrote a manual ([16] or the second part of [17]); it already had

The T_EXbook’s chapter titles but only ii+200 pages [38]. He didn’t spend as much time on it as he later needed for *The T_EXbook*; see below. However, users praised it as part of the first T_EX system [10, p. 22]: “*It is our conviction that the whole system, together with Knuth’s manual, is an eminently usable tool in the hands of any mathematician, even one who would sit at a computer terminal for the first time.*”

On the other hand, Richard Furuta [11, p. 1118] finds that Knuth’s goal to reach a high-quality output makes [16] unusually complex. He writes about [17] in 1983: “*The second part of the book describes T_EX in detail. There is an extraordinary amount of information here: not only is T_EX described, but one also finds a primer on style for mathematical typography and an introduction to typesetting terminology. Indeed, one’s initial reaction to T_EX may be: ‘Is formatting really this much work?’*”

The T_EXbook. The 1982 version of T_EX was accompanied by a new manual, which Knuth has continued to update through the current version of T_EX: *The T_EXbook* [18]. It has had many printings and special editions, for example, a *Millennium edition* in 2000 and most current is the *Jubilee edition* of 2021. Sometimes beginners ask how current a certain printing is. *The T_EXbook* is updated once in a while [30] but Knuth lists most changes in errata files; see CTAN’s package `knuth-errata`. So you can update your printing by hand. (Thanks to the book designer, the outer margin of the book is wide enough for notes.) However, T_EX has been frozen for decades; Knuth calls his programs *fixed points* [22]. Thus, T_EX and its manual get only bug fixes.

Digression: Knuth lists not only the errors in *The T_EXbook* but also the bugs in the program T_EX; see the file `errorlog.tex` in the above package. He does this as he wants to add data to the research about errors in software. It’s one of the rare sources that gives details about *all* errors found: during unit testing and later test phases. He cited other researchers in [23, p. 244] that made the request “*more data must be collected on different projects*” and he wanted to contribute.

The T_EXbook played a crucial role in the development of T_EX. Donald Knuth writes in [23, p. 255]: “*I debugged T_EX82 in the summer of ’82, then began to write the new manual — called The T_EXbook [18] — in October. The first manual had been written hastily and finished in 21 days, but I wanted The T_EXbook to meet much higher standards. Therefore I wasn’t able to finish it until a full year later. [¶] It was during this period, October ’82 to October ’83, that T_EX became a mature system. I had*

A look at complaints about *The T_EXbook*

to rethink every aspect of its design as I rewrote the manual.” This connection between program and manual is for Knuth also one of the reasons for T_EX’s success; he writes [20, p. 530]: “T_EX would have been much less successful if I had not put considerable effort into writing a user manual for it myself. The process of explaining the language gave me views of the system that I never would have perceived if I had merely designed it, implemented it, and used it.”

Knuth published *The T_EXbook* and made its source file, `texbook.tex`, as well as all private commands, so-called *macros* (see file `manmac.tex`), that he wrote for this book project available to be studied by experienced users. I surmise most T_EX experts have looked at these files in detail at least once.

A beginner’s manual. Knuth explicitly addresses beginners with *The T_EXbook* but he does it in a special way. He writes [18, p. v]: “This manual is intended for people who have never used T_EX before, as well as for experienced T_EX hackers. In other words, it’s supposed to be a panacea that satisfies everybody, at the risk of satisfying nobody.” For example, he introduces a dangerous-bend sign that marks material for readers with an advanced skill level. On page vi he explains his intention: “The reason for such different levels of complexity is that people change as they grow accustomed to any powerful tool. . . . At every stage in the development you’ll want a slightly different sort of manual. You may even want to write one yourself.”

A beginner who reads only the unmarked paragraphs finishes after the equivalent of ≈ 110 pages (not counting Appendices A and I but with B, E, F, H, and J), or $< 25\%$ of the complete book.

The T_EXbook not only describes T_EX but also teaches details about typography to all readers. This seems to be necessary as Knuth wants T_EX users to achieve high-quality typeset output [18, p. 1].

Other beginner’s manuals. Malcolm Clark prepared a manual that only addresses beginners. He writes about *The T_EXbook* [7, p. 378]: “The source reference for T_EX-as-we-know-him/her. It delights many, and frustrates others. Another class of reader is both delighted and frustrated.” (A. Hoenig on Clark’s book [15, p. 547]: “An engagingly written guide to plain T_EX, which is refreshingly complete.”) Clark’s book has almost exactly as many pages as *The T_EXbook* but does not handle all the details, and does not cover all of the `plain` format.

Arthur L. Samuel was probably the first author of a pure beginner’s T_EX manual [32]. It was a report of 34 pages from the Stanford University; the T_EX Users Group sold numerous copies of it [21,

p. 498]. Although limiting itself to “first grade T_EX”, Samuel explains quite complex constructions, for example, tables. He sometimes uses the output of the report in examples and shows what he entered to create it. Moreover, he describes his input in detail.

Other authors follow an approach similar to the one used by *The T_EXbook*. Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry [2, p. vii] state: “We also intend it to serve both newcomers to T_EX and those who are already familiar with T_EX.” They provide paths with “tutorial and reference information” [2, p. iv] to address the needs of these two groups. (Clark [7, p. 379] considers it a reference manual: “A good reference manual for those who already have a few inklings.”)

Beginner’s manuals have been published in languages other than English. I myself started with a 71-page DIN-A5 report by Norbert Schwarz from the Ruhr University Bochum [33]. (Later he extended this report and published it as a book [34].) A couple of weeks later I bought the spiral-bound edition of *The T_EXbook* and used it to learn T_EX. And I not only learned T_EX better, *The T_EXbook* gave me a basic introduction to typography.

What makes any book about T_EX difficult?

T_EX implements a lot of *primitive* commands and parameters; the format `plain` adds more parameters and many macros. T_EX knows in total ≈ 900 *control sequences* ([18, p. 9]), i.e., words and symbols preceded by T_EX’s *escape character*; T_EX’s default is the symbol ‘\’. David Bausum writes [4, p. x]: “For example, section A2.4 of *The C Programming Language* lists 32 reserved words, and Kernighan and Ritchie spend 150 leisurely pages explaining how those words work. In contrast, T_EX has 325 primitives. Plain T_EX adds hundreds of additional control sequences. Donald E. Knuth covers all of them in *The T_EXbook*’s 475 [sic, it’s x+483] pages (of which 175 [179] are appendices).”

One can try to reduce the complexity by listing commands and give brief descriptions as in the aforementioned report by A. L. Samuel. This gives you a set of commands for your documents, sufficient to write memos, letters, etc. But T_EX’s goal is to provide the highest quality, so that exceptional-looking books become possible. That’s why so many details are discussed in *The T_EXbook*, and that’s why the basics of typography are taught in this book. A reader gets a new view of these details and learns to understand why an author of a book should listen to a professional book designer and/or typographer.

Complaints about *The T_EXbook*. Earlier it was mentioned that *The T_EXbook* was written for users

of different skill levels. Knuth distinguishes at least three levels and guides the reader of *The T_EXbook* by paragraphs starting with none, one, or two dangerous-bend signs.

William R. Cheswick writes [6, p.707]: “The T_EXbook [Knuth] covers everything, of course. But it is actually three books in one: a beginner’s manual and two levels of reference manual guarded by dangerous-bend signs. Some dangerous curves are vital, others extremely arcane and usually irrelevant. Which ones should be read, and which ignored, and by what level of user?” Section 3 explains this. Later he adds: “There had to be a command in The T_EXbook somewhere, but how could I find it? I was the T_EXnician [see section 7], and T_EXHaX [a mailing list to ask for help] wasn’t available. It turns out that I wanted the `\string` command. It was hiding behind a double-dangerous curve . . .” Section 9 discusses *The T_EXbook*’s index and how to find information. (I was successful with each of three words that I directly link to T_EX’s concepts for `\string`.)

Andrew Booth’s remark about *The T_EXbook* being “disorganized” from section 1 might also refer to the dangerous-bend signs; as stated above, I handle this in the next section. Otherwise, I discuss in section 8 *The T_EXbook*’s structure, in case his complaint is about this aspect. He also makes the following remark [5] about an instructor’s workload that I do not discuss in this article. “Unlike the Knuth ‘T_EXbook,’ this L^AT_EX book does not have examples to instruct and test the novice user. This may or may not be a disadvantage. Knuth’s book includes complete solutions to every example, which leaves the instructor who wants to use it as a textbook with the task of inventing new ones.” The way *The T_EXbook* is written implies that it is suitable for self-study, if the reader accepts writing texts in English while studying the book. See also section 10.

One often finds a mixture of praises and complaints about *The T_EXbook*. But most statements are either very general or a truism. For example, Michael Doob writes [8, p.534]: “And then there is *The T_EXbook*. The amount of information in that book continues to amaze me; its organization is wonderful for the experienced user. But no book can be all things to all people. The very enthusiastic can learn all they need to know from *The T_EXbook*. For the less enthusiastic or less skilled, it is a rather unpleasant experience to try to do so. It is not a good starting point for most T_EX users (as, I’m sure, it was never intended to be).” The last sentence does not seem to follow directly from what was written before. Nevertheless, in section 10 I collect arguments for why *The T_EXbook* is a good book for be-

ginners. (Note: Unlike Booth, Doob finds *The T_EXbook* well-organized, at least for some readers.)

There is a *TUGboat* article by Angela Barden that looks at several T_EX manuals [3]. She articulates her criticism with concrete references to *The T_EXbook*. Here is a list of complaints about this book that I extracted from her article. From what I read elsewhere this list addresses often named concerns from other people too:

1. the paragraphs with dangerous-bend signs;
2. the manual’s “deliberate lying” about some of T_EX’s concepts when they are introduced;
3. the jokes that cannot be appreciated properly if the technical point is not understood;
4. the anatomy of T_EX and the imagery used: eyes, mouth, stomach, bowels, etc.;
5. the pronunciation and spelling of T_EX’s name.

Knuth did all these items intentionally; one might dislike them but they do not happen by accident.

3 Dangerous-bend paragraphs

Why are dangerous-bend paragraphs necessary at all? Couldn’t Knuth have put the advanced material into a chapter at the end of the book? Well, most chapters in *The T_EXbook* treat a major *concept*, as seen in the Table of Contents — a complete overview presented in one spread — and it seems that Knuth wants to keep advanced material about a concept in its chapter, i.e., in its context. For example, Chapter 5 introduces the concept *grouping* using ‘{’ and ‘}’; also known as *block structure*. Double dangerous-bend paragraphs at its end present the pair of T_EX commands `\begingroup` and `\endgroup`, which build a group that can start and end in different macros. (As mentioned above, all T_EX commands and parameters start with a so-called *escape character*, by default the backslash, i.e., ‘\’.)

Although some dangerous-bend paragraphs appear between the unmarked paragraphs in a chapter, the majority occur at the end of the chapter or, in Chapters 18 and 19, of each section.

Here is Barden’s complaint about the dangerous-bend paragraphs [3, p.167]: “Then, there is the issue of the dangerous bends signs. These signs begin on page five. Unfortunately, I have been warned against them — ‘don’t read the paragraph unless you need to’ is what the Preface says. Am I to understand that the dangerous bends on page five are as much to be skipped as the dangerous bends on page 212? No, I presume not, . . .” (The paragraph on page 5 starts with a second warning; see section 4.)

To be honest, I cannot follow the logic of this statement. *The T_EXbook*’s Preface warns about the

dangerous-bend signs; the author does extra work to mark paragraphs on page 5. Why should someone think the warning is not valid for this text? Only a person who assumes that the material is strictly ordered by difficulty, which is clearly not the case.

Beginners: Skip dangerous-bend paragraphs.

I'm convinced that a reader who does not follow Knuth's instructions gets frustrated, as Clark states. For example, page 5's dangerous-bend paragraphs require the knowledge of grouping, i.e., the topic of Chapter 5. You should also be aware of control sequences and how \TeX handles spaces after them; this is explained in Chapter 3 on page 8.

I think the main problem with the dangerous-bend paragraphs is that people overrate their abilities. They ignore the need for some development of their \TeX skills to master this material. Dangerous-bend paragraphs sometimes expect that you are able to apply the stuff of later chapters; that is, not yet introduced commands and concepts must be known. So skip the material, as true understanding is not yet possible and learning is low or frustrating.

Only after one has learned the basics of the topics presented in all chapters and has more practice can the single dangerous-bend paragraphs be read, while the double dangerous-bend paragraphs are for a third reading. Yes, a beginner has to read *The \TeX book* several times — at least three times. If you skip material as expected, you must read $\approx 110 + (110 + 120) + (110 + 120 + 120) = 690$ pages, not counting Appendices A, D, and I. (Repeated readings are probably required for all beginner's books; Samuel suggests it too [32, p. 1].)

Normal vs. dangerous-bend paragraphs. The normal, i.e., unmarked, paragraphs in *The \TeX book* introduce \TeX for every *beginner* who is able to work with a computer [18, p. v]: “*All that you really ought to know, before reading on, is how to get a file of text into your computer using a standard editing program.*” Thus, *The \TeX book* avoids jargon on this level when it describes \TeX . Terms like “control sequence”, “escape character”, etc. are carefully introduced and not assumed to be known. Sometimes certain aspects must be oversimplified in this level.

Chapter 6's request to “*go to the lab where the graphic output device is*” is outdated in our era of individual PC installations and services that require only a browser but everything else is still relevant.

Knuth sets a skill level for each command, but sometimes he mentions a higher-level command in a lower level. In unmarked paragraphs he probably does this on purpose as he often adds a reference for a later reading: See answer 5.1, pages 109 and 134

for `\kern`, `\looseness`, and `\catcode`, respectively. That he lists double-dangerous commands in single-dangerous contexts, to be comprehensive, seems acceptable: See pages 225 and 249 for `\cleaders & \xleaders` and `\span`, respectively.

Advanced material. Experienced readers find in the dangerous-bend paragraphs the details needed to master \TeX . In my opinion no other book comes close to these in-depth descriptions. That is probably one of the reasons why Hoenig makes the statement cited at the beginning of section 1.

The \TeX book has to present details, as Knuth sees neither \TeX without `plain` nor plain \TeX , i.e., \TeX with the format `plain`, as an endpoint for users. For example, other formats are shown in *The \TeX book's* Appendix E. Page 9 states: “*People hardly ever use \TeX 's primitive control sequences in their manuscripts, because the primitives are ... well ... so primitive.*” And on page 11 we find: “*However, you should keep in mind that plain \TeX is only one of countless formats that can be designed on top of \TeX 's primitives; if you want some other format, it will usually be possible to adapt \TeX so that it will handle whatever you have in mind. The best way to learn is probably to start with plain \TeX and to change its definitions, little by little, as you gain more experience.*”

Look at *The \TeX book's* format `manmac.tex`. It is based on `plain` but additional macros, sometimes replacing `plain's` macros, are added: font size macros, a new footnote macro, macros for two-column output, macros to start and end a chapter, etc.

Extremely advanced material. The double dangerous-bend paragraphs sometimes contain an additional warning. Besides advanced stuff that a reader can apply in their own texts, Knuth describes the details of \TeX 's algorithms. These facts fill several pages and are kept in their chapter instead of an appendix like Appendices G and H. Knuth gives a warning when such a detailed explanation starts; see page 94 or 123 for \TeX 's line-breaking or page-breaking procedure, respectively. Page 123 states: “*On the other hand, maybe you don't really want to read the rest of this chapter at all, ever.*” Only a \TeX wizard [18, p. 452] needs this information.

Nevertheless, even *The \TeX book* does not explain everything. (An example: `\showboxbreadth=0` acts like `plain's` default.) Guy L. Steele, Jr., says [35, p. 337]: “*There have been exactly two times, I think, that I was not able to fix a bug in my \TeX macros by reading *The \TeX book* and it was necessary to go ahead and read \TeX : The Program to find out exactly how a feature worked. In each case*

I was able to find my answer in 15 minutes because T_EX: The Program is so well documented and cross-referenced. That, in itself, is an eye-opener—the fact that a program can be so organized and so documented, so indexed, that you can find something quickly.” T_EX: The Program [19] is the second volume of *Computers & Typesetting*. Read it if you are a programmer.

My comments. It is absolutely important that a beginner follows Knuth’s advice and does not overrate their own abilities. Even if the reader has some knowledge about T_EX it is best to use the dangerous-bend signs as a guide for the first reading of the book. One gets an overview of its structure, sees Knuth’s style, learns to work with the exercises and at least with the Appendices A, B, F, and I.

4 Deliberate lying

When we explain a complex topic to a child we sometimes use simplifications up to a point that we become a liar. For example, we say: A hand has five fingers. The child doesn’t need to differentiate between finger and thumb yet. This teaching method is also used when we learn a language: A grammatical rule is stated and later the exceptions follow.

Knuth uses this teaching method in *The T_EXbook* [18, p. vi]: “*When certain concepts of T_EX are introduced informally, general rules will be stated; afterwards you will find that the rules aren’t strictly true. In general, the later chapters contain more reliable information than the earlier ones do. The author feels that this technique of deliberate lying will actually make it easier for you to learn the ideas.*”

Many people seem to read the above quote carelessly and assume that the book contains wrong information about plain T_EX. Barden writes in [3, p. 167]: “*The first thing that struck me in the Preface was that the author is going to tell lies and that ‘the later chapters contain more reliable information than the earlier ones do.’ Whereas, I presume the author is not going to give me straightforwardly incorrect information, perhaps my approach should be to blame him for my typesetting mistakes.*”

Let’s look at an example of how Knuth introduces a concept to beginners via a high-level overview, i.e., without directly stating every detail or every exception. As he writes in a colloquial style for someone who has typed text using a typewriter or editor, the following statement on page 64 (Chap. 11 “Boxes”) agrees with the experience of such a reader: “*Everything on a page that has been typeset by T_EX is made up of simple character boxes or rule boxes, pasted together in combination. T_EX pastes boxes*

together in two ways, either horizontally or vertically.” Five pages later the reader learns that a space is not an empty box: “*And there is space between words too; such space is not an ‘empty’ box, it is part of the glue between boxes. This glue can stretch or shrink so that the right-hand margin of each page comes out looking straight.*” This explains justified paragraphs, which one does not get with a typewriter. Next, on page 95, Knuth presents all elements of horizontal lists with this introduction: “*We have been saying informally that a horizontal list consists of boxes and glue; the truth is that boxes and glue aren’t the whole story.*” This development seems to be necessary for someone who has never heard about T_EX’s *boxes*, *glue*, and *horizontal mode*. Terms that are described in Chapters 11 to 13 before Knuth can reveal the complete truth in Chapter 14. It is a teaching method to build upon known concepts before T_EX’s new concepts are explained.

Skill levels. Knuth plays with them at least two times. First, on page 5 in the very first dangerous-bend paragraph that starts with the question: “*Are you sure you should be reading this paragraph?*” Second, on pages 43–44 when he explains three times the definition of the control sequence `\%` and asks the reader to check out Appendix B. The explanations appear (1) in an unmarked paragraph, (2) in a single dangerous-bend paragraph, and (3) in a double dangerous-bend paragraph. As the first two explanations are wrong one might say that there are two lies but for me it looks more like two tests. Only a lazy reader, who doesn’t consult Appendix B at least when the second explanation is found, is tricked.

White lies about *The T_EXbook* itself. The first white lie of this kind (my counting) occurs on page 3 and has nothing to do with T_EX. The letters that *The T_EXbook* uses for the input are introduced but on page 45 another letter, the Scandinavian glyph ‘æ’, appears in a double dangerous-bend paragraph. Another one is on the first page of Chapter 3, when Knuth writes “*we shall escape via backslashes in all the examples of this manual*” although on page 40 an example occurs in a double dangerous-bend exercise that uses a slash as the escape character.

I imagine no one complains about such things. The author assures the beginner that something is simple or standardized in the book. Later, for the experienced users, he shows advanced techniques, for example, how to change T_EX’s escape character.

I also find it harmless that page 29 states a line has badness 5000 although its badness is 4995; the correct value is given in the file `texbook.tex`. (“Badness” is a heuristic measure for how much glue

stretches or shrinks; not all values are possible. The value 5000 cannot be achieved.)

Small deviations often occur when Knuth makes a statement about Appendix B; see also the previous subsection. After the mentioned test a reader shouldn't trust such statements. But they are all inoffensive. For example, instead of the \TeX assignment `\hsize = 6.5 true in`, as stated on page 60, Appendix B contains the equivalent but more compact `\hsize6.5truein` on page 364.

How much may a rule be relaxed? Let me ask three questions: (1) Do we see a white lie if an incomplete general rule is presented in one paragraph and its exceptions are given in the next? (2) What if the first paragraph is unmarked and the second preceded by a dangerous-bend sign? (3) What if other paragraphs appear between rule and exception?

I answer the first question with “no”; an author must have some freedom of exposition. As long as everything is kept close together no harm is done.

The first example with the scenario of the second question appears on page 25. An unmarked paragraph states: “*If the first character of your response to ‘**’ is not a backslash, \TeX automatically inserts ‘\input’.*” The next dangerous-bend paragraph adds this information: “*If the first character after ** is an ampersand (‘&’), \TeX will replace its memory with a precomputed format file before proceeding. Thus, for example, you can type ‘&plain \input story’ ...*” The beginner should only use `plain`. The command to load another format should not be taught in an unmarked paragraph.

I also find acceptable what I described in section 3: At times *The \TeX book* oversimplifies \TeX 's behavior. See page 94: An unmarked paragraph starting with “*Roughly speaking*” summarizes \TeX 's line-breaking procedure. Next, twelve pages with dangerous-bend paragraphs explain all the details.

The situation of the third question can be found in dangerous-bend paragraphs on pages 78–79. On page 78 we find: “*Whenever a box is added to a vertical list, \TeX inserts ‘interline glue’ intended to make the distance between the baseline of the new box and the baseline of the previous box exactly equal to the value of `\baselineskip`.*” And after describing “interline glue” in more detail, page 79 states: “*Exception: No interline glue is inserted before or after a rule box.*” I think this is permissible as the reader who learns the concept isn't supposed to stop in the middle of the topic.

Sure, a user who reads only page 78 might miss the exception. *The \TeX book* is the definitive user manual and reference manual of \TeX ; its index entry

“interline glue” first returns the page range 78–79. The authoritative description gives a double dangerous-bend paragraph on page 80. It's the second reference for this entry and underlined (see section 9).

Is it a lie or a fib or a reader's notion? Often an incomplete general rule is marked as such through its wording, so that a careful reader sees that there's something more to be learned. For example, page 7 contains this statement: “*In case you're wondering what a ‘letter’ is, the answer is that \TeX normally regards the 52 symbols A...Z and a...z as letters.*” The word “normally” avoids that a “rule” becomes wrong. On page 25 an “essentially” keeps the general rule from becoming incorrect: “*Every time you begin a new line in your manuscript file it is essentially the same as typing a space.*” Page 37 shows the complete rule.

When one looks at what is listed in the previous paragraph, it seems that *The \TeX book* is carefully formulated to avoid the reader extracting a rule when it does not want to state one. Maybe the warning in the Preface also helps protect the text from wrong conclusions.

Knuth seems to expect bad inferences; he writes on page 37: “... \TeX responds to 16 different types of characters. At first this manual led you to believe that there were just two types — the escape character and the others — and then you were told about two more types, the grouping symbols { and }. In Chapter 6 you learned two more: ~ and %.” Earlier on page 7 he writes: “*In order to make a limited keyboard sufficiently versatile, one of the characters that you can type is reserved for special use, and it is called the escape character.*” I don't see this as a statement about a rule that only two types exist. Half a page later, letters and digits are described as being different, i.e., no reader should only assume two types of characters at the end of page 7.

Rewards. Knuth asks readers to report errors in his books. The first person who reports a valid typographical, technical, or historical error gets a little reward: A certificate for a hexadecimal dollar with a value of \$2.56.

This offer also exists for *The \TeX book*, but unlike most of his other books, he reviews errors only once every several years during the so-called *\TeX tune-ups* [30]. Between two tune-ups experts vet the reports, collect the valid ones, and forward them to Knuth when the moment for his evaluation comes.

One result of these rewards is that people read *The \TeX book* with an extreme persnickiness to detect an incorrect statement. No one reads any other book about \TeX with such scrutiny.

Because of Knuth’s deliberate lying there are many opportunities for readers to report imprecise wording. But such reports are rejected.

My advice is to check your report by answering the following. 1) Is the to-be-fixed text in an unmarked paragraph? 2) Is the to-be-fixed text not part of the chapter’s main theme? 3) Does a later paragraph in the chapter contain the missing information? 4) Does a later chapter contain the missing information? 5) Does your fix use jargon or a technical term not yet introduced? 6) Does your fix correct the value of a heuristic or binary-to-decimal or fixed-point calculation? The more “yes” answers, the smaller the chance that your report is valid.

Reports whose fix has an aesthetic impact must be checked against this statement [30, p. 6]: *“And in fact I’ve even become somewhat fond of such little glitches, now that I’ve been learning to appreciate the Japanese concept of wabi-sabi.”*

Truth. Knuth occasionally avoids a deliberate lie although the description could be shorter and maybe the understanding easier. For example, on page 104 he describes the places where \TeX inserts the parameter `\clubpenalty`. The text is marked with one dangerous-bend sign as a beginner does not need to know it. Knuth allows himself to write a precise but complicated explanation. *“Here’s how interline penalties are calculated: \TeX has just chosen the breakpoints for some paragraph, or for some partial paragraph that precedes a displayed equation; and n lines have been formed. The penalty between lines j and $j+1$, given a value of j in the range $1 \leq j < n$, is the value of `\interlinepenalty` plus additional charges made in special cases: The `\clubpenalty` is added if $j = 1$, i.e., just after the first line;”*

\TeX by Topic [9] gives a simpler description of `\clubpenalty`, see pages 207 and 270: *“Additional penalty for breaking a page after the first line of a paragraph.”* This wording is misleading/incorrect for (a) a one-line paragraph (above: $n > 1$) and (b) for the paragraph that \TeX builds from the input *“Have you ever seen $\text{\$E=mc^2}$ in a children’s book?”*. (The *displayed equation* between the dollar signs is placed on a line by itself preceded by a partial paragraph.) It misses (c) that this penalty can be inserted several times; for example, twice in a, say, seven-line paragraph if $\text{\$E=mc^2}$ occurs in the middle. (More context on page 209 allows a reader to infer that (a) and (b) are excluded, but an example on page 219 ignores this information again.)

All books contain errors and I don’t mention this case to blame someone. It shows how difficult it can be to describe \TeX ’s behavior correctly. For

most users the incorrect statement is sufficient as it indicates a way to *avoid* a page break between the first and second line of a paragraph, which leaves a so-called club line. But one needs the correct description to understand the topic “output routines”.

My comments. I suggest reading the text carefully if general statements are made. The above examples show that often a single word indicates that presumed rules don’t describe the complete truth. On the other hand I’m convinced that a beginner forgets the restriction and remembers only the inexact rule. To bring the careful formulation back to the reader’s mind is one of the reasons why a reader who starts to read the dangerous-bend paragraphs should reread the unmarked ones too.

5 Jokes

Yannis Haralambous summarizes Knuth’s humor in one sentence [13, p. 14]: *“Don’s humor is proverbial, irreverent, unrestrainable and ubiquitous.”* Andrew Booth seems to like Knuth’s style, as the quotation at the beginning shows: The book is *“enjoyable”* and *“full of wit”*. On the other hand, Bob Floyd wrote to Knuth [27, p. 6] about his humor in another book: *“I like the chapter, but I think it could be improved by chopping most of the humor”*

Knuth about his humor [24, p. 124]: *“Some of you who know me will realize that I have a peculiar sense of humor. I occasionally say preposterous things under the assumption that nobody will take them seriously.”* He made this statement during one of his “Theory and Practice” talks [24, Chaps. 6–9] about which he writes [24, p. x]: *“When I try to characterize my own life’s work, I think of it primarily as an attempt to balance theoretical studies with practical achievements.”* In 2011 he changed the title and talked about “Theory and Practice and Fun” [29, pp. 39–40]. The completion of his *Fun & Games* book [28] inspired him because he realized that *“the joy of such so-called ‘aha moments’”*, which he takes to be the driving force behind *“scientific discoveries and advances in technology”*, is an important aspect of doing his work.

Knuth’s humor manifests itself in many ways: riddles in a text [28, p. xiv], wordplays/puns, self-references, surprising index entries, nonsense in the style of a serious scientific article [28, Chap. 6], credits in DVI files, . . . , and jokes inserted in the text. Of course, it depends on the reader if this is considered funny. We find at least the second, the third, the fourth [38, but part 7], and the last item of this list in *The \TeX book*. More examples can be found in other volumes of *Computers & Typesetting*. For

A look at complaints about *The \TeX book*

example, self-references are also used in some letters of Volume E [25] and credits in DVI files are implemented in the source code of \TeX , see [19, §591].

Sometimes Knuth plays a trick on people; he might prefer to call it an “experiment” [29, p. 186]. He does this in *The \TeX book* too; see section 4.

Jokes in *The \TeX book*. With words that sound like a joke, Knuth states [18, p. vi]: “*However, most of the jokes can only be appreciated properly if you understand a technical point that is being made — so read carefully.*” Or, conversely: Every time you find a surprising or confusing statement, look closely, if it might belong to a joke that you missed because you do not understand the preceding paragraphs.

In [3, p. 167] we find: “*Then, we have the issue of jokes to enliven the dullness of the material. The problem with the jokes is that I may not appreciate them properly unless I understand the technical point that is being made. Already I feel as thick as two short planks.*” Nay, consider the detection of jokes as a test for yourself on the subject matter.

Here is such a joke that you can appreciate only if you understand the technical point; otherwise you might think Knuth made a typo in the last word. The following quotation is from the first chapter about mathematical typesetting, i.e., *The \TeX book’s* Chapter 16, page 132. Remember, Knuth writes for everyone including users without a background in mathematics. “*Furthermore, you should be careful to distinguish between ‘oh’ and ‘zero’: The italic letter O is almost never used in formulas unless it appears just before a left parenthesis, as in ‘ $O(n)$ ’; and the numeral 0 is almost never used just before a left parenthesis unless it is preceded by another digit, as in ‘ $10(n - 1)$ ’. Watch for left parentheses and you’ll be OK .*” Is it dangerous for a reader if the bad input “ OK ” (with 0 not O) is not detected? The joke helps someone without mathematical training to understand that the technical point is only valid for formulas, not for ordinary text. (The distinction between ‘ O ’ and ‘0’ in a font is for Knuth a serious topic; see *TUGboat* **35:3** (2014), 232–234, and **36:3** (2015), 191–199.)

The joke above involves an easy to detect typo; of course, this is unusual. See this example on page 223. (*Leaders* connect two pieces of text, for example, with a line of periods as in a Table of Contents; *glue* is variable-width white space (see section 4).) “ *\TeX treats leaders as a special case of glue; no, wait, it’s the other way around: \TeX treats glue as a special case of leaders. Ordinary glue fills space with nothing, while leaders fill space with any desired thing.*” Technical point: The `\leaders` primi-

tive uses a box; together they prefix the wanted glue. Put the “desired thing” into the box to fill the space; an empty box gets “nothing” in a complicated way.

In puns Knuth plays often with the pronunciation of \TeX ; see section 7. Another kind of funny reference: Knuth also applauds the jokes of others; for example, pages 410–411 mention P.D.Q. Bach.

Quotations. Knuth places quotations at the end of each chapter. For example, here is a posthumously published quote from the German poet and writer Goethe (p. 183): “*Mathematicians are like Frenchmen: whenever you say something to them, they translate it into their own language, and at once it is something entirely different.*” Other quotes use words that played a role in the chapter but refer to something different because of the context. For example, page 83 at the end of the chapter “Glue” quotes Mark Twain’s *Huckleberry Finn*: “*There was things which he stretched, but mainly he told the truth.*” Of course, Knuth uses the quotations also for other facets of his humor. For example, *The \TeX book* quotes \TeX (p. 287) and itself (p. 337).

My comments. Don’t be afraid of jokes; they do not hide but illustrate technical points as the above examples show. Sure, jokes hardly occur in textbooks or user manuals. Thus, they constitute an unusual feature for a reader. As mentioned, the margin is wide and you can mark jokes to skip them in the next reading if they stop your concentration.

6 \TeX ’s anatomy

One might find it a bad decision but Knuth opts to treat \TeX as an individual. In error messages and their help texts \TeX uses the pronoun “I”. Other do that too: See Clark’s quote in section 2 that refers to \TeX with “him/her”.

Knuth also gives \TeX a body. *The \TeX book* mentions “mouth” (once “mastication”) and “stomach” eleven times each. Appendix D “Dirty tricks”, marked with eleven dangerous-bend signs, contains 55% of the occurrences for \TeX ’s mouth. Surprisingly, “stomach” appears most often in Chapter 7 “How \TeX Reads What You Type”. The phrase “digestive process” is seen thirteen times and the seven words for other body parts or processes — bowels, eyes, gastrointestinal tract, gullet, guts, regurgitation, tummy — occur together twelve times. In total, \TeX ’s body is mentioned 47 times.

All of \TeX ’s body parts are presented in dangerous-bend paragraphs. (Exception: The second underlined page number in the index entry “anatomy of \TeX ” sends the reader to a drawing of the \TeX

lion by Duane Bibby at the start of the index.) No beginner who follows Knuth’s advice reads about eyes, mouth, or the gastrointestinal tract. Only an experienced user, who must have observed Knuth’s humor during repeated readings of *The T_EXbook*, is confronted with this level of funniness.

In [3, p.169] we find: “*Secondly, I would ask that manual writers be more careful about the imagery used. There is something vaguely revolting about the idea of these tokens being read by the eyes, fed through the mouth, down the gullet, to the stomach, into the intestine and then being sicked up again on receiving a new set of instructions. Even more revolting is a consideration of the route after the tokens have reached the intestines.*”

T_EX is a complex program that seems to work on different levels simultaneously. In [9] four “processors” — called input, expansion, execution, and visual processor — are described. Knuth avoids such technical terms or programmer’s jargon and presents a different view based on a living organism with eyes, mouth, stomach, and bowels. This metaphor can be understood by everyone. It supports the conception that a lot of things in T_EX happen in parallel at different places and also gives an explanation for a kind of sequential processing of the input. These aspects aren’t obvious from the names of the four processors.

Nevertheless, the imagery can be extended to T_EX’s final output as mentioned in the quote. But experienced T_EX users do not care as it does not matter anymore for any T_EX user or macro programmer what happens after the stomach. The program T_EX deals with these body parts, not the user. Except that every now and then an expert slips into the role of a doctor and takes one of T_EX’s provided stethoscopes, like `\tracingpages` [18, pp.112–113], to auscultate T_EX’s abdomen.

My comments. The technical terms did not replace the body parts; it seems that most people in the T_EX community understand and use *The T_EXbook*’s wording. As mentioned above, it is for experienced users, especially macro programmers. They work with three body parts — mouth (and gullet), stomach, and (less often) eyes — as they give a quick way to communicate about used methods.

Note: The imagery cannot be used as an excuse not to check T_EX’s final output carefully. Proofread!

7 T_EX’s name and its pronunciation

When I look at Knuth’s collected works, I notice that he was always concerned about notation and names of concepts. In [29, p.195] he says “*You don’t need a new name for something unless it’s going to*

be used over and over again, unless it’s an important combination of concepts.” His original name was “TEX”; see [29, p.63]: “*... the name ‘TEX’ came to me early on. ... That name changed to ‘T_EX’ quite awhile later. But I did have a name, yes; the minute TEX was named, I started to get ideas about it.*”

As said in section 6, T_EX is treated as an individual and Knuth assigned it a three-letter acronym (TLA): tau epsilon chi. (It might be a sense of Knuth’s humor to extend the number of TLAs by suggesting to use the Greek alphabet.) It is a clever choice, as the letters form the word stem “tech” in modern English and are thus reminiscent of “technology” and, at the same time, the Latin spelling “tex” creates a link to the word “text”. In *The T_EXbook*’s Chapter 1 Knuth explains that TEX was already in use by a commercial system by Honeywell Information Systems and that Knuth therefore decided to give T_EX a special pronunciation, a logo, and the ASCII-spelling “TeX” to distinguish the systems.

He also starts to replace “tech” in other words by “T_EX” or he joins words like “T_EX” and “hacker” to “T_EXacker” emphasizing the correct pronunciation of T_EX. The Preface, page v, has them still separated; see the quote in section 2. This is part of his humor, these words are jokes or, better, puns; see *The T_EXbook*’s index entry “T_EX, bad puns on”.

Some readers might feel patronized by the statements or dislike the puns; we find in [3, p.167]: “*The first exercise is the one to ensure that one knows the correct pronunciation of the word T_EX. However, when the answer is consulted, one is thrown into more confusion — what’s a T_EXacker? I know what a hacker is, but I do not know what an acker is. ... Would the author not have been better employed teaching me how to typeset the name?*” This question gets its answer on page 8 after the escape character is known. Chapter 1 “The Name of the Game” explains nothing about T_EX’s required input.

My comments. When T_EX was programmed there was obviously a need to avoid confusion; nowadays no one mistakes TEX for TeX or T_EX. However, I’m convinced that all the precautions, including the pronunciation, helped there to be no incompatible versions of the original T_EX. And as one would pronounce the name of a foreign friend as best as one can in that foreign language, one should try to pronounce ‘T_EX’ in the way it was specified.

8 The structure of *The T_EXbook*

Although I found no concrete complaints about *The T_EXbook*’s structure, it is time to look at its con-

tents. The statement by Guy L. Steele, Jr., in section 3 seems to indicate that it provides a complete description of what one needs.

The T_EXbook consists of 27 chapters and ten appendices. In section 7 we already looked at Chapter 1. A beginner who expected some motivation from the text must wait till Chapter 2 “Book Printing versus Ordinary Typing”. The first typographic terms and one of T_EX’s *special characters*, the dollar symbol ‘\$’ for *inline math mode*, are introduced.

Chapter 3 “Controlling T_EX” teaches the first T_EX command after the beginner learned the escape character. It is one of the many chapters that discuss a *concept* of T_EX. Beginners must learn all of T_EX’s concepts as nearly all of them are new if one has worked only with word processors before. Up to Chapter 15 all concepts that T_EX needs to typeset texts are taught. Information about typography and motivational hints continue to appear.

Two examples: Chapter 4 “Fonts of Type” also increases the motivation to learn T_EX as it shows a collection of fonts. In T_EX’s early days a typist gets this at best from a golf-ball typewriter and then with a lot of work. Next, Chapter 6 “Running T_EX” explains in detail how to prepare an input file, run it through T_EX, and print its typeset output. I surmise Knuth wants to present this hands-on exercise as early as possible but he needs a few concepts — T_EX’s command syntax and from Chapter 5 “Grouping” the meaning of the braces ‘{’ and ‘}’, which are introduced in Chapter 4.

Then seven chapters “How T_EX Reads What You Type”, “The Characters You Type”, “T_EX’s Roman Fonts”, “Dimensions”, “Boxes”, “Glue”, and “Modes” prepare every reader for what one might think are the main chapters of the book: Chapter 14 “How T_EX Breaks Paragraphs into Lines” together with its companion Chapter 15 “How T_EX Makes Lines into Pages”. At least they describe why we learn T_EX: We want to fill pages with our own texts.

Knuth did not develop T_EX because he saw a market for a typesetting system. It was a personal need to invent T_EX: His books could not be typeset in the high quality that he desired because technology changed and manual composition was no longer affordable for the publisher [29, pp. 57–60]. As his books contain a lot of mathematics, T_EX must be able not only to typeset text but also mathematical formulas. Thus, in Chapters 16 to 19, *The T_EXbook* describes the concepts and the details for high-quality typesetting of mathematics.

Chapters 20 and 21 treat topics that do indirectly belong to T_EX’s typesetting concepts; they describe how the underlying structures can be ap-

plied by users. Chapter 20 “Definitions (also called Macros)” is about macro programming and Chapter 21 “Making Boxes” explains how to make use of T_EX’s boxes and T_EX’s file input and output features. Sometimes I wish that Chapter 22 “Alignment” was presented earlier as it is something that users, who typeset texts, need. Sure, it works with two special characters, which are now introduced independently in Chapters 18 and 20. Next, Chapter 23 describes the method that T_EX uses to build pages, i.e., it is a kind of continuation of Chapter 15 with an application of macro programming.

Chapter 20 also contains a summary of T_EX’s primitives that are handled mainly in T_EX’s mouth (`\string` is one); see pages 209–215. On pages 266–293 we find the corresponding summary of primitives handled in the stomach. These pages, split by T_EX’s modes, build Chapters 24–26. They explain how numbers, dimensions, etc. must be constructed, list all parameters with a short description of what each one does, present the syntax of assignments and definitions in T_EX, and give for each remaining primitive a short paragraph with its description.

Chapter 27 carries the title “Recovery from Errors”. Of course, a beginner needs some information about this topic much earlier and Chapter 6 does that. During the hands-on exercise Knuth guides readers through T_EX’s error reporting method.

The book continues with Appendices A to J whose titles start with the letter that names the appendix: “Answers to All the Exercises”, “Basic Control Sequences”, “Character Codes”, “Dirty Tricks”, “Example Formats”, “Font Tables”, “Generating Boxes from Formulas”, “Hyphenation”, “Index”, and “Joining the T_EX Community”. It is a mixture of look-up information, detailed examples for experienced users (D and E), highly specialized facts about T_EX’s internals (G and H), and a call to enthusiastic users to join the community.

My comments. *The T_EXbook* teaches step by step the concepts that users need to understand to work with T_EX. The concepts are often intertwined, and before a deeper understanding of some are possible, it’s necessary to learn others to a certain extent.

9 *The T_EXbook’s index*

A. Barden does not only criticize, but also praises some aspects of *The T_EXbook* [3, p.167]: “*What was helpful in this book? The comprehensive index was helpful, as was page 340. On this page are the commands for producing the format on page 341.*”

Not every user finds the index helpful. Gary S. Wagman writes in T_EXhax (see section 2), Vol. 89,

Issue 77: “Donald Knuth is brilliant, but his \TeX -book is far from what I expect of a programmer reference manual (I only have ten fingers to stick in the book while looking for index references) and it certainly is not a usable user guide.” The statement shows that he never read the book in the suggested way; see section 3. *The \TeX book* is not written to be a programmer reference manual. Maybe complaints about the index express the wish for a different kind of book. Amy Hendrickson writes [14, p. 138]: “We \TeX ies have all spent hours poring over *The \TeX -book*, enjoying its lovely writing, its complex concepts, and its illuminating examples. One cannot help admiring it and its author, of course. Still, it must be admitted, when a macro writer is struggling to find some particular bit of information, a quick definition, a nudge in the right direction, he or she can find *The \TeX book index* thoroughly unhelpful and the bit of information wanted scattered over many different pages in many different chapters.”

When one cannot remember the name of a command, one should look up the associated \TeX concept. In any case, now one has a concrete index entry and should check the listed page numbers. *The \TeX book* starts the index on page 457 with an introduction: “Page numbers are underlined in the index when they represent the definition or the main source of information about whatever is being indexed. (Underlined entries are the most definitive, but not necessarily the easiest for a beginner to understand.) A page number is given in italics (e.g., ‘123’) when that page contains an instructive example of how the concept in question might be used.”

However, P. Abrahams writes [1, p. 303]: “The \TeX book is indispensable as the definitive source of information on \TeX , but a big problem with *The \TeX book* is that it’s very hard to retrieve information from it. Essential topics appear in odd places, e.g., the description of registers in the middle of the chapter on How \TeX Makes Lines into Pages. The index has so many references under each entry that it’s hard to locate the definitive one, and the one that’s marked as definitive often gives just the syntax of a command, not its semantics.” First, registers aren’t a major typesetting concept needing an own chapter; they provide storage space. Second, the quote’s last sentence is too general. For example, the index entry “registers” has four references: to the stated chapter, to *token registers* in Chapter 20, to Chapter 20’s summary (see section 8) for the primitive that extracts a register’s value, and to a list of registers used by `plain`. The specific register type “`\count` register” to store numbers has six references: the first for Chapter 15, the second for

the comparisons of numbers, then two underlined for the formal definition in the summaries (see section 8), and two in italics for examples (one is for the registers in `plain`). Is that too many references?

Only 15% of the page references are underlined and 21% are in italics. Less than 3% of the index entries/subentries have ≥ 10 references; more than two thirds have one to three. The entry ‘`{}`’ has the most references: 18 with 8 in italics and 0 underlined (for short: (10, 8, 0)). It is followed by ‘grouping’ with (16, 0, 1); two subentries have (1, 1, 0) and (4, 0, 3). The parameter `\hsize` for the *horizontal size* of a text collects the third most references: (7, 9, 0).

My comments. Beginners: First, study page 457 completely. Then look up the first or second reference for the index entry that you want to check. Also visit pages in the range 23–34 (Chapter 6) and page 340 (the large example) if it is listed. But exclude the summaries on pages 209–215 and 266–293.

Macro writers: Test the programmer reference manual [4]. Or read *The \TeX book* in the suggested way, see section 3, if you never did that. Then you know the book better and find things more easily.

10 *The \TeX book* for beginners

A. Barden: “*This book is for the serious student and not for the person who wishes to typeset as quickly as possible and have done with it.*” ([3, pp. 167])

i) A compact introduction. A beginner who follows Knuth’s advice has to read ≈ 110 pages; see section 2. Maybe it’s best not to use the computer for the first dozens of these pages and work only with paper and pencil until one gets to Chapter 6 “Running \TeX ” where a complete text is fed into \TeX .

The \TeX book is not easy reading: Its reader must be self-controlled and concentrated to study it in depth. I write “self-controlled” because of its dangerous-bend signs. After a beginner has mastered all unmarked paragraphs the earned title is “ \TeX nical typist” [18, p. 11] not “ \TeX Master” [18, p. 245]; one needs time to learn \TeX through *The \TeX book*.

ii) Input/output example. All beginners should also consult pages 339–342, say, after Chapter 14 or 15 to see a kind of quick reference guide. It contains an input/output comparison too. For many beginners it might give a simple starting point.

Such a before-after comparison is usually shown early in books for beginners; for example, [34, pp. 18 & 19] and [7, pp. 19 & 21] (without using any control sequences). It looks a little bit like in these books the comparisons are used to convince a reader that it’s worthwhile to learn \TeX .

iii) Hands-on exercise. *The T_EXbook* wants a reader to practice T_EX and gain some routine in its use. It guides a beginner to enter the file `story.tex` and to create its printed output in several runs with different parameters in Chapter 6; this is a *hands-on exercise*. One learns a lot if the reader is not too lazy to do this exercise.

iv) Exercises and examples in the text. *The T_EXbook* contains a large number of examples and exercises; see Booth’s comment in section 2. Every reader should try to solve the exercises that belong to the current skill level — at least a beginner should work on all unmarked exercises to see the benefits. After one worked on an exercise, even if it was solved, the answer in *The T_EXbook*’s Appendix A should be read, as it often has additional information. If one has worked on the exercise, the answer is usually easy to understand. Knuth requests this procedure in *The T_EXbook*’s Preface.

Knuth also lets us learn from errors he made; see [18, p. 211]. Perhaps even more important are his descriptions of how he made improvements to the typeset result after it was created, as the original planning could not foresee everything. For example, read the answer to exercise 22.1; it shows that no user should skip proofreading. T_EX enables the highest typographic quality, but it still needs the help of a human to achieve it in every situation.

v) Vocabulary. One important difference between a beginner and a more experienced user is that the latter is able to articulate problems in the terms that T_EX experts know. *The T_EXbook* combines words used in typography with new ones introduced by Knuth. For example, each concept represents a well-defined subject area that is best described by the word (or a name of an associated primitive) invented by Knuth. (It’s not because computers are involved; every more complex human task, for example, cooking, develops a specialized vocabulary.)

A beginner often lacks the right vocabulary. A study of *The T_EXbook* gives you the right words. So you can ask, for example, on TeX.SE questions using the terminology that others can easily understand.

vi) No jargon. Some people find that *The T_EXbook* uses programming jargon. For example, from [7, p. 99]: “A great many of T_EX’s mysteries are bound up in commands. Alternative terms are ‘macros’, ‘control sequences’, or ‘definitions’: although Knuth refers to them as macros, unless you are familiar with computer science jargon, there is no advantage in this usage.” Besides that *macros* are only a subset of T_EX’s control sequences, up to Chapter 20 Knuth uses this term only in dangerous-bend

paragraphs with one exception on page 44 where he writes about “the macros of Appendix E”.

Others are confused by the names of primitives and parameters. For example, [31, p. 561] states: “Not only is *The T_EXbook* oriented toward programming, but a lot of the commands don’t even look like they’re in English!” My impression is that programmers often use abbreviations for primitives as the names are *too verbose* because they are in English. (For example, some let `\ea` represent the double-dangerous primitive called `\expandafter`.) Besides the typeface switches, `\rm`, `\it`, etc., T_EX abbreviates for a beginner only names that refer to a T_EX mode. This leads, for example, to parameters with names like `\hsize` where ‘h’ stands for “horizontal”. And this is easily deduced from the text; see page 27.

vii) A path to learn. Normally, with more practice a user becomes more experienced. Then the user either starts to read *The T_EXbook*’s unmarked paragraphs a second time or adds to their rereading the single dangerous-bend paragraphs. In the second case the user will not see much difference in the first eleven chapters. Only in chapters 4, 8, and 10 does the new material add a noticeable amount of text. But in chapters 12–15, 18, and 20–23 the new material occupies more pages than the old one; chapters 16, 17, and 19 grow between 40% and 80%.

The intermediate user who keeps reading *The T_EXbook* still receives information (and reminders) how to achieve the best quality for the typeset output. Such a user does not struggle with T_EX in the way a beginner does. For example, such a user has learned to check during proofreading line breaks for unintended patterns. Moreover, such a user also starts to look at page breaks and rewrites the text or applies `\looseness`. A unique feature of *The T_EXbook* is that it gives a structured path to achieve different skill levels.

What more experienced users know is how to acquire new or refresh existing knowledge to reach the desired output. Even if a person applies T_EX frequently it can happen that one forgets certain details of rarely used T_EX commands or parameters. Then one might need *The T_EXbook* to look up the information.

viii) No programming. Recall that the beginner addressed by *The T_EXbook* is neither required to know how to program nor to realize how the setting of various parameters influences T_EX’s output.

But often beginners want to change the layout and implement their own ideas. Although Chapter 6 explains how to change the width of a text line, it does not show how to change the text height. Such

changes are sometimes necessary, for example, if a user wants to work with DIN-A4 paper. For beginners *The T_EXbook* shows how to set the parameters only on pages 251 and 340. So one must be patient.

A challenge occurs with T_EX's default fonts if a reader wants to write in a non-English language and has, for example, to enter accented letters with several keystrokes. T_EX can work with other fonts but one must install them. Advanced users can read [15]; it's the only cited reference that handles this topic. (More must be changed, for example, to get correct hyphenations. So either be patient until you can do all that or get help from a local users group.)

ix) Macros. A beginner might think that macro programming distinguishes beginners and experts. Sure, *The T_EXbook* guards most of the macro programming material with two dangerous-bend signs. But remember, *The T_EXbook*'s readers acquire skills to achieve high-quality typesetting results, not to turn into macro programmers. The book explains for beginners how simple macros help to get a consistent output. Double dangerous-bend paragraphs teach macro programming techniques in Chapter 20 and Appendices D and E on more than 70 pages.

Knuth writes ([26, p. 235]): “*T_EX was designed for typesetting, not for programming; . . .*” Or as he said in an interview [35, p. 597] many years later: “*I was really thinking of T_EX as something that the more programming it had in it, the less it was doing its real mission of typesetting. [\par] When I put in the calculation of prime numbers into the T_EX manual I was not thinking of this as the way to use T_EX. I was thinking, ‘Oh, by the way, look at this: dogs can stand on their hind legs and T_EX can calculate prime numbers.’*”

What readers probably need to learn is how to modify or create their own page layouts; see also the previous subsection. It is not difficult to write a letter with plain T_EX but it gets much easier if one is able to write a few macros that implement the well-known structure of letters with a letterhead, the current date, address label, etc. Appendix E of *The T_EXbook*, pages 403–408, shows how one can do that. Of course, you must adopt the presented macros for your letterhead and, in general, the whole layout and format to the conventions in your country. (Several programmers share their code; see the directory `macros/plain/contrib` of the *Comprehensive T_EX Archive Network* (CTAN).)

x) Your own macros. I suggest keeping macro packages simple, i.e., do not add bells and whistles that you don't need. Reuse code but change only what you need for the new project and don't try to

create a universal macro with many parameters that work for all your projects. Moreover, use verbose macro names so that you know what they do even if you have not worked with them for a while.

For a book project one might need an individual macro package that is easily modified if requirements change. This task requires a kind of software design, something that computer scientists learn. It's a job for a T_EX wizard who is able to understand the book design of a typographer and skilled enough to implement this design in T_EX.

Who should start with *The T_EXbook*? *The T_EXbook* might be a good book but is it still worth reading by beginners nowadays, as there are other formats and programs, more modern, for example, with support for UTF-8, PDF tagging, and so on?

If you want to learn the original T_EX with the `plain` format, then read *The T_EXbook*. But, please, follow the recommendations of section 3 even if you already know another variant. Plain T_EX used with only the Computer Modern fonts is very stable: All such documents since 1990 are typeset today and in the future with the same line and page breaks.

If you want to learn T_EX but have not yet decided which variant, then consider your reasons for learning T_EX and not another system. Good reasons to start with a variant other than `plain` are, for example, having the chance to participate in a class about this variant from a trusted organization, or your friends using this variant for their productive use. You might also look at, for example, the T_EX Users Group's page <https://tug.org/begin>, to get an overview about the complex T_EX world. Moreover, on this page you find information about methods to install T_EX on your computer.

No involuntary user asks the above questions. Such a user is presented with the task of learning something that this person never intended to learn. For example, your supervisor requests the use of T_EX for your thesis. I expect in this case that the variant is also specified. Otherwise, follow the advice of the previous paragraph.

There is nothing wrong with learning T_EX from *The T_EXbook* if you are still undecided. As mentioned above, it is suitable for self-study. Start to read the paragraphs without a dangerous-bend sign and you learn all fundamental T_EX concepts. Stop at the level that is sufficient for your needs. You can reuse what you learned if you switch the format or the engine. For example, LuaT_EX is another engine, much younger than T_EX, but based on it. And *The T_EXbook* is still relevant: “*This is the reference manual of LuaT_EX*. [v. 1.21, 2025-02-10] . . . we as-

sume that the reader knows about \TeX as described in ‘The \TeX book’,”

11 Digression: Books for beginners?

Let us look at two books with other approaches. The first aims at macro programmers, the second wants to teach \TeX without discussing its concepts.

Why not recommend reading \TeX by Topic first? Many people find \TeX by Topic by Victor Eijkhout useful and recommend it; see section 1. Its author brings new ideas to the literature about \TeX . But it is not a book for beginners and it does not want to be one. Let’s look briefly at this book to explain the previous sentence.

\TeX by Topic [9] structures the commands of \TeX a bit differently than *The \TeX book* [18]. But more than a dozen chapter headings of [18] and [9] can be mapped to each other as \TeX ’s typographic concepts are handled in both books. However, [9] splits some chapters of [18] into several, and it describes the macro programming topics—registers, conditionals, file input and output, etc.—in their own chapters. It calls itself a “ \TeX nician’s Reference” and as a reference book it contains examples but does not present any exercises.

Philip Taylor reviewed the book for *TUGboat*; he writes [36, p. 185]: “ \TeX by Topic is a reference manual to the \TeX language, arranged as its title suggests by topic. It makes no pretence to being an introduction to \TeX , plunging straight in to the four-level hierarchy (‘eyes’, ‘mouth’, ‘stomach’ and ‘bowels’) of the \TeX processor on page 1.” And on p. 188 he adds: “It seems unlikely that many would choose to learn \TeX solely by a study of \TeX by Topic . . . , but once past the initial learning stage, few would fail to derive benefit from easy access to a copy of \TeX by Topic.”

One of the first control sequences that a reader sees appears on page 6: `\expandafter`. That’s usually not considered to be a command that a beginner can understand. Terms like “macro”, “expansion”, “token”, etc. are used at this early stage too. The symbol ‘`_`’ appears on page 9 in this explanation “`_` Control space. Insert the same amount of space that a space would when `\spacefactor=1000`.” Can a beginner guess what it means?

Focused on \TeX ’s primitives, it does not always explain typographic exceptions in detail. For example, two primitives scale delimiters around math formulas, but might sometimes fail. Page 172 mentions that then `plain` macros can help; [18, p. 149] discusses three examples and shows in two cases the impact on the output.

Udo Wermuth

Of course, like all the books it contains a few errors. An example: On page 91 of [9] it is said that `\ifx` takes prefixes into account but that is corrected on page 121 where the prefix `\global` is no longer mentioned. Let’s cite the reviewer again [36, p. 186]: “This is, however, nit-picking: the technical accuracy is excellent. (I think I found one serious flaw in the whole book, and a few lesser infelicities; for example, on page 70, Victor asserts that `\hfilneg` (`\vfilneg`) is equivalent to `\hskip` (`\vskip`) 0 cm minus 1 fil; I would assert that it is equivalent to `\hskip` (`\vskip`) 0 cm plus -1 fil, which is entirely different.)” (See *The \TeX book*, page 72.) The error also exists in my edition [9]; now on page 67.

It is a pity that the 2014 edition by DANTE contains many errors in formatting, for example, when entering labels and in Appendix A. The errata lists a lot of bugs, but there are more. I cannot help but assume that proofreading by the editor was skipped; the author is blameless for this edition.

A book by involuntary users. Most quotes in this article are by persons who use \TeX for some time. They need it for their daily work. Often their job has to do with computer science, mathematics, physics, etc. Other \TeX users are asked to learn.

Hope Hamilton describes how that happened to her [12, p. 541]: “One of the more creative scientists placed a copy of *The \TeX book* on my desk with the comment ‘See how you like this word processing program!’ I scanned through the book, made a mental note to try to make some sense of it if and when I had the time, and returned to our state-of-the-art stand-alone workstations, Micoms and NBIs. Returning a week later, the scientist inquired if I had started using \TeX in our office production. I stammered: ‘No, not yet.’ I had been too embarrassed to follow up with “Why should I? I can’t understand the book. There’s no one to help me. I’m not a programmer or a computer scientist.’”

Of course, the organization should arrange a course to teach \TeX . But secretaries don’t find it easy to follow a \TeX expert. Robin L. Kubek writes [31, p. 561]: “ \TeX use seems to be largely confined to the technical professionals with degrees in the fields listed above, who have taught themselves with only *The \TeX book* to guide them (and wouldn’t have a clue as to how to teach anyone the program who doesn’t speak at least three programming languages). For those of us who had to look up the word ‘algorithm’ several times in the dictionary and still wouldn’t recognize one even if it asked us to dance, a major dilemma arose when faced with the prospect of learning ‘by the book’.”

Hope Hamilton managed to write a text [12, p. 542]: “I saved that first letter and stapled it to the input page of T_EX commands.” So she saved input and output for reference; something that Angela Barden finds useful too (see section 9). Input/output comparisons are also presented in [2, Chap. 3].

Hamilton continues: “A breakthrough occurred about six months later. I met a secretary, Eileen Boettner, from another NCAR site seven miles away. We discovered that we had reacted to the T_EX challenge in the same way. My new friend had also saved her samples, and we began to discuss our T_EX problems. Each of us, in her own location, had become a resource to other secretaries. Our samples were hopelessly chained to typewriter terms and NBI keyboard procedures. Glue? Boxes? Modes? Those concepts were obviously meant for programmers and hackers. We would have to manage without them.”

The input/output sheets were placed into a report of 320 pages, the Index [12, p. 543]: “Meanwhile, news of the Index began to spread by word of mouth: a typist’s tutorial for T_EX—no formal training nor programming experience necessary.” They justify this approach for themselves. “At this point, many doubts and questions began to give me increasing concern. Was the Index a satisfactory vehicle for teaching T_EX? Did the Index succeed in motivating nonscientific personnel to learn T_EX? Could this Index be used as effective documentation for learning T_EX? My answer, as would be the answer of any experienced teacher, was ‘If it works, use it.’”

Knuth’s exposition, the exercises and their solutions, and the many examples seem to allow readers to do this kind of extraction without developing an understanding for T_EX’s concepts. From my point of view it’s definitely not the best way to utilize T_EX and to get readable and maintainable input as well as a consistent looking output.

Robin L. Kubek developed a different strategy. She starts teaching operators using *The T_EXbook* and tries to overcome their initial problems in studying the book, so that they can help themselves [31, p. 566]: “One word of warning, however; once you have taught a word processing operator the basics of T_EX she may not want to type in WordPerfect, Microsoft Word, Word-11, or any of the myriad assortment of ordinary word processing applications, ever again. ... Furthermore, the more she learns about T_EX, the more she’s going to want to learn.”

12 Final remarks

Of course, *The T_EXbook* does not need an advocate. However, some potential readers might have heard

scary stories of the discussed complaints; this text might help to reevaluate them.

We saw that *The T_EXbook* is a good introduction to plain T_EX if a beginner follows the instructions. We realized that the intermediate gets a path to build up skills. We learned that it is a necessary book for every T_EXnician. We got the feedback that its index tests the patience of macro writers who either expect a programmer reference manual or do not know T_EX’s concepts backward and forward.

Of course, D. E. Knuth is aware that his book does not work for all users. In [29, p. 140] he said in an interview: “I enjoy exposition, but I can write best only for people who think like me. I’m not really the best writer for somebody who just wants to know the basic facts. Take *The T_EXbook*, for example. Some people think it is brilliant and just the right way to explain a system for typesetting, while others just hate it. They want another kind of ‘how to’ book, one that just spells out and defines the features needed to support standard formats.”

A beginner who is willing to invest time to learn plain T_EX using *The T_EXbook* can handle all kind of documents and starts to realize what the word “typography” means. Of course, I suggest that the beginner follows Knuth’s advice and plans to read the book several times—guided by the dangerous-bend sign. Such an educated user might work with `plain` or switch to another format. What was learned can be used in any other T_EX-based program to create high-quality output. Let’s hope you find a book as good as *The T_EXbook* to learn all about the other program or format if you decide to switch.

References

- [1] Paul Abrahams, “Response to Victor Eijkhout”, *TUGboat* **12**:2 (1991), 302–303.
tug.org/TUGboat/tb12-2/tb32letters.pdf
- [2] Paul W. Abrahams, Kathryn A. Hargreaves, Karl Berry, *T_EX for the Impatient*, free edition, 2003.
The latest edition (2020) and translations can be found under: ctan.org/pkg/impatient
- [3] Angela Barden, “Some T_EX manuals”, *TUGboat* **12**:1 (1991), 166–170.
tug.org/TUGboat/tb12-1/tb31barden.pdf
- [4] David Bausum, *T_EX Reference Manual*, Norwell, Massachusetts: Kluwer Academic Publishers, 2002.
tug.org/utilities/plain/cseq.html
- [5] Andrew Donald Booth, “Review of *Latex: a document preparation system*”, (1990).
computingreviews.com/review/review_review.cfm?review_id=123336
- [6] William R. Cheswick, “A Permuted Index for T_EX and L^AT_EX”, *TUGboat* **10**:4 (1989), 707–713.
tug.org/TUGboat/tb10-4/tb26cheswick.pdf

- [7] Malcolm Clark, *A plain T_EX Primer*, Oxford University Press, 1992.
- [8] Michael Doob, “Of the Computer Scientist, by the Computer Scientist, for the Computer Scientist”, *TUGboat* **10**:4 (1989), 533–539. tug.org/TUGboat/tb10-4/tb26doob.pdf
- [9] Victor Eijkhout, *T_EX by Topic* — A T_EXnician’s Reference, DANTE and Lehmanns, Berlin, 2014. Errata: dante.de/wp-content/uploads/2019/07/TexByTopic_Errata.pdf
See also: eijkhout.net/tex/tex-by-topic.html
Latest version: github.com/VictorEijkhout/tex-by-topic/blob/main/TeXbyTopic.pdf
- [10] Gérard G. Emch, Arnold K. Pizer, “Letter to the editors”, *TUGboat* **1**:1 (1980), 22–23. tug.org/TUGboat/tb01-1/tb01emch.pdf
- [11] Richard Furuta, “Review of *T_EX and METAFONT, New Directions in Typesetting* by D. E. Knuth”, *Proceedings of the IEEE* **71**:9 (1983), 1117–1118. ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1457003
- [12] Hope Hamilton, “Mastering T_EX with Templates”, *TUGboat* **10**:4 (1989), 541–548. tug.org/TUGboat/tb10-4/tb26hamilton.pdf
- [13] Yannis Haralambous, “T_EX as a path”, *TUGboat* **39**:1 (2018), 8–15. tug.org/TUGboat/tb39-1/tb121haralambous-knuth80.pdf
- [14] Amy Hendrickson, “Review of *T_EX by Topic* by Victor Eijkhout”, *MAPS* **8** (1992), 137–138. ntg.nl/maps/08/27.pdf
- [15] Alan Hoenig, *T_EX Unbound* — L^AT_EX & T_EX Strategies for Fonts, Graphics, & More, Oxford University Press, 1998.
- [16] Donald E. Knuth, *Tau Epsilon Chi, a system for technical text*, Stanford Computer Science Report *STAN-CS-78-675*, November 1978.
- [17] Donald E. Knuth, *T_EX and METAFONT: New Directions in Typesetting*, Providence, Rhode Island: American Mathematical Society, and Bedford, Massachusetts: Digital Press, 1979. bitsavers.org/pdf/dec/_Books/_Digital_Press/Knuth_TeX_and_Metafont_New_Directions_in_Typesetting_1979.pdf
- [18] Donald E. Knuth, *The T_EXbook*, Volume A of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1984. www-cs-faculty.stanford.edu/~knuth/abcde.html
- [19] Donald E. Knuth, *T_EX: The Program*, Volume B of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1986.
- [20] Donald E. Knuth “Notes on the Errors of T_EX”, *TUGboat* **10**:4 (1989), 529–531. tug.org/TUGboat/tb10-4/tb26knut.pdf
- [21] Donald E. Knuth, “Arthur Lee Samuel, 1901–1990”, *TUGboat* **11**:4 (1990), 497–498. tug.org/TUGboat/tb11-4/tb30knut-samuel.pdf
- [22] Donald E. Knuth, “The Future of T_EX and METAFONT”, *TUGboat* **11**:4 (1990), 489. tug.org/TUGboat/tb11-4/tb30knut.pdf
- [23] Donald E. Knuth, *Literate Programming*, CSLI Lecture Notes No. 27, 1992.
- [24] Donald E. Knuth, *Selected Papers on Computer Science*, CSLI Lecture Notes No. 59, 1996.
- [25] Donald E. Knuth, “Χαιρετισμός του καθηγητή Knuth”, *Eutypon* **1** (1998), 2. eutypon.gr/eutypon/pdf/e1998-01/e01-a00.pdf
- [26] Donald E. Knuth, *Digital Typography*, CSLI Lecture Notes No. 78, 1999.
- [27] Donald E. Knuth, *Selected Papers on Design of Algorithms*, CSLI Lecture Notes No. 191, 2010.
- [28] Donald E. Knuth, *Selected Papers on Fun & Games*, CSLI Lecture Notes No. 192, 2011.
- [29] Donald E. Knuth, *Companion to the Papers of Donald Knuth*, CSLI Lecture Notes No. 202, 2011.
- [30] Donald E. Knuth, “The T_EX tuneup of 2014”, *TUGboat* **35**:1 (2014), 5–8. tug.org/TUGboat/tb35-1/tb109knut.pdf
See also the latest tune-up of 2021: tug.org/TUGboat/tb42-1/tb130knuth-tuneup21.pdf
- [31] Robin L. Kubek, “T_EX for the Word Processing Operator”, *TUGboat* **10**:4 (1989), 561–566. tug.org/TUGboat/tb10-4/tb26kubek.pdf
- [32] Arthur L. Samuel, *First Grade T_EX* — A Beginner’s T_EX Manual, Stanford Computer Science Report *STAN-CS-83-985*, November 1983. tug.org/texniques/tn11/firstgrade-all.pdf
- [33] Norbert Schwarz, *Einführung in T_EX*, Rechenzentrums-Schriften, Bd. 5, Rechenzentrum der Ruhr-Universität Bochum, 3rd ed., January 1987.
- [34] Norbert Schwarz, *Einführung in T_EX*, 2nd ed., Bonn, Addison-Wesley (Deutschland), 1988. Third edition as PDF: ruhr-uni-bochum.de/TeX/einfuehrung-in-tex.pdf
- [35] Peter Seibel, *Coders at work* — Reflections on the craft of programming, Apress, 2009.
- [36] Philip Taylor, “Book review: *T_EX by Topic*”, *TUGboat* **13**:2 (1992), 185–188. tug.org/TUGboat/tb13-2/tb35resources.pdf
- [37] Boris Veytsman, answer to the question “Learning T_EX via *The T_EXbook*”, *T_EX – L^AT_EX.SE*, 2012. tex.stackexchange.com/questions/51695/learning-tex-via-the-texbook/51702#51702
- [38] *Web of Stories: Donald E. Knuth*
Part 60: Writing a users’ manual for T_EX. webofstories.com/play/donald.knuth/60
Part 7: My sense of humor. webofstories.com/play/donald.knuth/7

◇ Udo Wermuth
Dietzenbach, Germany
[u dot wermuth \(at\) icloud dot com](mailto:u dot wermuth (at) icloud dot com)