

第1章

TeX, LaTeXとその仲間



TeX とその仲間 (pdfTeX, XeTeX, LuaTeX, pTeX, upTeX) と LaTeX との関係について説明します。

1.1 TeX, LaTeXって何?

TeX /téék/ **[名]** 『コンピュータ』テフ, テック
《テキストベースの組み版システム; 数式の処理を得意とする》。
—— 『ジーニアス英和大辞典』(大修館書店, 2001年)

TeX は、^{くみはん}組版ソフトです。

組版 (typesetting) は印刷用語で、活字を組んで^{はん}版 (印刷用の板) を作ることを意味します。TeX は、コンピュータでテキストと図版をうまく配置して、版にあたるもの (PDF または PostScript ファイル) を出力する (タイプセットする) ためのソフトです^{*1}。

TeX には次のような特徴があります。

- TeX はオープンソースソフトですので、無料で入手でき、自由に中身を調べたり改良したりできます。商用利用も自由にできます。
- TeX は、Windows でも Mac や Linux などの UNIX 系 OS でも、まったく同じ動作をします。つまり、入力が同じなら、原理的には、まったく同じ出力が得られます。
- TeX への入力はテキスト形式なので、普通のテキストエディタで読み書きでき、再利用・データベース化が容易です。
- 自動ハイフネーション、ペアカーニング^{*2}、リガチャ^{*2}、孤立行 (ウィドウまたはオーファン) 処理^{*2} など、高度な組版技術が組み込まれています。
- 特に数式の組版については定評があり、数式をテキスト形式で表す事実上の標準となっています。

^{*1} 今ではPDFでの出力が普通になりましたが、過去には写研の写植機を含む個々のプリンタ等のTeX用ドライバが用意されていました。拙著『C言語による最新アルゴリズム事典』の最初の版 (1991年) はTeXから写研の写植機で出力しました。

^{*2} **ペアカーニング**: AVやToなど相補的な形の文字を食い込ませる処理。

リガチャ: fi, fl, ffi, ffl などのような合字 (対応フォントだけ)。

孤立行処理: 段落の最初の行だけ、あるいは最後の行だけが別ページになることを抑制する処理。

1.2 TeXの読み方・書き方

TeXの作者^{クヌース} Knuth 先生 (4 ページのコラム参照) によれば, TeX はギリシャ語から命名したもので, 最後の X は, 口の奥で発音する無声の「ハ」に近い音だそうですが, 英語でこれに一番近い音は /k/ なので, 「テック」と読む人が多いようです (ドイツ語では「テッヒ」が多いようです)。

日本では, 特に大学関係者の間では, 昔から「テフ」と呼びならわされていますが, 英語圏で TeX を覚えた人や出版関係者の間では「テック」という発音が広く行われています。

TeX は, ご覧のように E を少し下げて, 字間を詰めて書きます。このような文字の上げ下げや詰めは TeX が得意とするところですが, これができない場合は TeX と表記することになっています (TEX や Tex とは書かない約束ですが, なかなか守られていません)。

1.3 L^AT_EXって何?

L^AT_EX は DEC (現 HP) のコンピュータ科学者^{レスリー ランポルト} Leslie Lamport^{※3} によって機能強化された TeX です。もともとの TeX と同様, オープンソースソフトとして配布されています。

L^AT_EX は日本ではラテックまたはラテフと読まれます。英語圏では^{レイ}レイテックと読む人が多いようです。

参考 Web で“latex”を検索すると, latex (乳液, ラテックス) 関係のページがたくさん見つかってしまいます。こちらは英語読みでは^{レイ}レイテックスです (アクセントの位置を圏点付きの太字で示しました)。

最初の L^AT_EX は 1980 年代に作られましたが, 1993 年には^{ツニー}L^AT_EX 2_ε という新しい L^AT_EX ができ, 現在では L^AT_EX といえば L^AT_EX 2_ε を指すようになりました (古い L^AT_EX は L^AT_EX 2.09 と呼ばれます)。本書でも L^AT_EX 2_ε を以下では単に L^AT_EX と書くことにします。

L^AT_EX は, ご覧のように A を小さく上付きにして書きます。字の上げ下げができないなら LaTeX と書くことになっています。同様に, L^AT_EX 2_ε と書けないなら LaTeX2e と書きます^{※4}。

L^AT_EX の特徴は, 文書の論理的な構造と視覚的なレイアウトとを分けて考えることができることです。

例えば「はじめに」という^{セクション}節の見出しがあれば, 文書ファイルには

```
\section{はじめに}
```

※3 Lamportはその後2001年にMicrosoft Researchに移籍しました。2013年にはチューリング賞を受賞しています。

※4 日本では全角のε (エプシロン) を使ったLaTeX2ε という表記もよく見かけますが, Unicode時代なら最後の文字はU+1D700 (MATHEMATICAL ITALIC SMALL EPSILON) にするのがいいかもしれません。

のように書いておきます。この `\section{...}` という命令が、紙面上のデザイン、例えば「14 ポイントのゴシック体で左寄せ、前後のアキはそれぞれ何ミリを標準とし、何ミリ以内なら伸ばしてよい……」というレイアウトに対応するといったことは、様式・判型ごとに別ファイル（クラスファイル、スタイルファイル）に記述されています。標準のクラスファイルのデザインが気に入らないなら、自由に変更できます。クラスファイルだけ変更すれば、同じ文書ファイルでも違ったレイアウトで出力できます。

仮に文書ファイルに「ここは 14 ポイントのゴシック体で 3 行どり中央に……」などと書き込んでしまったのでは、あとで組み方を変更しようとする、原稿全体に手を入れなければなりません。下手をすると、節ごとに見出しの体裁が違ってしまふことにもなりかねません。文書の再利用も難しくなります^{※5}。

さらに、L^AT_EX は章・節・図・表・数式などの番号を自動的に付けてくれますし、参照箇所には番号やページを自動挿入できます。目次・索引・引用文献の処理まで自動的にしてくれます。また、^{はしら}柱（本書ではページ上部にあり、左ページには章の名前、右ページには節の名前を入れてあります）も自動的に作ってくれます。

このような便利な機能のため、L^AT_EX 利用者が飛躍的に増え、T_EX を使っているといっても実際には L^AT_EX であることが多くなりました。

L^AT_EX は、T_EX のプログラミング機能（マクロ機能）を使って作られたものです。一方、T_EX 本体（マクロと区別するためにエンジンと呼ぶことがあります）も改良され、特に日本では日本語の扱いに優れた p^{ビ-}T_EX というエンジンが広く使われるようになりました（より現代的なエンジンについては後述します）。p_TE_X 用にマクロを修正した L^AT_EX が pL^AT_EX (pL^AT_EX 2_ε) です。

L^AT_EX は理系の論文や本の製作に広く使われています。多くの論文誌や^{アーカイブ}arXiv^{※6} のようなプレプリントサーバは L^AT_EX での論文投稿を推奨^{※7} していますし、理系の出版社は多くの本を L^AT_EX で製作しています。一例を挙げれば、『岩波数学辞典』の最新版（第 4 版）はすべて L^AT_EX (pL^AT_EX 2_ε) で作られています。Wikipedia も数式は L^AT_EX 形式で書きます（Wikipedia サーバ上の L^AT_EX で SVG 画像にしています）。

※5 L^AT_EX による文書の構造化は HTML と同じ考え方だと気づかれたかもしれません。HTML は SGML に基づいて作られましたが、L^AT_EX の影響も受けています。L^AT_EX は Scribe というシステムの影響を受けていますが、Scribe は GML (SGML の元) と同じころ作られました。

※6 <https://arxiv.org/>

※7 “the best choice is TeX/LaTeX” (<https://arxiv.org/help/submit>)
arXiv 内で L^AT_EX 処理ができるようになっていました。2020 年 10 月 1 日付で T_EX Live 2016 から 2020 に更新されました。

◆ TeXは誰が作ったの?

COLUMN

クヌース……計算機科学の分野でもっとも偉大な学者の一人
——『岩波情報科学辞典』(岩波書店, 1990年)

TeXを作ったのはスタンフォード大学の Donald E. Knuth^{クヌース} 教授 (1938~) です (現在は退職されています)。Knuth 先生は数学者・コンピュータ科学者で, 1974年にチューリング賞 (コンピュータ科学で最も権威のある賞), 1996年に京都賞を受賞しています。主著 *The Art of Computer Programming* シリーズはコンピュータ科学の聖典ともいえるべきものです (邦訳がアスキードワゴから出ています)。これ以外にもたくさんの著書があります。数学小説『超現実数』(好田順治訳, 海鳴社, 1978) という型破りの数学書も著しておられます。

Knuth 先生の主著 *The Art of Computer Programming* は, 予定では全7巻ですが, 第1巻は1968年, 第2巻は1969年, 第3巻は1973年に出版され, 第1, 2巻の第3版が1997年に, 第3巻の第2版が1998年に, 第4A巻が2011年に出版されました。こまでは KADOKAWA から邦訳が出ています。その後, 第4B巻の2/3ほどが分冊5・6として仮出版されたところです。

この第1巻の第2版まではすべて職人が活字を組む活版印刷^{かっぱん}で作られました。しかし, 活字を組む職人の確保が次第に難しくなり, 第2巻の第2版はいったんコンピュータで組版されました (1976年)。ところが, この仕上がりは活版印刷に比べてかなり見劣りするものでした。

がっかりした Knuth 先生は, この出版を見合わせ, 活版印刷に劣らない美しい組版のできるコンピュータ・ソフトウェア TeX を作る決心をされたのです。

Knuth 先生はたいへんな完全主義者で, 古今の組版技術を研究し, その最も優れた部分を TeX に取り入れました。また, 文字をデザインするためのソフト METAFONT (メタフォント) を作り, Computer Modern というフォントをご自分でデザインされました。

こうしてできあがった TeX とフォントを使って *The Art of Computer Programming* 第2巻の第2版が組み上がったのは4年後の1980年 (実際の出版は1981年) です。

このあとも Knuth 先生は TeX やフォントの改良に余念がなく, 1982年には現在の TeX とほぼ同じものを完成させ, これを使って1984年の *The TeXbook* (邦訳がアスキーから出ていました), 1986年の *TeX: The Program* に始まる *Computers & Typesetting* シリーズ全5巻を書き上げました。

1982年以降は, Knuth 先生は TeX の拡張より安定化に力を注がれたので, 1989年に入力が7ビットから8ビットに拡張されたことを除き, 基本的な仕様の変更はほとんどありません。そして, TeX 第3.1版 (1990年9月) の時点で次のような終決宣言を出されました。

- もうこれ以上 TeX は拡張しない。
- もし著しい不具合があれば修正して第3.14版, 第3.141版, 第3.1415版, ... と番号を進めていき, 自分の死と同時に第 π 版とする。それ以後はどんな不具合があっても誰も手をつけてはならない。
- TeX に関することはすべて文書化したので, このノウハウを生かして新たにソフトを作ることは自由である。

TeX プロジェクト開始から40年近くの歳月がたち, 今や TeX はコンピュータ科学の大先生の作品と呼ぶにふさわしい完成度の高いソフトになりました。現在の Knuth 先生は TeX を使っての著作に専念しておられます。

1.4 TeX, L^AT_EXの処理方式

一般のワープロソフトと異なり、TeX, L^AT_EX は高度な最適化をしているので、段落の最後に1文字追加するだけで段落の最初の改行位置が変わることもあります。このような処理を、キーボードから1文字入力するごとに行うのは、かなりの計算パワーを必要とします。

そのため、TeX, L^AT_EX では、キーを打つたびに画面上の印刷結果のイメージを更新する方式^{※8}ではなく、一括して全体を処理するバッチ処理を採用しています。

原稿は自分の使いなれたソフト（テキストエディタ）で書いて、テキストファイルとして保存しておきます。これをあとでL^AT_EXで一括処理します。

例えばL^AT_EXで

```
\documentclass{jsarticle}
\begin{document}
```

これはサンプルの文書です。
 テキストファイル中では、
 どこで改行してもかまいません。
 印刷結果の改行の位置は勝手に決めてくれます。

段落の切れ目には空の行を入れておきます。

```
\end{document}
```

のようなテキストファイルを入力すると、

これはサンプルの文書です。テキストファイル中では、どこで改行してもかまいません。印刷結果の改行の位置は勝手に決めてくれます。
 段落の切れ目には空の行を入れておきます。

のように出力されます（\で始まる行はL^AT_EXの命令です）。

テキストファイルを使うことの利点は、たくさんあります。

- 文書入力自分の慣れているソフトで行うほうが楽です。どんな文書入力ソフトでもテキストファイルで保存できますので、L^AT_EXは入力ソフトを選びません。
- テキストファイルはコンピュータの機種に依存しません。どんなコンピュータやスマホでも、テキストファイルなら安全にやりとりできます。

※8 画面表示と印刷イメージが同じ(What You See Is What You Get)という意味で、WYSIWYG(ウィジウィグ)方式と呼ぶことがあります。

※左の入力例の\ (逆斜線, バックスラッシュ)はWindowsの和文フォントでは¥と表示されます。本書旧版では¥で統一していましたが、Windowsでも\と表示される場合が増えたため、第7版以降ではほぼすべての箇所でも\を使っています。

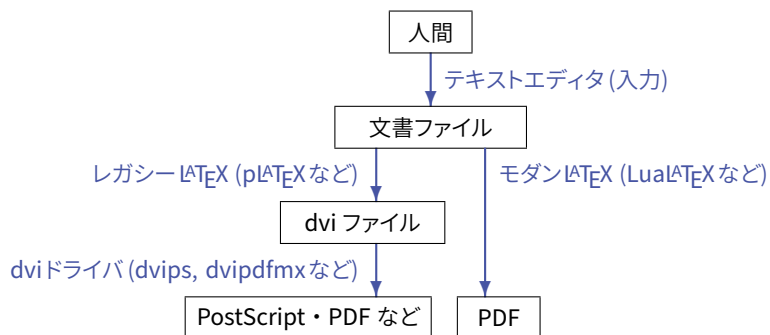
- テキストファイルの変換は簡単です。そのため、データベース出力や Web フォーム入力, XML 文書などから L^ATeX に変換して組版するといったことがよく行われています。
- テキストファイルで文書を用意するほうが, コンピュータのパワーユーザーの心理に合っているのかもしれませんが。数式を考えなければ Adobe InDesign のようなリアルタイムで L^ATeX 同様の処理をするレイアウトソフトがありますし, 数式についても WYSIWYG な数式エディタがあります。しかしこれらは今のところ L^ATeX を不要にするに至っていません。

1.5 TeX, L^ATeXの処理の流れ

もともとの TeX, L^ATeX は, 組版結果を ^{デーヴィーアイ}dvi ファイルという中間ファイルに書き出します^{※9}。dvi は device independent (装置に依存しない) という英語の略です。この dvi ファイルを読み込んでパソコンの画面, 各種プリンタ・写植機, および PostScript・PDF・SVG などのファイルとして出力するための専用ソフト (dvi ドライバ, dvi ウェア) が, 出力装置や出力ファイル形式ごとに用意されています。特に画面出力用の dvi ドライバのことを dvi ビューアともいいます。

このようなレガシーな TeX, L^ATeX に対して, モダンな TeX, L^ATeX では, dvi ファイルを介さず, いきなり PDF ファイルを出力します。文書ファイルも Unicode になり, 扱える文字数の上限が事実上なくなりました。

※9 dviはDVIとも書きま
す。Digital Visual Interfaceの
DVIとは無関係です。



1.6 TeX, LaTeXと日本語

日本語の文字数は非常に多く、多数のフォントに分割すればオリジナルの TeX でも扱えないことはありませんが、厄介です。

本格的に TeX を日本語化する試みはいくつかありましたが、今日広く使われているのは、在りし日の榊アスキーが開発した pTeX^{※10} およびそれを Unicode 対応にした upTeX (田中琢爾さん作) です。これらを LaTeX 化したものが pLaTeX, upLaTeX です。

日本語なら文字はみな同じ幅だから単純に組んでいけばよいかという、そうはいきません。次のような処理が必要です。

- 句読点、終わり括弧 (閉じ括弧) 類、中黒 (・)、繰返し (々々)、感嘆符 (!)、疑問符 (?) が行頭にこないようにする必要があります (行頭禁則処理)。促音文字 (っ)、拗音文字 (ゃゅょ)、長音記号 (ー、音引き) などなるべく行頭にきてほしくありません。
- 同様に、始め括弧 (開き括弧) 類が行末にこないようにする必要があります (行末禁則処理)。
- 「括弧類」「句読点」が「行頭」、「行末」にきたときや、「括弧類」「句読点」が連なったとき、空白が空きすぎて見えるので、詰める必要があります (この段落は例示のためにわざと括弧類を多用しました)。
- 段落の最後の行が 1 文字と句読点になるのは、なるべく避けたいところです。←なるべくここで終わってほしくありません (文字ウィドウ処理)。

これらの条件を満たすためには、字の間隔を微調整しなければなりません (詰め処理、延ばし処理)。しかし、調整しすぎると字の間隔が揃わず、かえって見苦しくなります。例えば、拗促音文字 (ゃゅよっ) はなるべく行頭にこないほうがよいし、文字ウィドウもなるべく避けたいのですが、あまり厳密にこれらのルールをあてはめるとかえって不自然になることがあります。そこで pTeX は、どの文字が行頭にくると何点減点、行末にくると何点減点、文字ウィドウは何点減点、字の間隔がどれだけ伸びると何点減点という具合に点数を計算し、減点の合計が最小になるように組みます。点数の配分は調節できます。

pTeX, pLaTeX の日本語組版がたいへん優れていたもので、広く使われるようになった一方で、海外で普及しつつあったモダン LaTeX (XeLaTeX や LuaLaTeX) でも日本語が扱えるようになりました。特に LuaLaTeX で日本語を扱う仕組み (LuaTeX-japan) の進歩のおかげで、今や pLaTeX 以上の品質の組版が可能になりました。本書は第 7 版までは pLaTeX で組んでいましたが、この第 8 版は全編 LuaLaTeX で組みました。

※10 pTeXの昔の版は「アスキー日本語TeX」と呼ばれていました。なお、pTeX以外にNTT JTeXもかつては広く使われていました。

1.7 TeX, L^ATeXのライセンス

本書付録 DVD-ROM に収めた TeX 関連のソフトは、すべてオープンソースのライセンスで配布されており、商用利用も含めて自由に使えます^{※11}。詳しくは各ソフトのマニュアルをご覧ください。ターミナルに「texdoc ソフト名」と打ち込めばマニュアルや関連ドキュメントが表示されます（26 ページ参照）。

オリジナルの TeX は、付加価値を付けたものを有償で販売することも自由です。ただし、TeX との完全な互換性を持たないものは TeX と名乗ってはいけません^{※12}。米国での TeX の商標は American Mathematical Society（米国数学会）が登録していますが、これは無関係な人に商標登録されることを防ぐため、TeX を使う際に「TeX は……の商標です」などと断る必要はありません。

L^ATeX は LPPL（L^ATeX Project Public License）^{※13} に従い、ファイル名さえ変えれば改変したものの再配布も自由です。

pL^ATeX 等については、在りし日の榎アスキーが開発したものですが、日本語 TeX 開発コミュニティ^{※14} による「コミュニティ版」に移行しています。これらは（修正）BSD ライセンスに従っており、オリジナルの著作権表示などを残す限り改変・再配布は自由です。

※11 詳しくはコラム「オープンソースライセンス」をご覧ください。

※12 例えば pTeX は TeX と完全に互換ではありませんので、TeX とは名乗っていません。

※13 <https://www.latex-project.org/lppl/>

※14 <https://texjp.org>

◆ オープンソースライセンス

COLUMN

TeX 関係のソフトウェアの多くは、オープンソースのライセンスで配布されています。「オープンソース」は、ソースコード（ソフトウェアの設計図にあたるもの）へのアクセス、改変、再配布が自由にできることを意味します。The Open Source Initiative（<https://opensource.org/>）の定める「The Open Source Definition」（オープンソースの定義）によれば、商用利用など利用分野についての制限や、特定の人あるいはグループについての制限は認められていません。TeX のライセンス、L^ATeX の LPPL、BSD ライセンス、GNU GPL は、このオープンソースの定義に合致します。

本書付録 DVD-ROM に収録した TeX Live などのソフトウェアはすべてオープンソースのライセンスで配布されています。ただ、TeX Live に含まれる多数のファイルの中には、微妙なものもあります。具体的には、multicol パッケージ（ファイル名 multicol.sty）と、これに基づく adjmulticol パッケージ（ファイル名 adjmulticol.sty）とは、ファイルの先頭に“Moral obligation”と題して「商用利用では有用さに応じて寄付を求める（有用でないとすれば払わなくてよい）」といった内容が書き込まれています。この文言は、捉え方によっては上述のオープンソースの定義とは相容れないものです。

一方、Aladdin Free Public License に従うものは、配布手数料も取ってはならないことになっています。この類のものは「nonfree」と分類され、通常の TeX Live の配布物には含まれていません（本書付録 DVD-ROM にも含まれていません）。これらは必要に応じて別途ダウンロードすることになります（367 ページ参照）。

1.8 TeXディストリビューション

もともとの TeX は Pascal をベースとした Knuth の WEB^{※15} という「文芸的プログラミング」ツールで作成されていますが、UNIX 上では通常は C 言語に変換してからコンパイルしています。これが現在の多くの TeX の実装の起源である Web2c の由来です。

この Web2c をベースに Thomas Esser が集大成した teTeX という TeX ディストリビューション^{※16} が広く使われるようになり、日本では土村展之さんがこれに基づく ptetex を配布されていました。

一方、Windows では角藤亮^{あきら}さんの W32TeX というディストリビューションが広く使われるようになります。

その後、TeX Live という超巨大な集大成が広く使われるようになり、土村さんもこれに基づく ptexlive を開発されました。これが現在ではすべて TeX Live 本体に取り込まれました。本書 DVD-ROM に収録したものは Windows 用も Mac 用もすべて TeX Live に基づくものです。

1.9 これからのTeX

最初の TeX が作られたのは 1978 年です。これだけ長い間安定して使われているソフトはほかに例を見ません。TeX はコンピュータ組版の歴史における一つの不動点^{フィクストポイント}と言えるでしょう。

しかし、既存の TeX に満足しては進歩がありません。今日に至るまで、TeX 本体（エンジン）およびマクロに、いろいろな拡張が行われています。エンジンの拡張としては次のようなものがあります。

- ϵ -TeX (e-TeX)^{※17} は、TeX の種々のレジスタ（変数）の個数を拡張（256 個 → 32768 個）したほか、右から左に組む機能などを追加したものです。現在では L^ATeX そのものが ϵ -TeX 拡張を仮定しています。
- TeX を日本語化した pTeX も、北川弘典さんにより ϵ -TeX 拡張されました (ϵ -pTeX (e-pTeX))。
- 田中琢爾さんの upTeX (upTeX, 読み方はユーピー TeX またはユブ TeX) は、(ϵ -)pTeX の内部を Unicode 化したものです。
- Hàn Thế Thành さん作の pdfTeX (pdfTeX) は TeX の出力形式を dvi ではなく PDF にし、 ϵ -TeX 拡張に加えて、microtypography という高度な組版アルゴリズムを組み込んだものです。現在ではオリジナルの TeX を置き換えて広く使われています。

※15 World Wide Web の Web とは無関係です。Knuth の WEB のほうが古いものです。

※16 ディストリビューションとは、配布用にパッケージされたソフトウェア群のことです。

※17 TeX 関係の名前の多くは、「TeX」のようなロゴと、「TeX」のような普通の文字だけで表した形があります。ここでは、後者を括弧に入れて示しています。

- Jonathan Kew さん作の X^{スリー}TeX (XeTeX) はもともと Mac 上で動作し、システムの OpenType フォントをそのまま使い、PDF を出力する TeX ですが、Windows や Linux にも移植されています^{※18}。入力ファイルは Unicode で、IVS (異体字セレクタ) にも対応し、日中韓の文字も自由に使えます。
- Taco Hoekwater さん, Hartmut Henkel さん, Hans Hagen さん作の Lua^{ルア}TeX は、pdfTeX に軽量スクリプト言語 Lua を組み込んだものです。IVS を含めて Unicode に対応し、システムの OpenType フォントに対応しています。pdfTeX の後継として今最も注目されているものです。LuaTeX-ja プロジェクトの成果と組み合わせれば、pTeX 以上の自由度で和文組版が可能になります。
- Clerk Ma (马起园) さんによる pTeX-ng^{※19} は、Web2c ベースではなく C 言語で開発された Y&Y TeX から出発して、pTeX, upTeX の機能を取り込んだもので、PDF を直接出力します。まだまだ開発中のものです。

※18 Windowsへの移植は角藤さんによって行われています。

※19 <https://github.com/clerkma/ptex-ng>

一方、L^ATeX も、L^ATeX 2_ε ができてもう 20 年経ちます。現在は L^ATeX 3 の開発が進行しており、すでに L^ATeX 3 の成果物が L^ATeX に組み込まれつつあります。今後 L^ATeX 内部は L^ATeX 3 の実装に次第に置き換えられ、機能的に進化する代わりに、「お行儀の良くない」L^ATeX 2_ε 文書は処理できなくなる可能性もあります。

また、L^ATeX とまったく異なるマクロパッケージ Con^{コン}TeXt (Hans Hagen さん作) も特にヨーロッパでユーザー層を広げています。現在の ConTeXt は LuaTeX 上で動きます。

さらに、TeX を置き換えるべく開発中の組版システムとして、諏訪敬之さんの SATySF^{サティスファイ}I^{すわたかし}があります^{※20}。これは、静的型付けにより可読性とエラー報告の向上を目指したものです。

※20 <https://github.com/gfngfn/SATySF/I>

また、Web 技術から発展した CSS 組版の流れでは、村上真雄さんたちが開発されている Vivliostyle^{しんぼう}があります^{※21}。

※21 <https://vivliostyle.org>

一方、L^ATeX の数式表記法は事実上の標準となり、Microsoft の Office や Apple の Pages, Numbers, Keynote, iBooks Author で L^ATeX 表記の数式入力が可能になっています。また、Web 上で L^ATeX とほぼ互換の数式組版機能を JavaScript, CSS, Web フォントで実現したものとして、MathJax や KaTeX (K^カTeX) が広く使われています。

Web ベースの帳票印刷システムでも、サーバ側で L^ATeX 経由で PDF を生成し、クライアント側で印刷するといった用途で L^ATeX を使うことがあります。

最近では、Markdown という非常に簡単な形式で文書を記述し、必要に応じて L^ATeX, HTML などに変換しようという流れがあります。多様な形式間を相互変換する pandoc などのツールと組み合わせると、文書処理の自動化に貢献しています。