

مقدمه ای نه خندان کوتاه بر L^AT_EX 2_ε

یا $L^A T_E X 2_\epsilon$ در ۱۲۶ دقیقه

by Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

Version 5.01, April 6, 2011

Translator: Mehdi Omidali

مترجم: مهدی امیدعلی

Copyright ©1995-2011 Tobias Oetiker and Contributors. All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

حق مؤلف ۱۹۹۵-۲۰۱۱ © توبیاس اوتیکر و دیگران. تمام حقوق محفوظ است.
این نوشتار آزاد است، تحت اجازه‌نامه عمومی گنو (نسخه ۲ یا نسخه‌های جدیدتر)، می‌توانید آن را پخش و/یا تغییر دهید.
این نوشتار به این امید تهیه شده است که مفید واقع شود ولی بدون هیچ‌گونه ضمانتی؛ حتی بدون این ضمانت که مناسب کار خاصی باشد. برای اطلاعات بیشتر به اجازه‌نامه عمومی گنو مراجعه کنید.
به همراه این نوشتار، باید یک نسخه از اجازه‌نامه عمومی گنو را دریافت کرده باشید؛ اگر این‌گونه نیست، با آدرس زیر تماس حاصل فرمایید:

Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

پیشگفتار مترجم

امروزه بیشتر مجله‌های علمی و پژوهشی از نویسندگان خود انتظار دارند مقاله خود را با لاتک تهیه کنند. مشهور است که کسانی که اولین بار با لاتک متنی را آماده می‌کنند، در میانه کار می‌گویند که دیگر از این نرم‌افزار استفاده نخواهند کرد؛ اما بعد از پایان کار به خود می‌گویند دیگر به هیچ عنوان به سراغ نرم‌افزارهایی مانند word نخواهند رفت. دلیل این کار روشن است؛ لاتک برای هر منظور، فرمانی دارد که باید این فرمان‌ها را بدانید تا بتوانید به طور بهینه از آن استفاده کنید. اگر در ابتدا زمان کافی برای یادگیری این فرمان‌ها صرف نکنید، در آینده چندین برابر این زمان را برای رفع مشکلات نوشتار خود باید صرف کنید. این کتاب به این امید ترجمه شده است که بتواند به شما در یادگیری فرمان‌های لاتک کمک کند.

به تازگی نرم‌افزار زیتک به بازار ارائه شده است که توانایی استفاده از قلم‌های مختلف را فراهم کرده است. زیلاتک، که همان لاتک بر پایه زیتک است، تمامی امکانات قوی لاتک را برای تهیه هر نوع مستندی، از جمله مستندات فارسی، ارائه کرده است. به همین منظور بسته‌ای با نام X_YPersian توسط آقای وفا خلیقی تهیه شده است که این ترجمه با استفاده از این بسته و به منظور بررسی سازگاری آن تهیه شده است. آقای وفا خلیقی دانشجوی دکتری ریاضی دانشگاه سیدنی هستند که واقعاً با تلاش غیرقابل توصیف کار تهیه این بسته را به عهده گرفتند و بدون چشم‌داشتی این کار بزرگ را انجام دادند. وظیفه خود می‌دانم که از طرف جامعه علمی کشور از ایشان کمال تشکر را داشته باشم و با افتخار این ترجمه ناچیز را به خود ایشان تقدیم کنم.

همچنین لازم است از زحمات آقای مصطفی واحدی به خاطر شروع اولین قدم‌های تهیه بسته‌ای برای نگارش فارسی و همچنین مبدل فارسی‌تک به یونیکد (به سبک مناسب زی‌پرشین) و همچنین ایجاد گروه فارسی لاتک گوگل^۱ تشکر نمایم. برای دریافت کمک و انتقال نظرات و پیشنهادات خود و همچنین دریافت آخرین اطلاعات می‌توانید به این گروه ملحق شوید. امکانات استفاده از BibTex توسط آقای محمود امین طوسی فراهم گردیده است که از ایشان سپاسگذاری می‌کنم. از آقای سید رضی علوی‌زاده برای تهیه افزونه نگارش فارسی به ویرایشگر Texmaker و از آقای امیرمسعود پورموسی برای تلاش بسیار ایشان در آماده‌سازی ویکی زی‌پرشین^۲ تشکر می‌کنم.

مهدی امیدعلی <mehdioa@gmail.com>

^۱<http://groups.google.com/group/farsilatex?hl=fa>

^۲<http://fa.parsilatex.wikia.com>

پیشگفتار

\LaTeX [۱] یک سیستم حروف‌چینی است که برای تولید نوشتار با کیفیت عالی علمی و ریاضی بسیار مناسب است. این سیستم همچنین برای تولید انواع دیگر نوشتار، از یک نامه ساده تا کتاب‌های کامل، مناسب است. \LaTeX از \TeX [۲] به عنوان موتور حروف‌چین استفاده می‌کند. این مقدمه کوتاه به معرفی لاتک می‌پردازد و برای بسیاری از کاربردهای آن کافی است. برای مشاهده شرح کاملی از سیستم لاتک به [۱، ۳] مراجعه کنید.

این مقدمه به ۶ فصل تقسیم می‌شود:

فصل ۱ شما را از ساختار ابتدایی نوشتارهای لاتک آگاه می‌سازد. همچنین کمی از تاریخچه لاتک نیز در این فصل گنجانده شده است. بعد از مطالعه این فصل، شمایی کلی از روش کار لاتک را می‌آموزید.

فصل ۲ به درون جزئیات حروف‌چینی نوشتار سفر می‌کند. این فصل بیشتر فرمان‌ها و محیط‌های اساسی لاتک را معرفی و تشریح می‌کند. بعد از مطالعه این فصل، توانایی تولید نوشتار خود را خواهید داشت.

فصل ۳ روش نگارش فرمول‌ها را در لاتک شرح می‌دهد. مثال‌های زیادی برای توضیح کامل قدرت واقعی لاتک در این زمینه ارائه شده است. در انتهای این فصل تمام نمادهای موجود لاتک در چندین جدول آورده شده است.

فصل ۴ روش تولید نمایه و کتاب‌نامه، و الصاق تصویرهای ای.پی.اس را شرح می‌دهد. همچنین روش تولید نوشته‌های پی.دی.اف به وسیله پی.دی.اف.لاتک بیان می‌شود و چندین بسته گسترش‌یافته معرفی می‌شود.

فصل ۵ روش تولید شکل را با کمک لاتک شرح می‌دهد. به جای رسم شکل‌ها به وسیله برنامه‌های کامپیوتری، ذخیره و الصاق آنها، یاد می‌گیرید که این شکل‌ها را چگونه در لاتک معرفی کنید و آنگاه لاتک آنها را برای شما رسم می‌کند.

فصل ۶ شامل اطلاعاتی خطرناک برای تغییر طرح نوشتار در لاتک است. این فصل به شما یاد می‌دهد که، بسته به توانایی شما، چگونه چیزهایی را تغییر دهید تا طرح زیبای خروجی لاتک را به شکلی زشت و ناراحت‌کننده تبدیل کنید.

بسیار مهم است که فصل‌های این مقدمه را به ترتیب مطالعه کنید — این کتاب آنقدر پرحجم نیست. مطمئن شوید که تمام مثال‌ها را به دقت مطالعه کرده‌اید، زیرا حجم گسترده‌ای از اطلاعات این کتاب در مثال‌هایش نهفته است.

لاتک برای بسیاری از انواع کامپیوترها، از کامپیوترهای شخصی گرفته تا مکینتاش و سیستم‌های بزرگ یونیکس و وی.ام.اس، وجود دارد. بر روی بسیاری از کامپیوترهای دانشگاه‌ها این سیستم نصب و آماده استفاده است. نصب خانگی لاتک در [۵] شرح داده شده است. اگر در نصب این سیستم به مشکل برخوردید، از کسی که این کتاب را به شما داده است کمک بگیرید. هدف این کتاب راهنمایی شما برای نصب لاتک نیست، بلکه هدف آن راهنمایی برای تولید نوشتار توسط لاتک است.

اگر به چیزهایی وابسته به لاتک احتیاج دارید، نگاهی به وبگاه شبکه آرشیو بزرگ تک (CTAN) بیندازید. صفحه خانگی این آرشیو در <http://www.ctan.org> قرار دارد. همه بسته‌های لاتک را می‌توانید از آرشیو اف.تی.پی <ftp://www.ctan.org> و سایت‌های آینه‌ای آن در سراسر جهان دریافت کنید. در کتاب ارجاع‌های دیگری به CTAN خواهید یافت، که به طور ویژه به نوشته‌ها و نرم‌افزارهایی مورد نیاز اشاره می‌کنند. به جای نوشتن متن کامل url، تنها کلمه CTAN را به همراه شاخه‌ای که باید بروید، نوشته‌ام.

اگر می‌خواهید لاتک را روی کامپیوتر خود راه‌اندازی کنید، به آدرس زیر نگاهی بیندازید:

CTAN:/tex-archive/systems

اگر نظری برای اضافه یا کم کردن این مقدمه دارید، لطفاً مرا آگاه سازید. در این رابطه که چه قسمت از این مقدمه مناسب است و چه قسمت باید بیشتر توضیح داده شود، بسیار مایل هستم که دیدگاه‌های افراد تازه‌کار را بدانم.

Tobias Oetiker <tobi@oetiker.ch>

OETIKER+PARTNER AG

Aarweg 15

4600 Olten

Switzerland

فهرست مطالب

پ	پیشگفتار مترجم
ث	پیشگفتار
۱	۱ چیزهایی که باید بدانید
۱	۱.۱ عنوان بازی
۱	۱.۱.۱ تک
۱	۲.۱.۱ لاتک
۲	۲.۱ میانی
۲	۱.۲.۱ نویسنده، طراحی کتاب، و حروف چینی
۲	۲.۲.۱ طراحی سبک
۳	۳.۲.۱ مزیت‌ها و اشکالات
۴	۳.۱ فایل‌های ورودی لاتک
۴	۱.۳.۱ فاصله‌ها
۴	۲.۳.۱ حروف ویژه
۵	۳.۳.۱ فرمان‌های لاتک
۵	۴.۳.۱ توضیحات
۶	۴.۱ ساختار فایل‌های ورودی
۸	۵.۱ یک دوره خط فرمان
۹	۶.۱ طرح‌بندی نوشتار
۹	۱.۶.۱ سبک نوشتار
۱۱	۲.۶.۱ بسته‌ها
۱۱	۳.۶.۱ شکل صفحات
۱۳	۷.۱ فایل‌هایی که با آنها مواجه می‌شوید
۱۴	۸.۱ پروژه‌های بزرگ
۱۷	۲ حروف چینی متن
۱۷	۱.۲ ساختار متن و زبان

۱۹	شکستن خط و صفحه	۲.۲
۱۹	ترازش پاراگرافها	۱.۲.۲
۲۰	شکستن کلمات	۲.۲.۲
۲۱	رشته‌های تعریف شده	۳.۲
۲۱	حروف و نمادهای ویژه	۴.۲
۲۱	علامت نقل قول	۱.۴.۲
۲۲	فاصله کلمات و شکستن	۲.۴.۲
۲۲	تیلدا (~)	۳.۴.۲
۲۲	اسلش (/)	۴.۴.۲
۲۲	علامت درجه (°)	۵.۴.۲
۲۳	نماد واحد پول اروپا (€)	۶.۴.۲
۲۳	سه نقطه (...)	۷.۴.۲
۲۴	چسبیدگی حروف	۸.۴.۲
۲۴	لهجه‌ها و حروف ویژه	۹.۴.۲
۲۵	پشتیبانی زبان‌های دیگر	۵.۲
۲۵	گزینه یونیکد	۱.۵.۲
۲۶	فاصله بین کلمات	۶.۲
۲۶	عنوان، فصل، و بخش	۷.۲
۲۸	ارجاع	۸.۲
۲۹	پانویس	۹.۲
۳۰	تاکید کلمات	۱۰.۲
۳۰	محیطها	۱۱.۲
۳۱	محیطهای تبصره، توضیح، و شماره‌دار	۱.۱۱.۲
۳۱	چپ، راست، و وسط چین	۲.۱۱.۲
۳۲	نقل قول و شعر	۳.۱۱.۲
۳۳	مقدمه	۴.۱۱.۲
۳۳	چاپ تحت‌اللفظ	۵.۱۱.۲
۳۴	جدول	۶.۱۱.۲
۳۶	اجسام شناور	۱۲.۲
۳۹	حفاظت از اجسام شکستنی	۱۳.۲
۴۱	حروف چینی فرمول‌های ریاضی	۳
۴۱	کلاف $AMS-LATEX$	۱.۳
۴۲	فرمول‌های تنها	۲.۳
۴۴	سبک ریاضی	۱.۲.۳
۴۵	ساختن بلوک‌های فرمولی	۳.۳

۵۰	تنظیم عمودی	۴.۳
۵۰	فرمول‌های چندگانه	۱.۴.۳
۵۰	آرایه و ماتریس	۲.۴.۳
۵۲	فاصله در محیط ریاضی	۵.۳
۵۳	اشباح	۱.۵.۳
۵۳	ریزه‌کاری با قلم‌های ریاضی	۶.۳
۵۴	حروف سیاه	۱.۶.۳
۵۴	قضیه‌ها، قانون‌ها	۷.۳
۵۷	فهرست نمادهای ریاضی	۸.۳
۶۹	ابزارهای ویژه	۴
۶۹	الصاق بسته‌های پست‌اسکرپت	۱.۴
۷۱	کتاب‌نامه	۲.۴
۷۲	نمایه‌سازی	۳.۴
۷۳	سریگ‌های تجملی	۴.۴
۷۵	بسته Verbatim	۵.۴
۷۵	نصب بسته‌های اضافی	۶.۴
۷۶	کار با پی.دی.اف لاتک	۷.۴
۷۷	نوشتارهای پی.دی.اف برای وب	۱.۷.۴
۷۸	قلم‌ها	۲.۷.۴
۷۸	استفاده از گرافیک	۳.۷.۴
۷۹	ارجاع متنی	۴.۷.۴
۸۱	مشکلات اتصال‌ها	۵.۷.۴
۸۲	مشکلات چوب الف	۶.۷.۴
۸۴	تولید اسلاید	۸.۴
۸۷	تولید شکل‌های ریاضی	۵
۸۷	مرور	۱.۵
۸۸	محیط تصویر	۲.۵
۸۸	فرمان‌های ابتدایی	۱.۲.۵
۹۰	پاره‌خط	۲.۲.۵
۹۱	پیکان‌ها	۳.۲.۵
۹۲	دایره	۴.۲.۵
۹۳	متن و فرمول	۵.۲.۵
۹۴	\linethickness و \multiput	۶.۲.۵
۹۵	بیضی	۷.۲.۵
۹۶	استفاده چندباره از جعبه‌های تصویر پیش‌ساخته	۸.۲.۵

۹۸	۹.۲.۵	خم‌های درجهٔ دوم بزیه
۹۹	۱۰.۲.۵	تسبیح
۱۰۰	۱۱.۲.۵	سرعت در نظریه نسبیت عام
۱۰۰	۳.۵	بستهٔ گرافیک TikZ & PGF
۱۰۳	۴.۵	Xy-pic
۱۰۷	۶	تنظیم شخصی لاتک
۱۰۷	۱.۶	فرمان‌ها، محیط‌ها، و بسته‌های جدید
۱۰۸	۱.۱.۶	فرمان‌های جدید
۱۰۹	۲.۱.۶	محیط‌های جدید
۱۰۹	۳.۱.۶	فاصله‌های اضافه
۱۱۰	۴.۱.۶	خط فرمان لاتک
۱۱۱	۵.۱.۶	بسته‌های شخصی
۱۱۱	۲.۶	قلم‌ها و اندازهٔ آنها
۱۱۱	۱.۲.۶	فرمان تغییر قلم
۱۱۴	۲.۲.۶	خطر، ویل رابینسون، خطر
۱۱۵	۳.۲.۶	توصیه
۱۱۵	۳.۶	فاصله‌گذاری
۱۱۵	۱.۳.۶	فاصلهٔ خط‌ها
۱۱۶	۲.۳.۶	شکل پاراگراف
۱۱۶	۳.۳.۶	فاصله افقی
۱۱۷	۴.۳.۶	فاصله عمودی
۱۱۸	۴.۶	طرح صفحه
۱۲۰	۵.۶	بازی بیشتر با طول‌ها
۱۲۱	۶.۶	جعبه‌ها
۱۲۳	۷.۶	\strut و \rule
۱۲۵		کتاب‌نامه

فهرست تصاویر

۷	یک فایل لاتک نمونه	۱.۱
۷	مثالی از یک فایل مقاله مجله	۲.۱
۷۴	مثال بارگذاری fancyhdr	۱.۴
۸۵	کد نمونه برای طبقه beamer	۲.۴
۱۱۱	مثال بسته	۱.۶
۱۱۹	پارامترهای طرح صفحه	۲.۶

فهرست جداول

۹	سبک‌های نوشتار	۱.۱
۱۰	گزینه‌های سبک نوشتار	۲.۱
۱۲	تعدادی از بسته‌هایی که به همراه توزیع لاتک ارائه می‌شوند	۳.۱
۱۲	سبک‌های صفحه از پیش تعریف شده لاتک	۴.۱
۲۳	کیسه‌ای پر از نماد اروپا	۱.۲
۲۵	لهجه‌ها و حروف ویژه	۲.۲
۳۷	پارامترهای قراردادن اجسام شناور	۳.۲
۵۷	لهجه‌های سبک ریاضی	۱.۳
۵۷	الفبای یونانی	۲.۳
۵۸	روابط دوتایی	۳.۳
۵۹	عملگرهای دوتایی	۴.۳
۵۹	عملگرهای بزرگ	۵.۳
۶۰	پیکان‌ها	۶.۳
۶۰	پیکان‌ها به عنوان لهجه	۷.۳
۶۱	حائل‌ها	۸.۳
۶۱	حائل‌های بزرگ	۹.۳
۶۱	نمادهای متفرقه	۱۰.۳
۶۲	نمادهای غیر ریاضی	۱۱.۳
۶۳	حائل‌های AMS	۱۲.۳
۶۳	AMS یونانی و عبری	۱۳.۳
۶۳	الفبای ریاضی	۱۴.۳
۶۳	عملگرهای دوتایی AMS	۱۵.۳
۶۴	روابط دوتایی AMS	۱۶.۳
۶۵	پیکان‌های AMS	۱۷.۳
۶۶	نقیض روابط دوتایی و پیکان‌های AMS	۱۸.۳

۶۷	متفرقه AMS	۱۹.۳
۷۰	نام کلیدها برای بسته graphicx	۱.۴
۷۳	مثال‌هایی از شکل کلیدها	۲.۴
۱۱۲	قلم‌ها	۱.۶
۱۱۲	اندازه قلم	۲.۶
۱۱۳	اندازه واقعی قلم در طبقه استاندارد	۳.۶
۱۱۳	قلم‌های ریاضی	۴.۶
۱۱۷	کمیت‌های تک	۵.۶

فصل ۱

چیزهایی که باید بدانید

اولین قسمت این فصل به بررسی فلسفه و تاریخچه $\text{\LaTeX} 2_{\epsilon}$ اختصاص دارد. قسمت دوم متمرکز به ساختار $\text{\LaTeX} 2_{\epsilon}$ است. بعد از مطالعه این فصل درمی یابید که \LaTeX چگونه کار می کند، که برای مطالعه ادامه کتاب لازم است.

۱.۱ عنوان بازی

۱.۱.۱ تک

تک یک برنامه کامپیوتری است که توسط دونالد کنوت [۲] ساخته شده است. هدف آن حروف چینی متن عادی و ریاضی است. کنوت در سال ۱۹۷۷ شروع به نوشتن تک کرد تا قدرت پنهانی ابزار چاپ دیجیتال را که در آن زمان در صنعت چاپ رخنه کرده بود مورد کاوش قرار دهد به این امید که بدی کیفیت حروف چینی کتابها و مقالات خودش را از بین ببرد. تک به این صورت که امروزه ما مورد استفاده قرار می دهیم در سال ۱۹۸۲ انتشار یافت و در سال ۱۹۸۹ امکانات حمایت حروف ۸ بیتی و دیگر زبانها به آن اضافه شد. شهرت تک در این است که بسیار پایدار است، روی هر سیستم عاملی قابل نصب است، و به طور مجازی فارغ از اشکال است. نسخه کنونی تک ۳,۱۴۱۵۹۲ است که به عدد π میل می کند.

۲.۱.۱ لاتک

لاتک یک بسته از ماکروها است که به نویسندگان امکان حروف چینی و چاپ کارهایشان را با بهترین کیفیت با استفاده از تعدادی طرح حرفه ای می دهد. لاتک در ابتدا توسط لژلی لمپورت [۱] نوشته شد که از تک به عنوان موتور حروف چین استفاده می کند. این روزها لاتک توسط فرآنگ میتل باخ حمایت می شود.

۲.۱ مبانی

۱.۲.۱ نویسنده، طراحی کتاب، و حروف چینی

برای انتشار چیزی نویسندگان نوشته خود را به مؤسسات انتشاراتی می‌دهند. یکی از طراحان کتاب در مورد سبک نوشته تصمیم می‌گیرد (عرض ستون، قلم، فاصله قبل و بعد از سربرگ، ...). طراح کتاب راهنمایی لازم را به حروف چین می‌کند تا کتاب را بر طبق آن حروف چینی کند.

طراح کتاب سعی می‌کند بفهمد خواست نویسنده هنگام نوشتن کتاب چه بوده است. او در مورد سربرگ فصل‌ها، ارجاع‌ها، مثال‌ها، فرمول‌ها، و غیره بر اساس اطلاعات حرفه‌ای خود و اطلاعات در مورد محتوای نوشته تصمیم می‌گیرد.

در محیط لاتک، لاتک نقش طراح کتاب را برعهده می‌گیرد و از تک به عنوان حروف چین استفاده می‌کند. اما لاتک تنها یک برنامه است و بنابراین نیاز به راهنمایی دارد. نویسنده باید اطلاعات کافی در مورد ساختار منطقی کارش را به لاتک بدهد. این اطلاعات در متن به صورت فرمان‌های لاتک وارد می‌شوند.

این کار کاملاً با روش WYSIWYG^۱ تفاوت دارد که بسیاری از پردازشگرهای متنی مانند MS Word یا Corel WordPerfect از آن پیروی می‌کنند. در این نرم‌افزارها، نویسنده سبک نوشتار را به صورت مستقیم هنگام نوشتن آن مشخص می‌کند. در این نرم‌افزارها شکل خروجی را، همزمان که نوشتار را تایپ می‌کنید، به صورت مستقیم می‌توان بر روی صفحه نمایش دید.

وقتی که از لاتک استفاده می‌کنید به طور نرمال نمی‌توانید همزمان با تایپ متن شکل خروجی را ببینید، اما می‌توانید آن را بعد از پردازش توسط لاتک مشاهده کنید. در این صورت تصحیحات را می‌توان قبل از فرستادن نوشته به چاپگر انجام داد.

۲.۲.۱ طراحی سبک

حروف چینی یک هنر است. نویسنده‌های ناوارد معمولاً اشتباهات اساسی در هنگام طراحی انجام می‌دهند زیرا فکر می‌کنند طراحی تماماً مربوط به علم زیبایی شناسی است «اگر یک متن از نظر زیبایی خوب باشد، خوب طراحی شده است.» اما از آنجا که یک کتاب را باید خواند نه آنکه در یک نمایشگاه عکس آویزان کرد، خوانایی و قابل فهم بودن آن بسیار مهم‌تر از ظاهر زیبای آن است. به عنوان مثال:

- نوع و اندازه قلم شماره‌بندی سربرگ باید به گونه‌ای انتخاب شود که ساختار فصل‌ها و بخش‌ها برای خواننده واضح باشد.
- طول خط‌ها باید به اندازه کافی کوتاه باشد تا چشمان خواننده را خسته نکند و همزمان باید به اندازه کافی بلند باشد تا زیبایی صفحات را از بین نبرد.

با سیستم‌های WYSIWYG، نویسنده‌ها معمولاً نوشتارهای زیبا اما فاقد ساختار سازگار را تولید می‌کنند. لاتک با مجبور کردن نویسنده به مشخص کردن ساختار منطقی نوشته‌اش از چنین اشتباهی جلوگیری می‌کند. لاتک آنگاه طراحی بهترین سبک را به عهده می‌گیرد.

^۱What you see is what you get.

۳.۲.۱ مزیت‌ها و اشکالات

افرادی که از سیستم WYSIWYG یا لاتک استفاده می‌کنند، اغلب در مورد «مزیت لاتک بر پردازشگرهای عادی» یا عکس آن بحث می‌کنند. بهترین کاری که هنگام مواجهه با این بحث باید انجام دهید این است که از ادامه بحث پرهیز کنید زیرا اغلب بدون نتیجه است. اما گاهی اوقات فرار از چنین بحثی ممکن نیست.

بنابراین کمی مهمات همراه داشته باشید. مهمترین مزیت لاتک بر یک سیستم پردازشگر عادی متن از قرار زیر است:

- سبک‌های زیبای حرفه‌ای موجودند که متن را آن گونه طراحی می‌کنند که واقعاً باید چاپ شود.
- حروف چینی فرمول‌های ریاضی به بهترین شکل حمایت می‌شود.
- کاربر تنها کافی است تعدادی فرمان آسان را یاد بگیرد تا ساختار منطقی نوشته‌اش را طراحی کند. معمولاً لازم نیست در مورد ساختار واقعی متن نگران باشید.
- حتی ساختارهای پیچیده مانند پانوشته‌ها، ارجاع‌ها، فهرست مطالب، و کتاب‌نامه به راحتی قابل تولید هستند.
- بسته‌های اضافی مجانی بسیاری برای کارهایی که لاتک انجام نمی‌دهد وجود دارند. به عنوان مثال بسته‌های پست‌اسکرپت برای گرافیک یا بسته‌هایی برای قرار دادن ارجاع‌ها به شکل استاندارد وجود دارند. بسیاری از این بسته‌ها در [۳] توضیح داده شده‌اند.
- لاتک نویسنده‌ها را تشویق می‌کند نوشته‌های خود را با ساختار مناسب بنویسند، زیرا این روشی است که لاتک از آن پیروی می‌کند.
- تک، موتور لاتک، بسیار قابل انعطاف و مجانی است. بنابراین، این سیستم روی هر سیستم عاملی کار می‌کند.
- لاتک دارای بدیهی‌هایی نیز می‌باشد که برای من سخت است آنها را حدس بزنم، با این وجود مطمئنم افراد دیگر ممکن است صدتا از آنها را به شما گوشزد کنند (-)؛
- لاتک برای افرادی که روح خودشان را فروخته باشند مناسب نیست ...
- با وجودی که بعضی از پارامترها را می‌توان در یک نوشتار تنظیم کرد، طراحی یک سبک جدید سخت و زمان‌بر است.^۲
- بسیار سخت است که متن‌های بدون ساختار نوشت.
- همستر^۳ شما حتی با تشویق‌های اولین قدم‌ها، ممکن است هیچ‌گاه مفهوم نقاط علامت گذاری شده را درنیابد.

^۲شایعاتی وجود دارد که رفع این مشکل مهمترین کار لاتک ۳ است.

^۳Hamster

۳.۱ فایل‌های ورودی لاتک

ورودی لاتک یک فایل اَسْکی ساده است که می‌توان آن را با هر ویرایشگری نوشت. این ورودی شامل متن و فرمان‌هایی است که مشخص می‌کند متن چگونه باید حروف‌چینی شود.

۱.۳.۱ فاصله‌ها

لاتک با حروف «فاصله سفید» مانند حرف فاصله^۴ یا تب^۵ به طور یکسان به عنوان «فاصله» رفتار می‌کند. با فاصله‌های متوالی همانند یک فاصله رفتار می‌شود. فاصله سفید در ابتدای خط بی‌اثر است، و با یک شکستن خط مانند «فاصله سفید» رفتار می‌شود.

یک خط خالی بین دو خط از متن پایان یک پاراگراف را مشخص می‌کند. «چند» خط خالی متوالی مانند تنها «یک» خط خالی است. متن زیر یک نمونه است. در سمت چپ متن ورودی قرار دارد و در سمت راست شکل خروجی قرار دارد.

It does not matter whether you
enter one or several spaces
after a word.

An empty line starts a new
paragraph.

It does not matter whether you enter one or
several spaces after a word.

An empty line starts a new paragraph.

۲.۳.۱ حروف ویژه

نمادهای زیر حروف اختصاصی هستند که یا دارای معنای ویژه در لاتک هستند یا در همه قلم‌ها وجود ندارند. اگر آنها را مستقیماً در متن به‌کار برید در خروجی ظاهر نمی‌شوند و لاتک را مجبور به کاری غیر مرتبط می‌کنند.

\$ % ^ & _ { } ~ \

همان‌طور که خواهید دید این حروف را می‌توانید در متن با افزودن یک پیشوند بک‌اسلش^۶ مورد استفاده قرار دهید:

\# \\$ \% \^{} \& _ \{ \} \~{} \

\$ % ^ & _ { } ~

^۴Blank

^۵Tab

^۶backslash

بقیه نمادها و بسیاری چیزهای دیگر را می‌توان در فرمول‌های ریاضی یا به عنوان لهجه‌های مختلف با فرمان‌هایی چاپ کرد. بک‌اسلش را نمی‌توان با افزودن یک بک‌اسلش دیگر مانند \backslash چاپ کرد؛ این رشته برای شکستن خط به‌کار می‌رود.^۷

۳.۳.۱ فرمان‌های لاتک

فرمان‌های لاتک به کوچک و بزرگ بودن حروف حساس است و یکی از دو شکل زیر را می‌پذیرند:

- با یک بک‌اسلش \backslash شروع می‌شوند و دارای اسمی هستند که تنها از حروف تشکیل شده است. اسم فرمان‌ها با یک فاصله یا یک عدد و یا هر «غیر حرف» پایان می‌یابد.
- از یک بک‌اسلش و تنها یک غیر حرف تشکیل شده‌اند.

لاتک از فاصله خالی بعد از فرمان‌ها چشم‌پوشی می‌کند. اگر می‌خواهید بعد از آنها فاصله خالی داشته باشید بعد از فرمان، $\{$ به همراه یک فاصله قرار دهید یا از یک فرمان ویژه فاصله استفاده کنید. $\}$ باعث می‌شود لاتک تمام فضای خالی بعد از فرمان را از بین نبرد.

```
I read that Knuth divides the
people working with \TeX{} into
\TeX{}nicians and \TeX perts.\
Today is \today.
```

```
I read that Knuth divides the people working
with TEX into TEXnicians and TEXperts.
Today is November 5, 2014.
```

بعضی از فرمان‌ها احتیاج به پارامتر دارند که آنها را در آکولاد $\{ \}$ قرار می‌دهیم. بعضی از فرمان‌ها پارامترهای اختیاری قبول می‌کنند که آنها را در کروشه $[]$ قرار می‌دهیم. مثال‌های بعد چند فرمان در لاتک را نشان می‌دهند. نگران نباشید، آنها را بعداً توضیح می‌دهیم.

```
You can \textsl{lean} on me!
```

```
You can lean on me!
```

```
Please, start a new line
right here!\newline
Thank you!
```

```
Please, start a new line right here!
Thank you!
```

۴.۳.۱ توضیحات

هنگام پردازش فایل ورودی، وقتی لاتک با یک $\%$ مواجه می‌شود، ادامه خط، شکست خط، و فاصله‌های خالی خط بعد را نادیده می‌گیرد. با استفاده از این موضوع می‌توان چیزهایی را در متن آورد که در هنگام چاپ ظاهر نشوند.

^۷ به جای آن از \backslash استفاده کنید. این کار باعث چاپ \backslash می‌شود.

```
This is an % stupid
% Better: instructive <----
example: Supercal%
         ifragilist%
         icexpialidocious
```

This is an example: Supercalifragilisticexpialidocious

از % می‌توان استفاده کرد و خط‌های فایل ورودی را شکست حتی وقتی که فاصله خالی یا شکست خط در خروجی مورد نظر نیست. برای توضیحات طولانی باید از محیط `comment` از بسته `verbatim` استفاده کرد. برای این منظور باید عبارت `\usepackage{verbatim}` را در آغاز فایل ورودی قبل از استفاده از آن وارد کنید همان‌طور که در مثال زیر آمده است.

```
This is another
\begin{comment}
rather stupid,
but helpful
\end{comment}
example for embedding
comments in your document.
```

This is another example for embedding comments in your document.

توجه داشته باشید که این کار را در محیط‌های پیچیده مانند محیط ریاضی نمی‌توانید انجام دهید.

۴.۱ ساختار فایل‌های ورودی

وقتی لاتک یک فایل ورودی را پردازش می‌کند انتظار دارد که فایل از یک ساختار پیروی کند. بنابراین هر فایل ورودی باید با فرمان

```
\documentclass{...}
```

آغاز شود. این کار مشخص می‌کند که چه نوع نوشتاری را می‌خواهید بنویسید. بعد از آن فرمان‌های مورد نیاز را باید معرفی کنید و یا بسته‌هایی را بارگذاری کنید که امکانات جدیدی را به لاتک اضافه می‌کنند. برای بارگذاری یک بسته از فرمان زیر استفاده می‌کنیم:

```
\usepackage{...}
```

وقتی تمام این مقدمات انجام شد،^۸ باید متن به همراه فرمان‌های مفید را وارد کنید. در انتهای فایل ورودی فرمان

```
\end{document}
```

^۸فاصله بین `\documentclass` و `\begin{document}` سرآغاز یا *preamble* نامیده می‌شود.

را وارد کنید تا به لاتک بفهمانید همه چیز تمام شده است. بعد از این فرمان چیزی توسط لاتک در نظر گرفته نمی‌شود.

شکل ۱.۱ محتویات یک فایل ساده لاتک را نشان می‌دهد. مثالی کمی پیچیده‌تر از یک فایل ورودی در شکل ۲.۱ آورده شده است.

```

\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}

```

شکل ۱.۱: یک فایل لاتک نمونه

```

\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}

```

شکل ۲.۱: مثالی از یک فایل مقاله مجله. تمام فرمان‌هایی که در این مثال وجود دارند بعداً در مقدمه شرح داده خواهند شد.

۵.۱ یک دوره خط فرمان

شرط می‌بندم تصمیم گرفته‌اید مثال جمع‌وجور صفحه ۷ را شخصاً انجام بدهید. چند راهنمایی: خود لاتک بدون هیچ رابط کاربر گرافیکی^۹ یا کلیدهای تجملی ارائه می‌شود. لاتک فقط یک برنامه است که فایل ورودی را پردازش می‌کند. بعضی از توزیع‌های لاتک دارای رابط کاربری هستند که با فشردن یک دکمه می‌توانید فایل خود را پردازش کنید. در غیر این صورت باید در یک خط فرمان چند فرمان را تایپ کنید تا لاتک فایل ورودی را پردازش کند. پس اجازه دهید این کار را کمی توضیح دهیم. توجه: این توضیحات بر این فرض استوار است که شما لاتک را روی سیستم خود داشته باشید.^{۱۰}

۱. فایل لاتک ورودی خود را بنویسید. این فایل باید یک متن ساده اسکی باشد. در لینوکس تمام ویرایشگرها می‌توانند این کار را انجام دهند. در ویندوز مطمئن شوید فایل را به فرم اسکی یا متن ساده ذخیره کرده‌اید. از `tex`. به عنوان پسوند فایل خود استفاده کنید.

۲. لاتک را روی فایل خود اجرا کنید. اگر موفق شوید یک فایل `dvi`. بدست خواهد آمد. ممکن است لازم باشد لاتک را چندین بار روی فایل خود اجرا کنید تا فهرست و تمام ارجاع‌های داخلی را داشته باشید. وقتی که فایل ورودی مشکل داشته باشد لاتک به شما پیغام خواهد داد و پردازش را متوقف می‌کند. `ctrl-D` را تایپ کنید تا به خط فرمان برگردید.

```
latex foo.tex
```

۳. حال می‌توانید فایل `DVI` را مشاهده کنید. چندین راه برای انجام این کار وجود دارد. می‌توانید فایل را روی صفحه نمایش با فرمان

```
xdvi foo.dvi &
```

مشاهده کنید. این کار را تنها روی سیستم لینوکس مجهز به X11 انجام دهید. اگر سیستم شما ویندوز است از `yap`^{۱۱} استفاده کنید. همچنین می‌توانید فایل `dvi` را به پست‌اسکرپت برای مشاهده با گوست‌اسکرپت^{۱۲} یا چاپ تبدیل کنید.

```
dvips -Pcmz foo.dvi -o foo.ps
```

اگر خوش‌شانس باشید سیستم لاتک شما دارای ابزار `dvipdf` است که به شما اجازه می‌دهد فایل `dvi`. را مستقیماً به `pdf` تبدیل کنید.

```
dvipdf foo.dvi
```

^۹GUI

^{۱۰}لاتک روی تمام سیستم‌های لینوکس که کامل نصب شده باشند وجود دارد، و... مردها با لینوکس کار می‌کنند، بنابراین (-)؛

^{۱۱}yet another previewer

^{۱۲}Ghostscript

۶.۱ طرح‌بندی نوشتار

۱.۶.۱ سبک نوشتار

وقتی که لاتک یک فایل ورودی را پردازش می‌کند اولین اطلاعاتی را که باید بداند سبک نوشتار است. این موضوع با فرمان `\documentclass` مشخص می‌شود.

```
\documentclass[options]{class}
```

در اینجا `class` سبک نوشتار را معرفی می‌کند. جدول ۱.۱ سبک‌های نوشتاری را نشان می‌دهد که در این مقدمه شرح داده خواهند شد. توزیع لاتک سبک‌های نوشتار دیگری مانند `letter` و `slide` را نیز شامل است. پارامترهای گزینه (`options`) رفتار سبک نوشتار را کنترل می‌کنند. پارامترها توسط ویرگول از یکدیگر جدا می‌شوند. معمول‌ترین گزینه‌ها برای سبک‌های نوشتار استاندارد در جدول ۲.۱ آورده شده است.

مثال: یک فایل ورودی لاتک می‌تواند به صورت زیر شروع شود

```
\documentclass[11pt,twoside,a4paper]{article}
```

که به لاتک می‌گوید نوشتار را به صورت مقاله با اندازه قلم پایه ۱۱ پوینت حروف‌چینی کند، و سبک دورو را برای چاپ روی صفحه A4 طراحی کند.

جدول ۱.۱: سبک‌های نوشتار

برای مقالات مجله‌ها، ارائه‌ها، گزارش‌های کوتاه، اسناد برنامه‌ها، دعوت‌نامه، ...	article
سبکی برای گزارش پیشرفت برپایه سبک article	proc
کوچکترین چیزی که می‌توان قرار داد. تنها شامل یک صفحه و یک قلم است. عموماً به منظور غلط‌یابی به کار می‌رود.	minimal
برای گزارش‌های مفصل‌تر که شامل چند فصل هستند، کتاب، پایان‌نامه، ...	report
برای کتاب‌های کامل	book
برای اسلاید. این سبک از حروف بزرگ سازز سریف استفاده می‌کند. به جای آن ممکن است بخواهید از فویل‌تک استفاده کنید. ^۱	slides

جدول ۲.۱: گزینه‌های سبک نوشتار

اندازه قلم اصلی نوشتار را تعیین می‌کند. اندازه پیش فرض 10pt است.	10pt, 11pt, 12pt
اندازه صفحه را مشخص می‌کند. اندازه پیش فرض letterpaper است. بجز این‌ها a5paper, b5paper, executivepaper, و legalpaper نیز قابل استفاده هستند.	a4paper, letterpaper, ...
فرمول‌ها به جای وسط‌چین چپ‌چین می‌شوند.	fleqn
شماره فرمول‌ها در سمت چپ به جای سمت راست ظاهر می‌شوند.	leqno
مشخص می‌کند که آیا صفحه‌ای جدید بعد از صفحه عنوان شروع شود یا نه. سبک article صفحه‌ای جدید به صورت پیش فرض شروع نمی‌کند در حالی که سبک‌های report و book این کار را انجام می‌دهند.	titlepage, notitlepage
لاتک را راهنمایی می‌کنند که نوشتار را در یک ستون یا در دو ستون حروف‌چینی کند.	onecolumn, twocolumn
مشخص می‌کند که خروجی به صورت یک‌رو است یا دورو. به صورت پیش فرض سبک‌های article و report یک‌روهستند و سبک book دورو است. توجه کنید که این گزینه فقط به سبک نوشتار مربوط است. گزینه twoside به چاپگر فرمان چاپ دورو نمی‌دهد.	twoside, oneside
سبک نوشتار را به صورت افقی (landscape) تبدیل می‌کند.	landscape
باعث می‌شود فصل‌ها در صفحه‌های سمت راست یا در صفحه بعدی شروع شوند. این گزینه با سبک article کار نمی‌کند زیرا در این سبک فصل وجود ندارد. سبک report به صورت پیش فرض فصل‌ها را در صفحه بعدی و سبک book آنها را در صفحات سمت راست شروع می‌کند.	openright, openany

۲.۶.۱ بسته‌ها

هنگامی که در حال نوشتن نوشتار خود هستید، ممکن است به مراحل برسی که لاتک نتواند مشکلات شما را حل کند. اگر می‌خواهید تصویر، متن رنگی یا کد یک مطلب را در نوشتار خود وارد کنید، احتیاج به بالابردن توانایی لاتک دارید. این کار را با استفاده از بسته‌ها انجام می‌دهیم. یک بسته را فرمان زیر فعال می‌سازد

```
\usepackage [options] {package}
```

که *package* نام یک بسته است و *options* لیستی از کلمه‌های کلیدی است که امکانات ویژه‌ای از بسته را فعال می‌سازند. بعضی از بسته‌ها با توزیع پایه لاتک ارائه می‌شوند (جدول ۳.۱ را ببینید). تعدادی دیگر از این بسته‌ها به‌طور جداگانه عرضه می‌شوند. می‌توانید اطلاعات بسته‌های نصب شده روی سیستم خود را در [۵] ببینید. منبع اولیه برای اطلاعات در مورد بسته‌های لاتک [۳] است که شامل شرح صدها بسته است و همچنین اطلاعاتی در مورد نوشتن بسته‌هایی برای افزودن به لاتک است. توزیع‌های جدید تک با تعداد بسیار زیادی از بسته‌های از پیش نصب شده همراه است. اگر با لینوکس کار می‌کنید فرمان `texdoc` را وارد کنید تا اطلاعات بسته‌ها را دریافت کنید.

۳.۶.۱ شکل صفحات

لاتک سه نوع از پیش تعریف شده سربرگ^{۱۳} / ته‌برگ^{۱۴} را حمایت می‌کند که به سبک صفحه^{۱۵} معروف هستند. پارامتر *style* از فرمان

```
\pagestyle{style}
```

مشخص می‌کند که کدام پارامتر باید مورد استفاده قرار گیرد. جدول ۴.۱ حاوی سبک‌های صفحه از پیش تعریف شده است.

می‌توان سبک صفحه جاری را با فرمان

```
\thispagestyle{style}
```

عوض کرد. توضیحی بر این که چگونه سربرگ و ته‌برگ مناسب خود را طراحی کنید در [۲] و در بخش ۴.۴ در صفحه ۷۳ داده شده است.

^{۱۳} footer

^{۱۴} header

^{۱۵} page style

جدول ۳.۱: تعدادی از بسته‌هایی که به همراه توزیع لاتک ارائه می‌شوند

doc	اجازه برنامه اطلاعات لاتک را می‌دهد. شرح آن در فایل doc.dtx ^آ و در [۳] داده شده است.
exscale	اندازه قلم‌های ریاضی را فراهم می‌کند. در فایل ltexscale.dtx توضیح داده شده است.
fontenc	مشخص می‌کند لاتک باید از چه رمزینۀ قلم ^ب استفاده کند. در فایل ltoutenc.dtx توضیح داده شده است.
ifthen	فرمان‌های به شکل 'if...then do...otherwise do...' را فراهم می‌کند. در فایل ifthen.dtx و [۳] توضیح داده شده است.
latexsym	برای دستیابی به نماد L ^A T _E X باید از بسته latexsym استفاده کنید. در فایل latexsym.dtx و در [۳] توضیح داده شده است.
makeidx	شامل فرمان‌هایی برای تولید نمایه است. در بخش ۳.۴ و در [۳] توضیح داده شده است.
syntonly	یک نوشتار را پردازش می‌کند بدون آنکه آن را حروف‌چینی کند.
inputenc	اجازه رمزینۀهایی مانند ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows, user-defined را می‌دهد. در inputenc.dtx توضیح داده شده است.

^آ این فایل باید روی سیستم نصب شده باشد و می‌توانید یک فایل dvi را با نوشتن فرمان latex doc.dtx در هر پرونده‌ای که اجازه نوشتن در آن داشته باشید دریافت کنید. مطلب مشابهی برای فایل‌های دیگر این جدول برقرار است.

^ب font encoding

جدول ۴.۱: سبک‌های صفحه از پیش تعریف شده لاتک

plain	شماره صفحه را در وسط انتهای صفحه در تهر برگ چاپ می‌کند. این سبک پیش فرض است.
headings	عنوان فصل جاری را در سربرگ در تمام صفحات چاپ می‌کند، اما تهر برگ خالی باقی می‌ماند. (این سبکی است که در این مقدمه مورد استفاده قرار گرفته است)
empty	سربرگ و تهر برگ را خالی چاپ می‌کند.

۷.۱ فایل‌هایی که با آنها مواجه می‌شوید

وقتی که با لاتک کار می‌کنید با انبوهی از فایل‌ها با پسوندهای مختلف مواجه می‌شوید که احتمالاً هیچ ایده‌ای از دلیل وجود آنها ندارید. لیست زیر انواع فایل‌هایی را توضیح می‌دهد که هنگام کار با لاتک با آنها مواجه می‌شوید. توجه داشته باشید که این لیست تمام فایل‌های ممکن را دربر ندارد، ولی اگر فکر می‌کنید نوع مهمی از قلم افتاده است لطفاً به من اطلاع دهید.

.tex فایل ورودی تک یا لاتک. لاتک آن را پردازش می‌کند.

.sty بستهٔ ماکروهای لاتک. این نوعی از فایل است که شما با فرمان `\usepackage` به فایل ورودی وارد می‌کنید.

.dtx اطلاعات تک. این نوع اساسی‌ترین نوع برای فایل‌های استایل است. اگر یک فایل از این نوع را پردازش کنید، اطلاعات بستهٔ شامل آن فایل را بدست می‌آورید.

.ins فایل نصب‌کنندهٔ فایل‌های موجود در فایل `.dtx`. اگر بسته‌ای را از اینترنت دانلود کنید به طور نرمال شامل یک فایل `.dtx` و یک فایل `.ins` است. فایل `.ins` را توسط لاتک پردازش کنید تا فایل `.dtx` را باز کنید.

.cls فایل‌های کلاس که سبک نوشتار را مشخص می‌کنند. این فایل‌ها را با فرمان `\documentclass` فراخوانی می‌کنیم.

.fd فایل‌های قلم که لاتک را از آنها آگاه می‌سازد.

وقتی که لاتک فایل را پردازش می‌کند فایل‌های زیر را تولید می‌کند:

.dvi فایل مستقل از دستگاه. این فایل مهمترین خروجی لاتک است. محتویات آن را می‌توان با نمایشگر مخصوص آن ببینید یا می‌توانید آن را توسط `dvips` یا چیزی شبیه به آن به چاپگر بفرستید.

.log شامل همهٔ اتفاقاتی است که در هنگام پردازش قبل اتفاق افتاده است.

.toc تمام عنوان‌های بخش‌ها را ذخیره می‌کند. این فایل در زمان اجرای بعدی خوانده می‌شود و برای چاپ فهرست مطالب مورد استفاده قرار می‌گیرد.

.lof این فایل مانند فایل `.toc` است اما برای لیست تصاویر.

.lot و همین‌طور این فایل برای لیست جدول‌ها است.

.aux فایل دیگری که وظیفهٔ آن انتقال اطلاعات از پردازش قبلی به پردازش جاری است و شامل ارجاع‌ها است.

`.idx`. اگر فایل شما دارای نمایه باشد، لاتک تمام کلماتی را که باید به نمایه انتقال یابند در این فایل ذخیره می‌کند. این فایل را با `makeindex` پردازش کنید. به بخش ۳.۴ در صفحه ۷۲ برای اطلاعات بیشتر مراجعه کنید.

`.ind`. فایل پردازش شده `.idx`. که آماده تزریق به نوشتار در پردازش بعدی است.

`.ilg`. فایلی که نشان می‌دهد `makeindex` چه کاری انجام داده است.

۸.۱ پروژه‌های بزرگ

وقتی روی نوشتارهای بزرگ کار می‌کنید، ممکن است دوست داشته باشید که فایل ورودی را به چند قسمت تقسیم کنید. لاتک دو فرمان برای انجام این کار دارد.

```
\include{filename}
```

این فرمان را می‌توانید در متن نوشتار وارد کنید تا محتویات فایل `filename.tex` را به نوشتار اضافه کنید. توجه داشته باشید که لاتک یک صفحه جدید را قبل از پردازش محتویات `filename.tex` تولید می‌کند. فرمان دوم را می‌توانید در آغاز نوشتار وارد کنید. این کار به لاتک اجازه می‌دهد تنها تعدادی از فایل‌های `\include` شده را در متن وارد کند.

```
\includeonly{filename,filename,...}
```

بعد از این که این فرمان در آغاز پردازش شد، تنها فرمان‌های `\include` مربوط به آن فایل‌هایی اجرا می‌شود که نام آنها در آرگومان `\includeonly` آورده شده باشد. توجه داشته باشید که نباید هیچ فاصله‌ای بین اسم فایل‌ها و ویرگول‌ها باشد.

فرمان `\include` باعث حروف‌چینی فایل الصاق شده در یک صفحه جدید می‌شود. این موضوع به ویژه وقتی که از فرمان `\includeonly` استفاده می‌کنید مفید است زیرا شکست صفحه‌ها تغییر نمی‌کند حتی اگر بعضی از فایل‌ها الصاق شده حذف شده باشند. بعضی مواقع این کار مطلوب نیست. در این حالت می‌توانید از فرمان زیر استفاده کنید:

```
\input{filename}
```

این فرمان به طور ساده فایل‌های عنوان شده را الصاق می‌کند. بدون هیچ زرق و برقی و هیچ چیز اضافه. برای این که لاتک را مجبور کنید نوشتار شما را بررسی کند از بسته `syntonly` استفاده کنید. این بسته لاتک را مجبور می‌کند نوشتار را برای خطاهای احتمالی مورد بازرسی قرار دهد اما هیچ خروجی تولید نمی‌کند. از آنجا که لاتک در این حالت سریع‌تر اجرا می‌شود می‌تواند در ذخیره زمان بسیار مفید

باشد. کاربرد آن بسیار آسان است:

```
\usepackage{syntonly}  
\syntaxonly
```

وقتی که می‌خواهید خروجی تولید کنید تنها خط دوم را غیر فعال کنید (با افزودن یک علامت درصد).

فصل ۲

حروف چینی متن

بعد از مطالعه فصل پیش، چیزهای ابتدایی را می‌دانید که لاتک با آنها سروکار دارد. در این فصل مطالب دیگری را خواهید آموخت که برای تولید نوشته‌های واقعی مفید هستند.

۱.۲ ساختار متن و زبان

انتقال اطلاعات و ایده‌ها به خواننده مهمترین نکته نوشتن یک متن است. اگر نوشتار به درستی ساختار بندی شده باشد خواننده به راحتی مطالب را می‌فهمد و این مطلب موقعی اتفاق می‌افتد که ساختار حروف چینی متن انعکاس دهنده ساختار محتوای متن باشد.

تفاوت لاتک با دیگر سیستم‌های حروف چینی در این است که تنها باید ساختار منطقی و زیبایی نوشتار را به لاتک معرفی کرد. آنگاه لاتک با استفاده از قوانینی که در متن و در فایل‌های الصاقی ارائه شده است حروف چینی نوشتار را انجام می‌دهد.

مهمترین واحد در لاتک (و در حروف چینی) پاراگراف است. ما به آن «واحد متن» می‌گوییم زیرا پاراگراف قسمت به هم چسبیده‌ای است که یک ایده را بازگو می‌کند. در این بخش یاد می‌گیریم که چگونه خط را با فرمان `\` و پاراگراف را با خالی گذاشتن یک خط بشکنیم. بنابراین اگر مطلب جدیدی قرار است که شروع شود باید پاراگراف جدید نیز شروع شود. اگر در مورد شکستن پاراگراف مطمئن نیستید، نوشتار را به عنوان حامل ایده‌ها در نظر بگیرید. اگر در نقطه‌ای شکست پاراگراف دارید ولی ایده قبلی هنوز ادامه دارد، شکست را باید از بین ببرید. اگر ایده کاملاً جدیدی در یک خط وارد شده است، آنگاه باید یک شکست پاراگراف داشته باشید.

بعضی از افراد به کلی اهمیت دانستن محل دقیق شکستن پاراگراف‌ها را نمی‌دانند. خیلی از افراد حتی مفهوم شکستن یک پاراگراف را نمی‌دانند، یا، به خصوص در لاتک، پاراگراف‌های جدید ایجاد می‌کنند بدون این که بدانند چنین کاری کرده‌اند. این اشتباه به خصوص اگر در متن فرمول وجود داشته باشد، بیشتر اتفاق می‌افتد. به مثال‌های زیر توجه کنید و سعی کنید دریابید که چرا گاهی اوقات خط خالی (شکست پاراگراف) قبل یا بعد از یک فرمول قرار می‌گیرد و گاهی اوقات قرار نمی‌گیرد. (اگر هنوز همه فرمان‌های این مثال‌ها را متوجه نمی‌شوید، این فصل و فصل بعد را مطالعه کنید و دوباره این بخش را مرور کنید.)

```
% Example 1
\ldots when Einstein introduced his formula
\begin{equation}
e = m \cdot c^2 \ ; \ ,
\end{equation}
which is at the same time the most widely known
and the least well understood physical formula.
```

```
% Example 2
\ldots from which follows Kirchhoff's current law:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

Kirchhoff's voltage law can be derived \ldots

```
% Example 3
\ldots which has several advantages.

\begin{equation}
I_D = I_F - I_R
\end{equation}
is the core of a very different transistor model. \ldots
```

جملات، واحدهای کوچک‌تر متن هستند. در جملات انگلیسی فاصله بعد از یک نقطه پایان خط بیشتر از فاصله بعد از نقطه‌ای است که یک کلمه مخفف را تمام می‌کند. لاتک سعی می‌کند بفهمد کدام یک از این دو مورد نظر است. اگر لاتک اشتباه کرد، باید به او بگویید کدام یک مورد نظر است. روش این‌کار را در ادامه این فصل خواهید دید.

ساختار متن حتی به داخل جملات نیز رسوخ می‌کند. بسیاری از زبان‌ها دارای آیین نگارش پیچیده‌ای هستند، اما در بسیاری از زبان‌ها (مثلاً آلمانی و انگلیسی^۱)، ویرگول را می‌توان با به خاطر سپردن یک اصل ساده در محل درست قرار دهید: در محل‌هایی که متن دارای توقف کوچک است. اگر مطمئن نیستید در کجا ویرگول قرار دهید، جمله را با صدای بلند بخوانید و در هر نقطه‌ای که یک ویرگول دارید یک نفس کوتاه بگیرید. اگر از این کار احساس خوبی نداشتید آن ویرگول را حذف کنید؛ اگر در نقطه‌ای احساس

^۱ مترجم: و صد البته فارسی

نیاز به یک نفس تازه (یا یک توقف کوتاه) داشتید، در آن نقطه یک ویرگول وارد کنید. دست آخر این که پاراگراف‌ها را باید به‌طور منطقی در فصل‌ها، بخش‌ها، زیربخش‌ها، و غیره قرار دهید. با این وجود، تأثیر حروف چینی به صورت

`\section{The Structure of Text and Language}`

آنقدر واضح است که تقریباً مشخص می‌کند این ساختار بندی چگونه انجام می‌شود.

۲.۲ شکستن خط و صفحه

۱.۲.۲ ترازش پاراگراف‌ها

کتاب‌ها معمولاً به این صورت حروف چینی می‌شوند که تمام خط‌ها دارای طول یکسان هستند. لاتک خط‌ها را به صورت مناسب می‌شکند و فاصله مناسب بین کلمات را رعایت می‌کند تا محتوای پاراگراف‌ها را بهینه کند. اگر لازم باشد حتی کلمات را در انتهای خط‌ها می‌شکند. این که پاراگراف‌ها چگونه حروف چینی می‌شوند بستگی به طبقه نوشتار دارد. به طور نرمال اولین خط یک پاراگراف دارای تورفتگی است، و فاصله ویژه‌ای بین پاراگراف‌ها وجود ندارد. برای اطلاعات بیشتر به بخش ۲.۳.۶ مراجعه کنید. در حالات ویژه ممکن است لازم باشد که لاتک را مجبور به شکستن یک خط کنیم. فرمان

`\\ یا \newline`

یک خط جدید بدون شکستن پاراگراف شروع می‌کند. فرمان

`*`

بعلاوه از ایجاد یک شکست صفحه بعد از شکست خط جلوگیری می‌کند. فرمان

`\newpage`

یک صفحه جدید را آغاز می‌کند. فرمان‌های

`\linebreak[n], \nolinebreak[n], \pagebreak[n], \nopagebreak[n]`

جاهایی را پیشنهاد می‌کنند که یک شکست باید یا نباید انجام شود. این فرمان‌ها به نویسنده امکان تغییر پارامتر n را می‌دهند، که می‌تواند عددی بین صفر تا چهار باشد. با انتخاب n کمتر از چهار به لاتک اجازه می‌دهید فرمان شما را در صورت خیلی بد بودن نتیجه اثر ندهد. این فرمان‌های “break” را با فرمان‌های “new” اشتباه نگیرید. حتی موقعی که از فرمان “break” استفاده می‌کنید، لاتک سعی می‌کند طول خط و طول صفحه را گسترش دهد که این مطلب در بخش بعد توضیح داده شده است. این کار ممکن است

فاصله‌های نامطلوب در نوشتار شما ایجاد کند. اگر واقعاً می‌خواهید یک خط جدید یا صفحه جدید را شروع کنید آنگاه از فرمان مربوط به این کارها استفاده کنید. نام این فرمان‌ها را حدس بزنید! لاتک همواره سعی می‌کند بهترین شکست خط‌های ممکن را ایجاد کند. اگر لاتک نتواند خط‌ها را طبق استانداردهای پیشرفته بشکند، این اجازه را به خط می‌دهد که از سمت راست به بیرون پاراگراف کشیده شود. در این حالت لاتک هشدار “overfull hbox” را در زمان پردازش می‌دهد. این اتفاق وقتی رخ می‌دهد که لاتک مکان مناسبی برای شکستن کلمات در انتهای خط پیدا نکند.^۲ می‌توانید استانداردهای لاتک را با فرمان \sloppy کمی پایین بیاورید. این فرمان باعث ایجاد فاصله‌های بین کلمه‌ای طولانی می‌شود حتی اگر خروجی بهینه نباشد. در این حالت لاتک هشدار “underfull hbox” را به کاربر می‌دهد. در اغلب اوقات نتیجه این کار خیلی جالب نیست. فرمان \fussy استانداردهای لاتک را به حالت پیش فرض برمی‌گرداند.

۲.۲.۲ شکستن کلمات

لاتک کلمات را در صورت لزوم می‌شکند. اگر الگوریتم شکستن کلمات نتواند مکان دقیقی برای شکستن کلمه پیدا کند، می‌توانید لاتک را در این راه یاری کنید.

فرمان

```
\hyphenation{word list}
```

باعث می‌شود کلماتی که در لیست آمده است تنها در نقاط با علامت “-” شکسته شود. آرگومان فرمان تنها باید شامل کلماتی باشد که از حروف علامت‌های عادی تشکیل شده باشد. راهنمایی شکستن کلمات برای یک زبان ویژه در حافظه باقی می‌ماند تا آن زبان فعال شود. این بدان معنی است که اگر فرمان شکستن را در سرآغاز نوشتار وارد کنید تنها زبان انگلیسی را مورد نظر قرار می‌دهد. اگر فرمان شکستن را بعد از \begin{document} قرار دهید و از بسته‌ای مانند babel استفاده کنید، آنگاه راهنمایی‌های شکستن کلمات برای زبانی که توسط babel انتخاب شده است فعال می‌شود.

مثال زیر به “hyphenation” اجازه می‌دهد تا همانند “Hyphenation” شکسته شود، و مانع از شکسته شدن “FORTRAN”، “Fortran” و “fortran” می‌شود. هیچ حرف یا نماد ویژه‌ای را نمی‌توان در آرگومان فرمان شکستن قرار داد.

مثال:

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

فرمان - یک پیشنهاد برای شکستن کلمه را ایجاد می‌کند. این نقطه تنها نقطه‌ای می‌شود که کلمه مجاز است در آنجا شکسته شود. این فرمان به ویژه برای کلماتی که دارای حرف ویژه‌ای هستند مفید است (مانند حروف لهجه‌ها)، زیرا لاتک این کلمات را نمی‌تواند به‌طور خودکار بشکند.

^۲ با وجود این که لاتک هشدار در مورد وقوع overfull hbox می‌دهد، معمولاً آسان نیست که خط مورد نظر را پیدا کنیم. اگر از گزینه draft در \documentclass استفاده کنید، در حاشیه سمت راست این خط‌ها یک نشان پهن سیاه ایجاد می‌شود.

I think this is: su\~per\~cal\~%
i\~frag\~i\~lis\~tic\~ex\~pi\~%
al\~i\~do\~cious

I think this is: supercalifragilisticexpialido-
cious

چند کلمه را می‌توان در یک خط با فرمان زیر نگهداشت:

`\mbox{text}`

این فرمان باعث می‌شود آرگومان‌هایش تحت هر شرایطی در کنار هم قرار بگیرند.

My phone number will change soon.
It will be `\mbox{0116 291 2319}`.

My phone number will change soon. It will be
0116 291 2319.

The parameter
`\mbox{\emph{filename}}` should
contain the name of the file.

The parameter *filename* should contain the
name of the file.

`\fbox` مشابه `\mbox` است، با این تفاوت که کادری دور متن قرار می‌گیرد.

۳.۲ رشته‌های تعریف شده

در بعضی از مثال‌های صفحه قبل، یک فرمان خیلی ساده برای حروف‌چینی رشته‌های ویژه را دیدید:

فرمان	مثال	توضیح
<code>\today</code>	November 5, 2014	زمان جاری
<code>\TeX</code>	\TeX	حروف‌چین مورد علاقه شما
<code>\LaTeX</code>	\LaTeX	عنوان بازی
<code>\LaTeXe</code>	$\LaTeX 2_{\epsilon}$	شکل کنونی

۴.۲ حروف و نمادهای ویژه

۱.۴.۲ علامت نقل قول

برای نقل قول نباید مانند ماشین تایپ از " استفاده کنید. برای انتشار از علامت دیگری برای این کار استفاده می‌شود. در لاتک، از دو علامت ^۳ برای شروع نقل قول و از دو علامت ^۴ برای پایان نقل قول استفاده می‌شود. برای نقل قول منفرد از یکی از این علامت‌ها استفاده می‌کنیم.

^۳grave accent

^۴vertical quote

``Please press the `x' key.''

“Please press the ‘x’ key.”

می‌دانم که روش مناسبی نیست که از ‘ برای شروع نقل قول و از ’ برای اتمام آن استفاده کرد.

۲.۴.۲ فاصله کلمات و شکستن

لاتک چهار نوع فاصله بین کلمات را می‌شناسد. سه تا از این فاصله‌ها را می‌توان با نوشتن چند دَش (خط تیره) پشت سر هم تولید کرد. علامت چهارم دَش نیست و در حقیقت همان علامت منهای ریاضی است:

daughter-in-law, X-rated\\
pages 13--67\\
yes---or no? \\
\$0\$, \$1\$ and \$-1\$

daughter-in-law, X-rated
pages 13-67
yes-or no?
٠, ١ and -١

نام این دَش‌ها این است: hyphen ‘-’، en-dash ‘—’، em-dash ‘—’ و minus sign ‘-’.

۳.۴.۲ تیلدا (~)

کاراکتری که معمولاً در صفحات وب ظاهر می‌شود علامت تیلدا است. برای تولید این کاراکتر در لاتک می‌توانید از ~ کمک بگیرید ولی حاصل آن ~ است که دقیقاً آن چیزی نیست که می‌خواهید. به جای آن از روش زیر استفاده کنید:

http://www.rich.edu/~bush \\
http://www.clever.edu/~demo

http://www.rich.edu/~bush
http://www.clever.edu/~demo

۴.۴.۲ اسلش (/)

برای نمایش اسلش بین دو واژه، می‌توان به‌سادگی نوشت، برای نمونه read/write، ولی این کار باعث می‌شود لاتک هر دو واژه را چسبیده نمایش دهد. شکست واژه‌ها برای این دو واژه ناپویا می‌شود و ممکن است خطای «سرریز» دریافت کنید. برای رفع این مشکل slash را بکار ببرید. برای نمونه بنویسید ‘read/slash write’ که شکست کلمه‌ها را ممکن می‌کند. ولی ‘/’ نرمال هنوز می‌تواند برای کسر و واحدها بکار رود، ب.ن. 5 MB/s.

۵.۴.۲ علامت درجه (°)

مثال زیر نشان می‌دهد چگونه می‌توان علامت درجه را در لاتک نوشت:

It's $\$-30\text{\textcircled{C}}$.
I will soon start to
super-conduct.

It's $-30\text{ }^\circ\text{C}$. I will soon start to super-conduct.

بسته `textcomp` علامت درجه را با فرمان `\textcelsius` نیز قابل دسترسی می‌کند.

۶.۴.۲ نماد واحد پول اروپا (€)

این روزها نماد واحد پول اروپا بسیار به‌کار می‌رود. بیشتر قلم‌های کنونی دارای کاراکتر ویژه برای این نماد هستند. بعد از فراخوانی بسته `textcomp` در سرآغاز نوشتار

```
\usepackage{textcomp}
```

از فرمان

```
\texteuro
```

برای نمایش این کاراکتر می‌توانید استفاده کنید. اگر قلم شما این نماد را ندارد یا از شکل آن خوشتان نمی‌آید، کارهای دیگری می‌توانید انجام دهید. ابتدا این که بسته `eurosym` نماد رسمی واحد پول اروپا را فراهم می‌کند:

```
\usepackage[official]{eurosym}
```

اگر نمادی را می‌پسندید که با قلم شما هم‌خوانی داشته باشد، از گزینه `gen` به جای `official` استفاده کنید.

جدول ۱.۲: کیسه‌ای پر از نماد اروپا

LM+textcomp	<code>\texteuro</code>	€	€	€
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€

۷.۴.۲ سه نقطه (...)

بر روی ماشین تایپ، یک ویرگول یا یک فاصله دارای همان طول یک حرف هستند. در یک کتاب این کاراکترها تنها فضای کوچکی را اشغال می‌کنند. بنابراین سه نقطه را نمی‌توان تنها با نوشتن سه نقطه

نشان داد. برای این منظور فرمان ویژه‌ای وجود دارد:

```
\ldots
```

```
Not like this ... but like this:\\
New York, Tokyo, Budapest, \ldots
```

```
Not like this ... but like this:
New York, Tokyo, Budapest, ...
```

۸.۴.۲ چسبیدگی حروف

بعضی از کلمات تنها با قراردادن متوالی چند حرف بدست نمی‌آیند بلکه باید نمادهای ویژه‌ای برای نمایش آنها به کار برد.

ff fi fl ffi... به جای ff fi fl ffi ...

چسبیدگی حروف را می‌توان با قراردادن یک `\mbox{}` بین دو حرف مورد نظر از بین برد. این کار به عنوان مثال برای کلمه‌هایی لازم است که از ترکیب دو کلمه بدست می‌آیند.

```
\Large Not shelfful\\
but shelf\mbox{ }ful
```

```
Not shelfful
but shelfful
```

۹.۴.۲ لهجه‌ها و حروف ویژه

لاتک استفاده از لهجه‌ها و حروف ویژه را به شکل‌های مختلف پشتیبانی می‌کند. جدول ۲.۲ تمام لهجه‌های مختلف را نشان می‌دهد که بر حرف o قرار می‌گیرند. این کار برای حروف دیگر هم قابل انجام است. برای قراردادن یک لهجه بر روی حرفی مانند i یا z ابتدا باید نقطه روی آن را حذف کرد. برای انجام این کار از `\i` و `\j` استفاده کنید.

```
H\^otel, na\"i ve, \'el`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{ }
Stra\ss e
```

```
Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße
```

۵.۲ پشتیبانی زبان‌های دیگر

۱.۵.۲ گزینه یونیکد

اگر می‌خواهید نوشتاری از چندین زبان بنویسید، یونیکد راه آن است، به ویژه اگر این زبان‌ها از خط لاتین استفاده نکنند. دو موتور تک هستند که توانایی پردازش ورودی‌های یونیکد را دارند:

زیلاتک برای سیستم مک گسترش یافت اما اکنون برای همه سیستم‌ها در دسترس است. نخستین بار در تک‌لایو ۲۰۰۷ ارائه شد.

لواتک نسخه آینده پی‌دی‌اف‌تک است. نخستین بار در تک‌لایو ۲۰۰۸ ارائه شد.

نوشته زیر زیلاتک ارائه شده در تک‌لایو ۲۰۱۰ را توضیح می‌دهد.

زبان‌ها راست به چپ

بسته xepersian از زبان پارسی پشتیبانی می‌کند. این بسته دستورهای فارسی لاتک را فراهم می‌کند که به این ترتیب می‌توانید دستورهایی مانند \section را به پارسی بکار ببرید، که این ویژگی برای پارسی زبانان جذاب است. xepersian تنها بسته‌ای است که از کشیدگی حرف‌ها پشتیبانی می‌کند. بسته همانندی برای زبان سریانی در حال گسترش است.

قلم نستعلیق، که توسط شورای عالی اطلاع رسانی درست شده است، از تارنمای آنها در دسترس است

<http://www.scict.ir/Portal/Home/Default.aspx>.

جدول ۲.۲: لهجه‌ها و حروف ویژه

ò	\`o	ó	\'o	ô	\^o	õ	\~o
ō	\=o	ò	\.o	ö	\"o	ç	\c c
ö	\u o	ö	\v o	ő	\H o	q	\c o
q	\d o	o	\b o	öo	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ł	\l	Ł	\L
ı	\i	ı	\j	ı	!\`	ı	?`

۶.۲ فاصله بین کلمات

برای این که در خروجی، حاشیه سمت راست به صورت منظم ظاهر شود، لاتک فاصله مناسب بین کلمات ایجاد می‌کند تا خط را پر کنند. همچنین لاتک فاصله بیشتری را در انتهای یک خط قرار می‌دهد، زیرا این کار باعث خوانایی بهتر متن می‌شود. لاتک فرض می‌کند انتهای یک جمله نقطه، علامت سؤال یا تعجب است. اگر یک نقطه بعد از یک حرف بزرگ ظاهر شود، لاتک این نقطه را پایان یک خط نمی‌داند، زیرا معمولاً بعد از اسامی ویژه که با حروف بزرگ نوشته می‌شوند یک نقطه قرار می‌گیرد.

هر فرض دیگری به غیر از اینها را نویسنده باید به لاتک اطلاع دهد. یک بک اسلش در جلوی یک فاصله، فاصله‌ای را تولید می‌کند که نمی‌تواند گسترش یابد. حرف تیلدا فاصله‌ای را تولید می‌کند که نمی‌تواند گسترش یابد و به علاوه از شکستن خط جلوگیری می‌کند. فرمان @ در جلوی یک نقطه بیان می‌کند که این نقطه انتهای یک خط است، حتی اگر این نقطه بعد از یک حرف بزرگ ظاهر شده باشد.

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

```
Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?
```

فاصله اضافی بعد از نقطه را می‌توان با فرمان زیر غیر فعال کرد

```
\frenchspacing
```

که به لاتک می‌گوید بعد از نقطه فاصله‌ای بیشتر از فاصله بین کلمات قرار ندهد. این کار در اکثر زبان‌ها معمول است، به جز در هنگام نوشتن کتاب‌نامه. اگر از فرمان `\frenchspacing` استفاده کنید، فراخوانی فرمان `\@` لازم نیست.

۷.۲ عنوان، فصل، و بخش

برای این که خواننده را به هنگام خواندن کار شما راهنمایی کنید، باید نوشتار خود را به فصل‌ها، بخش‌ها، و زیربخش‌ها تقسیم کنید. لاتک این کار را با اختصاص فرمان‌های ویژه‌ای امکان‌پذیر می‌کند که عنوان هر بخش را به عنوان آرگومان می‌پذیرند. این وظیفه شماست که ترتیب آنها را درست بیان کنید. فرمان‌های زیر در طبقه `article` موجودند:

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

اگر می‌خواهید نوشتار را به قسمت‌هایی تقسیم کنید که شماره‌گذاری بخش‌ها و فصل‌ها را تغییر

ندهد از فرمان

`\part{...}`

استفاده کنید.

وقتی که از طبقه‌های `report` و `book` استفاده می‌کنید، فرمان

`\chapter{...}`

هم قابل استفاده است که هر فصل در برگیرنده چندین بخش می‌تواند باشد.

از آنجا که طبقه `article` فرمان `chapter` را نمی‌شناسد، قرار دادن یک مقاله به عنوان یک فصل از یک کتاب بسیار آسان است. فاصله بین بخش‌ها، و شماره‌گذاری آنها و همچنین اندازه قلم عنوان‌ها به طور خودکار توسط لاتک تعیین می‌شود.

دو فرمان از این دسته فرمان‌ها دارای ویژگی‌هایی هستند که در زیر به آنها اشاره شده است:

- فرمان `\part` شماره‌گذاری مسلسل فصل‌ها را تغییر نمی‌دهد.
- فرمان `\appendix` هیچ آرگومانی را نمی‌پذیرد. این فرمان تنها شماره‌گذاری فصل‌ها را به صورت حرفی تغییر می‌دهد.^۵

لاتک فهرست مطالب را با قراردادن عنوان بخش‌ها و صفحه مربوط به آنها که از آخرین پردازش بدست آمده است تولید می‌کند. فرمان

`\tableofcontents`

هر جا که ظاهر شود باعث نمایش فهرست مطالب در همان نقطه می‌شود. یک نوشتار جدید باید دوبار پردازش شود تا `\tableofcontents` به صورت درست درج گردد. گاهی اوقات لازم است فایل را سه‌بار پردازش کنید، لاتک در این مورد به شما پیغام مناسب را می‌دهد.

تمام فرمان‌های بخش‌بندی که در بالا ذکر شد دارای حالت ستاره‌دار نیز می‌باشند. حالت ستاره‌دار این فرمان‌ها به راحتی با افزودن یک علامت * به انتهای نام فرمان درست می‌شود. این فرمان‌ها باعث تولید بخش مربوطه می‌شوند با این تفاوت که شماره‌دار نیستند و در فهرست مطالب ظاهر نمی‌شوند. برای این کار، به عنوان مثال به جای فرمان `\section{Help}` باید از فرمان `\section*{Help}` استفاده کنید.

عنوان بخش‌ها به طور نرمال در فهرست مطالب ظاهر می‌شوند. گاهی اوقات این کار امکان‌پذیر نیست زیرا عنوان بخش طولانی است و در یک خط جا نمی‌شود. در این صورت می‌توان عنوانی را که در فهرست مطالب ظاهر می‌شود با یک گزینه انتخابی در جلوی عنوان واقعی تعیین کرد.

^۵در طبقه مقاله، این فرمان شماره‌گذاری بخش‌ها را حرفی می‌کند.

```
\chapter[Title for the table of contents]{A long
and especially boring title, shown in the text}
```

عنوان کلی نوشتار با فرمان

```
\maketitle
```

چاپ می‌شود. محتویات عنوان نوشتار را می‌توان با فرمان‌های زیر قبل از فرمان `\maketitle` تعیین کرد:

```
\title{...}, \author{...}, \date{...}
```

در آرگومان فرمان `\author` می‌توانید چندین نام را وارد کنید که با فرمان `\and` از یکدیگر جدا می‌شوند. مثالی از فرمان‌هایی را که در بالا معرفی کردیم می‌توانید در جدول ۲.۱ در صفحه ۷ ببینید. علاوه بر فرمان‌های بخش‌بندی که در بالا اشاره شد، لاتک سه فرمان دیگر به همراه طبقه `book` ارائه می‌کند. این فرمان‌ها برای تقسیم نوشتار به کار می‌آیند. این فرمان‌ها سربرگ و شماره صفحه را در یک کتاب تغییر می‌دهند:

`\frontmatter` باید اولین فرمان بعد از شروع متن نوشتار باشد (`\begin{document}`). این فرمان شماره صفحه‌ها را به اعداد لاتین تغییر می‌دهد و بخش‌ها را بدون شماره ظاهر می‌کند. رفتار این فرمان روی بخش‌بندی‌ها همانند این است که از فرمان‌های بخش‌بندی ستاره‌دار استفاده کنید (به عنوان مثال `\chapter*{Preface}`) با این تفاوت که عنوان این بخش‌ها همچنان در فهرست مطالب ظاهر می‌شوند.

`\mainmatter` این فرمان دقیقاً قبل از اعلان اولین فصل به کار می‌رود که باعث می‌شود شماره صفحه به سبک عددی تغییر یابد و آن را از یک شروع می‌کند.

`\appendix` پیوست‌های نوشتار را شروع می‌کند. بعد از این فرمان، فصل‌ها با حروف شماره‌گذاری می‌شوند.

`\backmatter` باید قبل از آخرین آیتم کتاب، مانند کتاب‌نامه و نمایه ظاهر شود. در یک طبقه استاندارد، این فرمان هیچ تاثیری ندارد.

۸.۲ ارجاع

در کتاب‌ها، گزارش‌ها، و مقالات معمولاً ارجاع‌هایی مانند شکل‌ها، جدول‌ها و قسمت‌های ویژه از متن وجود دارد که به آنها ارجاع‌های متنی^۶ می‌گویند. لاتک فرمان‌های زیر را برای تولید ارجاع‌های متنی ارائه

^۶cross-references

می‌کند

```
\label{marker}, \ref{marker} و \pageref{marker}
```

که *marker* یک نشانگر است که توسط کاربر انتخاب می‌شود. لاتک تمام فرمان‌های `\ref` را با شماره بخش، زیربخش، شکل، جدول، یا قضیه‌ای نمایش می‌دهد که فرمان `\label` در آن ظاهر شده است. فرمان `\pageref` شماره صفحه‌ای را نمایش می‌دهد که `\label` مورد نظر قرار دارد.^۷

A reference to this subsection
`\label{sec:this}` looks like:
```see section-\ref{sec:this} on  
page-\pageref{sec:this}.''`

A reference to this subsection looks like: "see  
section 8.2 on page 29."

## ۹.۲ پانویس

با فرمان

```
\footnote{footnote text}
```

پانویس در انتهای صفحه جاری نوشته می‌شود. پانویس‌ها همواره باید بعد از کلمه یا جمله‌ای قرار داده شود<sup>۸</sup> که به آن اشاره می‌کند. بنابراین پانویسی که به کل یک عبارت اشاره می‌کند باید بعد از ویرگول یا نقطه انتهای آن جمله قرار داده شود. با توجه به این که هر کسی که نوشتار را می‌خواند نهایتاً پانویس‌ها را هم مطالعه می‌کند (زیرا که ما موجودات کنجکاوی هستیم) پس چرا تمام مطالب را در خود متن بیان نکنیم؟<sup>۹</sup>

Footnotes\footnote{This is  
a footnote.} are often used  
by people using \LaTeX.

Footnotes<sup>a</sup> are often used by people using  
 $\text{\LaTeX}$ .

<sup>a</sup>This is a footnote.

<sup>۷</sup> توجه داشته باشید که این فرمان‌ها از محتوای چیزی که به آن ارجاع می‌کنند اطلاعی ندارند. فرمان `\label` تنها آخرین شماره تولید شده را ذخیره می‌کند. وقتی که این شماره، شماره یک بخش باشد شماره مورد نظر از پردازش قبل را ذخیره می‌کند.  
<sup>۸</sup> فعل شدن یکی از افعال معمول فارسی است.  
<sup>۹</sup> تو که لالایی بلدی پس چرا خوابت نمی‌برد:-)

## ۱۰.۲ تاکید کلمات

اگر با یک ماشین تایپ متنی را بنویسید، کلمات مهم به صورت زیرخط تایپ می‌شوند.

```
\underline{text}
```

در کتاب‌های تایپ شده، کلمه‌های مهم را به صورت ایتالیک نمایش می‌دهند. لاتک فرمان

```
\emph{text}
```

را برای تأکید کلمه‌ها به کار می‌برد. تأثیر فرمان به متن بستگی دارد:

```
\emph{If you use
emphasizing inside a piece
of emphasized text, then
\LaTeX{} uses the
\emph{normal} font for
emphasizing.}
```

*If you use emphasizing inside a piece of emphasized text, then  $\text{\LaTeX}$  uses the normal font for emphasizing.*

لطفاً به تفاوت این که لاتک چیزی را تأکید کند و یا این که از قلم دیگری استفاده کنیم توجه کنید.

```
\textit{You can also
\emph{emphasize} text if
it is set in italics,}
\textsf{in a
\emph{sans-serif} font,}
\texttt{or in
\emph{typewriter} style.}
```

*You can also emphasize text if it is set in italics, in a sans-serif font, or in typewriter style.*

## ۱۱.۲ محیط‌ها

لاتک محیط‌های مختلفی را برای کارهای مختلف ارائه می‌کند:

```
\begin{environment} text \end{environment}
```

که *environment* نام محیطی است که مورد استفاده قرار می‌گیرد. محیط‌ها می‌توانند تودرتو باشند، مادامی که ترتیب درست آنها اعمال شده باشد.

```
\begin{aaa}... \begin{bbb}... \end{bbb}... \end{aaa}
```

در بخش بعد انواع محیط‌ها را مورد بررسی قرار می‌دهیم.

## ۱.۱۱.۲ محیط‌های تبصره، توضیح، و شماره‌دار

محیط `itemize` برای تولید لیست‌های ساده مفید است، `enumerate` برای تولید لیست‌های شماره‌دار، و `description` برای محیط توضیحات مفید است.

```
\flushleft
\begin{enumerate}
\item You can mix the list
environments to your taste:
\begin{itemize}
\item But it might start to
look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not
become smart because they are
in a list.
\item[Smart] things, though,
can be presented beautifully
in a list.
\end{description}
\end{enumerate}
```

1. You can mix the list environments to your taste:
  - But it might start to look silly.
  - With a dash.
2. Therefore remember:

**Stupid** things will not become smart because they are in a list.

**Smart** things, though, can be presented beautifully in a list.

## ۲.۱۱.۲ چپ، راست، و وسط چین

محیط‌های `flushright` و `flushleft` پاراگراف‌هایی را تولید می‌کنند که چپ‌چین یا راست‌چین هستند. محیط `center` متن را وسط‌چین می‌نویسد. اگر شکست خط را با فرمان `\\` اعلان نکنید، لاتک به صورت خودکار شکست خط‌ها را تعیین می‌کند.

```
\begin{flushleft}
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushleft}
```

This text is left-aligned.  $\LaTeX$  is not trying to make each line the same length.

```
\begin{flushright}
This text is right-\\aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushright}
```

This text is right-aligned.  $\LaTeX$  is not trying to make each line the same length.

```
\begin{center}
At the centre\\of the earth
\end{center}
```

At the centre  
of the earth

### ۳.۱۱.۲ نقل قول و شعر

محیط quote برای عبارت‌های نقل قول و مثال‌ها مفید است.

```
A typographical rule of thumb
for the line length is:
\begin{quote}
On average, no line should
be longer than 66 characters.
\end{quote}
This is why \LaTeX{} pages have
such large borders by default
and also why multicolumn print
is used in newspapers.
```

A typographical rule of thumb for the line length is:

On average, no line should be longer than 66 characters.

This is why  $\LaTeX$  pages have such large borders by default and also why multicolumn print is used in newspapers.

دو محیط مشابه دیگر وجود دارد: محیط quotation و verse. محیط quotation برای نقل قول‌های طولانی که بیش از یک پاراگراف باشند مفید است. محیط verse برای نگارش شعر مفید است که شکست‌ها خیلی مهم هستند. در این محیط شکست‌ها با فرمان \\ در انتهای خط مورد نظر و یک خط خالی بعد از هر قطعه انجام می‌گیرد.

```
I know only one English poem by
heart. It is about Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\
Humpty Dumpty had a great fall.\
All the King's horses and all
the King's men\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

I know only one English poem by heart. It is about Humpty Dumpty.

Humpty Dumpty sat on a wall:  
Humpty Dumpty had a great  
fall.  
All the King's horses and all the  
King's men  
Couldn't put Humpty together  
again.

#### ۴.۱۱.۲ مقدمه

در مطالب علمی معمولاً نوشتار را با یک چکیده شروع می‌کنند. لاتک محیط abstract را برای انجام چنین کاری پیش‌بینی کرده است. به طور نرمال یک چکیده در مقالات به کار می‌رود.

```
\begin{abstract}
The abstract abstract.
\end{abstract}
```

The abstract abstract.

#### ۵.۱۱.۲ چاپ تحت‌اللفظ

متن‌هایی که بین `\begin{verbatim}` و `\end{verbatim}` نوشته می‌شوند، همانند این که با ماشین تایپ نوشته شده باشند ظاهر می‌شوند، با تمام شکست خط‌ها و بدون تأثیر هیچ فرمان لاتک. برای یک پاراگراف این کار را می‌توان به صورت زیر انجام داد.

```
\verb+text+
```

+ تنها یک مثال از یک کاراکتر حائل است. بسیاری از مثال‌های این مقدمه به کمک همین محیط نوشته شده‌اند.

The `\verb|\ldots|` command `\ldots`

```
\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

The `\ldots` command ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
the starred version of
the verbatim
environment emphasizes
the spaces in the text
\end{verbatim*}
```

```
the_starred_version_of
the_verbatim
environment_emphasizes
the_spaces_in_the_text
```

فرمان `\verb` را می‌توان به صورت ستاره‌دار به کار برد:

```
\verb*|like this :-)|
```

```
like_this_:-)|
```

محیط `verbatim` و فرمان `\verb` را نمی‌توان به صورت پارامتر فرمان‌های دیگر به کار برد.

## ۶.۱۱.۲ جدول

محیط `tabular` را می‌توان برای طراحی جدول‌های زیبا با خط‌های افقی و عمودی به کار برد. لاتک عرض ستون‌ها را به صورت خودکار تشخیص می‌دهد. آرگومان `table spec` از فرمان

```
\begin{tabular}[pos]{table spec}
```

سبک جدول را تعریف می‌کند. از `l` برای یک ستون چپ‌چین، `r` برای راست‌چین، `c` برای وسط‌چین استفاده کنید؛ از `p{width}` برای یک ستون شامل یک متن چیده شده با شکست خط، و `l` برای یک خط عمودی استفاده کنید.

اگر متن درون یک ستون گسترده‌تر از صفحه باشد، لاتک آن را به طور خودکار نمی‌شکند. با استفاده از فرمان `p{width}` می‌توانید نوع ویژه‌ای از ستون را تعریف کنید که پیرامون یک متن مشخص شده گرد شده است.

آرگومان `pos` مکان عمودی جدول را نسبت به خط کرسی متنی دور آن تعیین می‌کند. از یکی از گزینه‌های `c`، `b`، `t` برای تعیین این مقدار به بالا، پایین و وسط استفاده کنید.

در یک محیط `tabular`، با درج `&` به ستون بعد می‌رویم و `\\` یک خط جدید را شروع می‌کند و `\hline` یک خط افقی رسم می‌کند. می‌توانید خط را از ستون `j`-ام تا ستون `i`-ام با فرمان `\cline{j-i}` رسم کنید.



```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

|             |             |
|-------------|-------------|
| 7C0         | hexadecimal |
| 3700        | octal       |
| 11111000000 | binary      |
| 1984        | decimal     |

```
\begin{tabular}{|p{4.7cm}|}
\hline
Welcome to Boxy's paragraph.
We sincerely hope you'll
all enjoy the show. \\
\hline
\end{tabular}
```

|                                                                                          |
|------------------------------------------------------------------------------------------|
| <p>Welcome to Boxy's paragraph.<br/>We sincerely hope you'll all enjoy<br/>the show.</p> |
|------------------------------------------------------------------------------------------|

جداکننده ستون‌ها را می‌توان با `@{...}` ساخت. این فرمان فاصله بین ستون‌ها را از بین می‌برد و به جای آن از چیزی استفاده می‌کند که در آکولاد ارائه کرده‌اید. مورد معمول استفاده از این فرمان در چین بر اساس ممیز است. کاربرد دیگر آن از بین بردن فاصله بالایی یک جدول با استفاده از فرمان `@{}` است.

```
\begin{tabular}{@{} l @{}}
\hline
no leading space \\
\hline
\end{tabular}
```

|                         |
|-------------------------|
| <p>no leading space</p> |
|-------------------------|

```
\begin{tabular}{l}
\hline
leading space left and right \\
\hline
\end{tabular}
```

|                                     |
|-------------------------------------|
| <p>leading space left and right</p> |
|-------------------------------------|

از آنجا که هیچ راه درونی برای مرتب کردن اعداد در یک جدول به صورت ممیزچین وجود ندارد<sup>۱۰</sup> این کار را می‌توان با یک حقه و داشتن دو ستون انجام داد: یکی به صورت راست‌چین، و دیگری به صورت عدد اعشاری چپ‌چین. فرمان `@{.}` در خط‌های محیط `\begin{tabular}` فاصله عادی

<sup>۱۰</sup> اگر کلاف ابزار روی سیستم شما نصب است، نگاهی به بسته `dcolumn` بیندازید.

بین ستون‌ها را تنها با یک نقطه نشان می‌دهد که نماد معمولی ممیز است. فراموش نکنید که باید قسمت اعشاری عددتان را با فرمان & از قسمت درست آن جدا کنید. برچسب یک ستون را می‌توان با فرمان `\multicolumn` تعیین کنید.

```
\begin{tabular}{c r @{.} l}
Pi expression &
\multicolumn{2}{c}{Value} \\
\hline
π & 3&1416 \\
π^{π} & 36&46 \\
$\pi^{\pi^{\pi}}$ & 80662&7 \\
\end{tabular}
```

| Pi expression       | Value   |
|---------------------|---------|
| $\pi$               | 3.1416  |
| $\pi^{\pi}$         | 36.46   |
| $(\pi^{\pi})^{\pi}$ | 80662.7 |

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

|      |      |
|------|------|
| Ene  |      |
| Mene | Muh! |

تمام متن یک جدول همواره در یک صفحه قرار می‌گیرد. اگر می‌خواهید جدول‌های بزرگتری را طراحی کنید، باید از محیط `longtable` استفاده کنید.

## ۱۲.۲ اجسام شناور

امروزه بسیاری از چیزهایی که به چاپ می‌رسند دارای تعداد زیادی جدول و شکل هستند. این اشیاء به حفاظت بیشتری احتیاج دارند، زیرا نمی‌توانند بین صفحه‌ها شکسته شوند. یک روش برای این کار این است که هرگاه یک جدول یا شکل آنقدر بزرگ باشد که در ادامهٔ صفحه جا نگیرد، آنگاه یک صفحهٔ جدید برای نمایش آن تولید شود. این کار باعث می‌شود که تعدادی از صفحات خالی باشند که بسیار بد منظره است.

راه حل این مشکل این است که شکل‌ها و جدول‌هایی را که در صفحه نمی‌گنجند به ابتدای صفحهٔ بعد منتقل کنیم، و ادامهٔ صفحهٔ اول را با متن پرکنیم. لاتک دو محیط برای حفاظت این گونه اجسام شناور تعبیه کرده است؛ یکی برای جدول و یکی برای شکل. برای استفادهٔ بهینه از این دو محیط باید به طور تقریبی بدانید لاتک در درون خودش با اجسام شناور چگونه رفتار می‌کند. در غیر این صورت این موضوع یک معضل برای شما می‌شود زیرا لاتک هیچگاه این اجسام را در نقطه‌ای که شما می‌خواهید قرار نمی‌دهد.

ابتدا اجازه دهید به فرمان‌هایی که برای اجسام شناور تعبیه شده‌اند نظری بیندازیم:

هر چیزی که در میان محیط `figure` و `table` قرار می‌گیرد به عنوان یک شیء شناور منظور می‌شود. هر دو محیط شناور

`\begin{figure}[placement specifier]` یا `\begin{table}[...]`

پارامترهای اختیاری قبول می‌کنند که به آن مشخص کننده مکان<sup>۱۱</sup> می‌گوییم. این پارامتر برای نشان دادن مکان مورد نظر برای جسم شناور به کار می‌رود. این پارامتر به صورت یک رشته از مکان‌های ممکن تعیین می‌شود. جدول ۳.۲ را ببینید. یک جدول را می‌توان به صورت زیر تولید کرد:

`\begin{table}[!hbp]`

مشخص کننده مکان `[!hbp]` به لاتک اجازه می‌دهد که جدول را در همان نقطه یا در پایین صفحه و یا در یک صفحه شامل تنها اشیاء شناور قرار دهد، و یا حتی در هر کدام که ممکن است با وجود این که ممکن است حاصل کار زیبا نباشد. اگر هیچ مکانی معرفی نگردد مقدار پیش فرض آن `[tbp]` است. لاتک هر جسم شناور را همان جایی که کاربر فرمان داده است قرار می‌دهد. اگر این کار در صفحه جاری امکان پذیر نباشد، لاتک آن را به صف نوع جسم شناور انتقال می‌دهد.<sup>۱۲</sup> هرگاه یک صفحه جدید شروع می‌شود، لاتک ابتدا بررسی می‌کند که آیا جسم شناوری در صف انتظار برای الصاق موجود است. اگر این کار امکان پذیر نباشد، با هر جسم در صف مربوط به خودش به ترتیبی رفتار می‌شود که انگار در همین نقطه از متن طبق راهنمایی نویسنده قرار است قرار داده شود (به جز `h` که دیگر مورد نظر قرار نمی‌گیرد). هر جسم دیگر در متن به مکان مناسب در صف مربوطه انتقال می‌یابد. لاتک به طور منظم ترتیب اولیه هر جسم در صف را مد نظر قرار می‌دهد. به همین دلیل است که اگر شکلی قابل ظاهر شدن در متن نباشد به انتهای نوشتار انتقال داده می‌شود و بنابراین تمام شکل‌های بعد از آن نیز به انتهای نوشتار انتقال می‌یابند. بنابراین:

<sup>۱۱</sup>placement specifier

<sup>۱۲</sup>این صفاها به شکل اولین ورودی — اولین خروجی ظاهر می‌شوند!

جدول ۳.۲: پارامترهای قراردادن اجسام شناور

| Spec | اجازه قرار دادن جسم ...                                                                    |
|------|--------------------------------------------------------------------------------------------|
| h    | اینجا ( <i>here</i> ) در همان جایی از متن که فرمان ظاهر شده است. برای اجسام کوچک مفید است. |
| t    | در بالای ( <i>top</i> ) صفحه.                                                              |
| b    | در پایین ( <i>bottom</i> ) صفحه.                                                           |
| p    | در یک صفحه ویژه که تنها شامل اجسام شناور است.                                              |
| !    | بدون در نظر گرفتن بسیاری از پارامترهای داخلی <sup>آ</sup>                                  |

<sup>آ</sup>مانند ماکسیم تعداد اشیاء شناور در یک صفحه

اگر لاتک اجسام شناور را آن طور که شما می‌خواهید قرار نمی‌دهد اغلب به این دلیل است که تنها یکی از این اجسام را نمی‌تواند در هیچ نقطه‌ای از متن قرار دهد.

وقتی که تنها یک مکان مناسب برای جسم وجود داشته باشد، این موضوع ممکن است مشکل‌ساز شود. اگر جسمی در مکان پیشنهاد شده قابل نمایش نباشد، معمولاً یک مشکل از این نوع پدید می‌آید. به خصوص این که هیچ‌گاه نباید از گزینه [h] استفاده کنید، این کار آنقدر مشکل‌ساز است که در نسخه‌های جدید لاتک این گزینه به طور خودکار به [ht] تبدیل می‌شود. حال که مشکلات محیط‌های جدول و شکل را کمی توضیح دادیم، چند موضوع دیگر نیز نیاز به توضیح بیشتر دارند. با فرمان

```
\caption{caption text}
```

می‌توانید عنوان یک جسم شناور را تعریف کنید. یک شماره و یک عنوان شکل یا جدول به طور خودکار توسط لاتک قبل از این عنوان قرار می‌گیرد.  
دو فرمان

```
\listoffigures و \listoftables
```

همانند فرمان `\tableofcontents` لیست جدول‌ها و شکل‌ها را چاپ می‌کند. این لیست‌ها عنوان کامل شیء مورد نظر را نمایش می‌دهند، بنابراین اگر عنوان این شکل‌ها طولانی است، باید عنوان کوچکتری را به عنوان گزینه اختیاری معرفی کنید. این کار به صورت زیر امکان‌پذیر است.

```
\caption[Short]{LLLLLoooooonnnnnnggggg}
```

با فرمان `\label` و `\ref` می‌توانید ارجاعی به این اجسام شناور داشته باشید. توجه داشته باشید که فرمان `\label` باید بعد از فرمان `\caption` قرار بگیرد زیرا باید شماره مربوطه با این فرمان دوم تولید شده باشد.

مثال زیر مربعی را رسم می‌کند و آن را در متن قرار می‌دهد. می‌توانید از این کار برای اختصاص یک تصویر با ابعاد مشخص در پایان کار استفاده کنید.

```
Figure~\ref{white} is an example of Pop-Art.
\begin{figure}[!hbt]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Five by Five in Centimetres.\label{white}}[A
\end{figure}
```

در مثال بالا، لاتک به سختی (!) سعی می‌کند تا شکل را دقیقاً در همین نقطه از متن قرار دهد.<sup>۱۳</sup> اگر این کار امکان‌پذیر نباشد سعی می‌کند شکل را در انتهای صفحه قرار دهد. اگر هیچ‌کدام از این کارها امکان‌پذیر

<sup>۱۳</sup> فرض کنید صف مربوط به شکل‌ها خالی باشد.

نباشد، لاتک بررسی می‌کند که آیا می‌تواند شکل را در یک صفحه خالی به همراه مثلاً یک جدول قرار دهد. اگر محتویات لازم برای پرکردن یک صفحه شناور موجود نباشد، لاتک یک صفحه جدید تولید می‌کند و یک بار دیگر همین مراحل را از سر می‌گیرد. تحت شرایط ویژه‌ای اگر لازم باشد از فرمان

`\clearpage` یا `\cleardoublepage`

استفاده کنید. این فرمان لاتک را مجبور می‌کند تا تمام اشیاء باقیمانده در صف را قرار دهد و یک صفحه جدید تولید کند. فرمان `\cleardoublepage` به صفحه سمت راست بعدی می‌رود. بعداً در این مقدمه یاد خواهید گرفت چگونه شکل‌های پست‌اسکرپت را در متن خود قرار دهید.

## ۱۳.۲ حفاظت از اجسام شکستنی

متنی که توسط فرمان‌های `\caption` و `\section` در متن ظاهر می‌شود ممکن است در نوشتار چندین بار تکرار شود (به عنوان مثال در فهرست مطالب یا متن نوشتار). بعضی از فرمان‌ها هنگام استفاده در درون فرمان‌هایی مانند `\section` ممکن است شکسته شوند و پردازش فایل میسر نباشد. این فرمان‌ها را فرمان‌های شکستنی می‌نامند، به عنوان مثال `\footnote` و `\phantom`. این فرمان‌های شکستنی احتیاج به حفاظت دارند (ما چطور!). می‌توانید آنها را با فرمان `\protect` در جلوی آنها مورد حفاظت قرار دهیم.

`\protect` تنها بر فرمانی که بعد از آن ظاهر می‌شود اثر دارد، و حتی بر پارامترهای آن تاثیری ندارد. در بیشتر مواقع یک فرمان اضافی `\protect` هیچ ضرری ندارد.

```
\section{I am considerate}
\protect\footnote{and protect my footnotes}}
```



## فصل ۳

# حروف چینی فرمول‌های ریاضی

حال آماده هستید! در این فصل به قویترین قسمت تک، حروف چینی ریاضی، حمله می‌کنیم. اما توجه داشته باشید، این فصل فقط سطح کار را صیقل می‌دهد. با وجود این که مطالب این فصل برای بسیاری از افراد کافی است، اگر نتوانستید در آن پاسخ بعضی از نیازهای حروف چینی ریاضی خود را بیابید ناامید نشوید. به احتمال بسیار زیاد جواب شما در  $\text{AMS-LATEX}$  داده شده است.

### ۱.۳ کلاف $\text{AMS-LATEX}$

اگر می‌خواهید حروف چینی (پیشرفته) ریاضی انجام دهید، باید از کلاف  $\text{AMS-LATEX}$  استفاده کنید. کلاف  $\text{AMS-LATEX}$  مجموعه‌ای از بسته‌ها و طبقه‌ها برای حروف چینی ریاضی است. ما بیشتر به بررسی بسته  $\text{amsmath}$  می‌پردازیم که جزیی از این کلاف است.  $\text{AMS-LATEX}$  توسط انجمن ریاضی آمریکا تولید شده است و به‌طور گسترده برای حروف چینی ریاضی مورد استفاده قرار می‌گیرد. خود لاتک دارای محیط‌هایی ابتدایی برای ریاضی است، اما این محیط‌ها محدود هستند (یا برعکس:  $\text{AMS-LATEX}$  نامحدود است) و در بعضی حالات ناپایدار نیز هستند.

$\text{AMS-LATEX}$  جزیی از توزیع مورد نیاز است و توسط تمام توزیع‌های اخیر لاتک ارائه می‌شود.<sup>۱</sup> در این فصل فرض بر این است که  $\text{amsmath}$  در سرآغاز نوشتار فراخوانی شده است:

```
\usepackage{amsmath}
```

---

<sup>۱</sup> اگر آن را ندارید، به [CTAN:macros/latex/required/amslatex](http://CTAN:macros/latex/required/amslatex) مراجعه کنید.

## ۲.۳ فرمول‌های تنها

دو راه برای چیدن یک فرمول وجود دارد: در متن داخل یک پاراگراف (سبک متنی<sup>۲</sup>)، یا پاراگراف می‌تواند برای نمایش جداگانه شکسته شود (سبک نمایشی<sup>۳</sup>). فرمول‌های ریاضی درون متن یک پاراگراف در میان دو نماد \$ وارد می‌شوند:

Add \$a\$ squared and \$b\$ squared  
to get \$c\$ squared. Or, using  
a more mathematical approach:  
 $a^2 + b^2 = c^2$

Add  $a$  squared and  $b$  squared to get  $c$  squared.  
Or, using a more mathematical approach:  $a^2 + b^2 = c^2$

$\TeX$  is pronounced as  
 $\tau\epsilon\chi$   
100 m<sup>3</sup> of water  
This comes from my  $\heartsuit$

$\TeX$  is pronounced as  $\tau\epsilon\chi$   
100 m<sup>3</sup> of water  
This comes from my ♥

اگر می‌خواهید فرمول‌های بیشتری را جدا از بقیه پاراگراف بنویسید، مناسب‌تر است که آن را نمایش دهید به‌جای آنکه پاراگراف را بشکنید. برای انجام این کار از محیط فرمول استفاده کنید و فرمول‌ها را بین  $\begin{equation}$  و  $\end{equation}$  قرار دهید.<sup>۴</sup> آنگاه می‌توانید به فرمول یک برچسب ( $\label$ ) بدهید و در دیگر نقاط نوشتار با فرمان  $\eqref$  به آن ارجاع دهید. اگر می‌خواهید به فرمول اسم ویژه‌ای بدهید به‌جای این‌کار از فرمان  $\tag$  استفاده کنید. از  $\eqref$  نمی‌توانید برای  $\tag$  استفاده کنید.

<sup>۲</sup>text style

<sup>۳</sup>display style

<sup>۴</sup>این یک فرمان  $\amsmath$  است. اگر به این بسته دسترسی ندارید از محیط  $\displaymath$  مربوط به خود لاتک استفاده کنید.



Add  $a$  squared and  $b$  squared to get  $c$  squared. Or, using a more mathematical approach

```
\begin{equation}
 a^2 + b^2 = c^2
\end{equation}
```

Einstein says

```
\begin{equation}
 E = mc^2 \label{clever}
\end{equation}
```

He didn't say

```
\begin{equation}
 1 + 1 = 3 \tag{dumb}
\end{equation}
```

This is a reference to `\eqref{clever}`.

Add  $a$  squared and  $b$  squared to get  $c$  squared. Or, using a more mathematical approach

$$a^2 + b^2 = c^2 \quad (3.1)$$

Einstein says

$$E = mc^2 \quad (3.2)$$

He didn't say

$$1 + 1 = 3 \quad (\text{dumb})$$

This is a reference to (3.2).

اگر نمی‌خواهید لاتک فرمول‌ها را شماره‌گذاری کند، از شکل ستاره‌دار محیط `equation` استفاده کنید، `equation*`، یا حتی آسان‌تر، فرمول را بین دو علامت `[` و `\]` قرار دهید:<sup>۵</sup>

Add  $a$  squared and  $b$  squared to get  $c$  squared. Or, using a more mathematical approach

```
\begin{equation*}
 a^2 + b^2 = c^2
\end{equation*}
```

or you can type less for the same effect:

```
\[a^2 + b^2 = c^2 \]
```

Add  $a$  squared and  $b$  squared to get  $c$  squared. Or, using a more mathematical approach

$$a^2 + b^2 = c^2$$

or you can type less for the same effect:

$$a^2 + b^2 = c^2$$

به تفاوت حروف چینی بین سبک متنی و سبک نمایشی توجه کنید:

<sup>۵</sup> این فرمان دوباره از `amsmath` است. اگر این بسته را فراخوانی نکرده‌اید، از محیط `equation` مربوط به خود لاتک استفاده کنید. نام فرمان‌های `amsmath/LaTeX` ممکن است به نظر برسد که کمی گیج‌کننده هستند، ولی این واقعاً یک مشکل برای کسانی که از این بسته استفاده می‌کنند نیست. بهتر است این بسته را از ابتدا فراخوانی کنید زیرا ممکن است بعداً مجبور به استفاده از آن شوید، و آنگاه محیط‌های غیر شماره‌گذاری شده خود لاتک ممکن است توسط این بسته شماره‌گذاری شود.

This is text style:

```
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}.
```

And this is display style:

```
\begin{equation}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{equation}
```

This is text style:  $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$ .  
And this is display style:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (3.3)$$

در سبک متنی، عبارات طولانی یا عمیق را در `\smash` محصور کنید. این کار لاتک را وادار می‌سازد ارتفاع عبارت را نادیده بگیرد و باعث یکنواخت شدن فاصله بین خط‌ها می‌شود.

A `\mathop{d_{ep}}` mathematical expression followed by a `\mathop{h^{gh}}` expression. As opposed to a `\smash{\mathop{d_{ep}}}` expression followed by a `\smash{\mathop{h^{gh}}}` expression.

A  $d_{ep}$  mathematical expression followed by a  $h^{gh}$  expression. As opposed to a `\smash{d_{ep}}` expression followed by a `\smash{h^{gh}}` expression.

### ۱.۲.۳ سبک ریاضی

همچنین تفاوت‌هایی بین سبک ریاضی و سبک متنی وجود دارد. به عنوان مثال در سبک ریاضی:

۱. بسیاری از فاصله‌ها و شکست خط‌ها در سبک ریاضی بی‌اهمیت هستند، زیرا تمام فاصله‌ها در عبارات ریاضی یا به طور منطقی ایجاد می‌شوند، و یا این که باید توسط فرمان‌هایی مانند `\quad` یا `\qquad` تولید شوند (بعداً به این فرمان‌ها می‌رسیم، بخش ۵.۳ را ببینید).

۲. خط‌های خالی مجاز نیستند. هر فرمول تنها در یک پاراگراف قرار داده می‌شود.

۳. هر حرف به عنوان نام یک متغیر در نظر گرفته می‌شود و به همین منظور چیده می‌شود. اگر می‌خواهید در یک فرمول متن عادی بنویسید (قلم نرمال ایستاده و فاصله نرمال) آنگاه باید متن را بوسیله فرمان `\text{...}` وارد کنید (همچنین بخش ۶.۳ در صفحه ۵۳ را ببینید).

```
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
```

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0$$

```
$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}$
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}$$

ریاضیدان‌ها از نمادهای پیچیده‌ای استفاده می‌کنند: مناسب است که در اینجا از قلم `blackboard` `bold` استفاده کنیم، که با استفاده از `\mathbb` از بسته `amssymb` بدست می‌آید.<sup>۶</sup> آخرین مثال عبارت است از

```
$x^{2} \geq 0 \qquad \text{for all } x \in \mathbb{R}$
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}$$

جدول ۱۴.۳ در صفحه ۶۳ و جدول ۴.۶ در صفحه ۱۱۳ را برای دیدن قلم‌های دیگر ریاضی ببینید.

### ۳.۳ ساختن بلوک‌های فرمولی

در این بخش، مهمترین فرمان‌های مورد استفاده در حروف‌چینی ریاضی را شرح می‌دهیم. بسیاری از فرمان‌های این بخش احتیاج به `amsmath` ندارند (اگر احتیاج داشته باشند، صریحاً بیان می‌شود) اما به‌رحال این بسته را فراخوانی کنید.

حروف یونانی کوچک به صورت `\alpha`، `\beta`، `\gamma`، ... وارد می‌شوند و حروف بزرگ به صورت `\Delta`، `\Gamma`، ... وارد می‌شوند.<sup>۷</sup> به جدول ۲.۳ در صفحه ۵۷ برای دیدن لیستی از حروف یونانی نظری بیندازید.

```
\lambda, \xi, \pi, \theta, \mu, \Phi, \Omega, \Delta
```

$$\lambda, \xi, \pi, \theta, \mu, \Phi, \Omega, \Delta$$

توان‌ها و اندیس‌ها را می‌توان توسط `^` و `_` نوشت. بسیاری از فرمان‌ها سبک ریاضی تنها روی اولین حرف بعد از خودشان تأثیر دارند، بنابراین اگر می‌خواهید یک فرمان بر روی چند حرف تأثیر داشته باشد، باید آن حروف را توسط `{...}` در یک گروه قرار دهید. جدول ۳.۳ در صفحه ۵۸ شامل بسیاری از عملگرها مانند `\subseteq` و `\perp` است.

```
p^3_{ij} \quad m_{\text{Knuth}} \quad a^{x+y} \neq a^{x+y} \quad e^{x^2} \neq \{e^x\}^2
```

$$p_{ij}^3 \quad m_{\text{Knuth}} \quad a^x + y \neq a^{x+y} \quad e^{x^2} \neq e^{x^2}$$

رادیکال توسط `\sqrt` و ریشه  $n$ -ام به صورت `\sqrt[n]` نوشته می‌شود. لاتک اندازه علامت رادیکال را به‌طور خودکار مشخص می‌کند. اگر تنها علامت رادیکال مورد نیاز باشد از `\surd` استفاده کنید.

در جدول ۶.۳ در صفحه ۶۰ دیگر پیکان‌ها مانند  $\Leftarrow$  و  $\Rightarrow$  آورده شده‌اند.

`amssymb` قسمتی از کلاف نیست، اما ممکن است هنوز قسمتی از توزیع لاتک شما باشد. توزیع خود را بررسی کنید یا به `CTAN:/fonts/amssymb/latex/` بروید و آن را دریافت کنید.<sup>۸</sup> در لاتک حروف بزرگ آلفا، بتا، و غیره تعریف شده نیستند زیرا به شکل `A`، `B`، ... به نظر می‌رسند. همینکه رمزینه جدید ریاضی تمام شود، همه چیز تغییر می‌کند.

$\sqrt{x} \Leftrightarrow x^{1/2}$   
 $\sqrt[3]{2}$   
 $\sqrt{x^2 + y}$   
 $\sqrt{x^2 + y^2}$

$$\sqrt{x} \Leftrightarrow x^{1/2} \quad \sqrt[3]{2} \quad \sqrt{x^2 + y} \quad \sqrt{x^2 + y^2}$$

معمولاً از نقطه برای نمایش دادن عمل ضرب هنگام کار با نمادها استفاده می‌شود؛ با این وجود گاهی اوقات از چند نقطه برای کمک کردن به خواننده جهت گروه‌بندی فرمول‌ها استفاده می‌شود. برای نوشتن یک نقطه در وسط از  $\cdot$  استفاده می‌شود. سه نقطه در وسط قرار می‌دهد درحالی‌که  $\dots$  نقطه‌ها را روی خط کرسی قرار می‌دهد. بعلاوه،  $\vdots$  برای قرار دادن عمودی و  $\ddots$  برای قراردادن کج وجود دارند. مثال دیگری را می‌توانید در بخش ۲.۴.۳ ببینید.

$\Psi = v_1 \cdot v_2$   
 $\cdot \dots \cdot$   
 $n! = 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$

$$\Psi = v_1 \cdot v_2 \cdot \dots \quad n! = 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$$

فرمان‌های  $\overline{\quad}$  و  $\underline{\quad}$  خط افقی درست در بالا یا پایین عبارت قرار می‌دهند:

$\overline{\overline{1/3}}$   
 $\underline{\underline{1/3}}$

$$\overline{\overline{1/3}} = \underline{\underline{1/3}}$$

فرمان‌های  $\overbrace{\quad}$  و  $\underbrace{\quad}$  کرشۀ افقی در بالا یا پایین یک عبارت قرار

می‌دهند:

$\underbrace{\overbrace{a+b+c}^6}_{\text{meaning of life}} = 42$   
 $\overbrace{d+e+f}^9$

$$\underbrace{\overbrace{a+b+c}^6 \cdot \overbrace{d+e+f}^9}_{\text{meaning of life}} = 42$$

برای افزودن لهجه مانند پیکان کوچک یا علامت تیلدا به متغیرها، فرمان‌های ارائه شده در جدول ۱.۳ در صفحه ۵۷ ممکن است مفید باشند. کلاه و تیلدا که روی چند حرف قرار می‌گیرد با  $\widetilde{\quad}$  و  $\widehat{\quad}$  درست می‌شود. به تفاوت بین محل قرار گرفتن  $\widehat{\quad}$  و  $\bar{\quad}$  متغیرهایی که دارای اندیس هستند توجه کنید. علامت  $\hat{\quad}$  تولید پرایم می‌کند:

$f(x) = x^2 \quad f'(x) = 2x$   
 $f''(x) = 2$   
 $\widehat{XY} \quad \widehat{\widehat{XY}}$   
 $\bar{x}_0 \quad \bar{\bar{x}}_0$

$$f(x) = x^2 \quad f'(x) = 2x \quad f''(x) = 2$$

$$\widehat{XY} \quad \widehat{\widehat{XY}} \quad \bar{x}_0 \quad \bar{\bar{x}}_0$$

<sup>^</sup>apostrophe

بردارها اغلب با افزودن یک علامت پیکان بر روی یک متغیر بدست می‌آیند. این‌کار را با فرمان `\vec` انجام می‌دهیم. دو فرمان `\overleftarrow` و `\overrightarrow` برای نشان دادن پیکان از  $A$  به  $B$  به‌کار می‌روند:

```

 \vec{a} \quad
 \vec{AB} \quad
 \overrightarrow{AB}

```

$$\vec{a} \quad \vec{AB} \quad \overrightarrow{AB}$$

نام یک تابع مانند لگاریتم اغلب با قلم ایستاده نوشته می‌شود، بنابراین لاتک فرمان‌های زیر را برای نوشتن نام مهمترین توابع به‌کار می‌برد:

```

\arccos \cos \csc \exp \ker \limsup
\arcsin \cosh \deg \gcd \lg \ln
\arctan \cot \det \hom \lim \log
\arg \coth \dim \inf \liminf \max
\sinh \sup \tan \tanh \min \Pr
\sec \sin

```

```

 $\lim_{x \rightarrow 0}$
 $\frac{\sin x}{x} = 1$

```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

برای توابعی که در لیست بالا قرار ندارند، از فرمان `\DeclareMathOperator` استفاده کنید. حتی حالت ستاره‌دار این فرمان‌ها برای توابعی که حد بالا یا پایین دارند وجود دارد. این فرمان‌ها تنها در سرآغاز باید فعال شوند بنابراین مثال زیر باید در سرآغاز قرار داده شود.

```

% \DeclareMathOperator{\argh}{argh}
% \DeclareMathOperator*{\nut}{Nut}
[3\argh = 2\nut_{x=1}]

```

$$3 \operatorname{argh} = 2 \operatorname{Nut}_{x=1}$$

برای تابع هنگ، دو فرم وجود دارد: `\bmod` برای عملگر دوتایی  $a \bmod b$  و `\pmod` برای عبارتی به شکل  $x \equiv a \pmod{b}$ :

```

 $a \bmod b$ \quad
 $x \equiv a \pmod{b}$

```

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

کسر ایستاده را با فرمان `\frac{\dots}{\dots}` می‌نویسیم. در حالت متنی، کسر کوچک نوشته می‌شود تا در ارتفاع خط قرار بگیرد. این فرم را در سبک نمایشی نیز با `\dfrac` می‌توانید اجرا کنید. اغلب فرم کج  $1/2$  بهتر است، زیرا برای کسرهای کوچک خواناتر است:

In display style:

`\[3/8 \quad \frac{3}{8}`  
`\quad \tfrac{3}{8} \]`

In display style:

$$3/8 \quad \frac{3}{8} \quad \tfrac{3}{8}$$

In text style:

`\$1\frac{1}{2}\$~hours \quad`  
`\$1\dfrac{1}{2}\$~hours`

In text style:  $1\frac{1}{2}$  hours      $1\frac{1}{2}$  hours

در اینجا فرمان `\partial` برای مشتق جزئی به کار رفته است:

`\[\sqrt{\frac{x^2}{k+1}}\quad`  
`x^{\frac{2}{k+1}}\quad`  
`\frac{\partial^2 f}{\partial x^2} \]`

$$\sqrt{\frac{x^2}{k+1}} \quad x^{\frac{2}{k+1}} \quad \frac{\partial^2 f}{\partial x^2}$$

برای نوشتن ضرایب دوجمله‌ای یا چیزهایی شبیه این، از فرمان `\binom` از بسته `amsmath` استفاده

می‌شود:

Pascal's rule is

`\begin{equation*}`  
`\binom{n}{k} = \binom{n-1}{k}`  
`+ \binom{n-1}{k-1}`  
`\end{equation*}`

Pascal's rule is

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

برای عملگرهای دوتایی ممکن است قرار دادن نمادها بر روی هم مفید باشد. فرمان

`\stackrel{\#1}{\#2}`

نماد درون `\#1` را به اندازه قلم توان روی `\#2` قرار می‌دهد که در محل معمول آن قرار می‌گیرد.

`\begin{equation*}`

`f_n(x) \stackrel{*}{\approx} 1`  
`\end{equation*}`

$$f_n(x) \stackrel{*}{\approx} 1$$

عملگر انتگرال با فرمان `\int`، عملگر جمع با `\sum`، و عملگر ضرب با `\prod` تولید می‌شوند.

حد بالا و پایین این عملگرها با `\sim` و `\_` مانند اندیس و توان نوشته می‌شوند:

`\begin{equation*}`

`\sum_{i=1}^n \quad`  
`\int_0^{\pi} \quad`  
`\prod_{\epsilon}`  
`\end{equation*}`

$$\sum_{i=1}^n \quad \int_0^{\pi} \quad \prod_{\epsilon}$$

برای کنترل بیشتر روی محل قرار گرفتن اندیس‌ها در عبارات پیچیده، `\substack` فرمان `amsmath` را ارائه می‌کند:

```
\begin{equation*}
\sum^n_{\substack{0<i<n \\
j\subseteq i}}
P(i,j) = Q(i,j)
\end{equation*}
```

$$\sum_{\substack{0 < i < n \\ j \subseteq i}}^n P(i, j) = Q(i, j)$$

لاتک همهٔ انواع براکت و حائل (مانند  $\updownarrow$  || < >) را حمایت می‌کند. براکت‌های گرد و مربعی را می‌توان با کلید مربوط به خودشان نوشت و آکولاد را می‌توان با `\{` نوشت اما همهٔ حائل‌ها را می‌توان با فرمان‌هایی ویژه نوشت (مانند `\updownarrow`).

```
\begin{equation*}
\{a,b,c\} \neq \{a,b,c\}
\end{equation*}
```

$$a, b, c \neq \{a, b, c\}$$

اگر فرمان `\left` را در ابتدای یک حائل چپ، و فرمان `\right` را در ابتدای یک حائل راست قرار دهیم، لاتک به‌طور خودکار اندازهٔ حائل را تصحیح می‌کند. توجه داشته باشید که تمام فرمان‌های `\left` را باید با فرمان متناظر `\right` ببندید. اگر در سمت راست چیزی نمی‌خواهید از `\right` نامرئی استفاده کنید:

```
\begin{equation*}
1 + \left(\frac{1}{1-x^2}\right)^3 \quad
\left. \ddagger \frac{\sim}{\sim} \right)
\end{equation*}
```

$$1 + \left(\frac{1}{1-x^2}\right)^3 \quad \ddagger-$$

گاهی اوقات لازم است تا اندازهٔ درست یک حائل ریاضی را دستی تنظیم کنیم که با فرمان‌های `\Bigg`، `\bigg`، `\Big`، `\big` و `\Bigg` به عنوان پیشوند بیشتر فرمان‌های حائل امکان‌پذیر است:

```
\Big((x+1)(x-1)\Big)^2 \\
\big(\ \Big(\ \bigg(\ \Bigg(\ \quad
\big\} \Big\} \bigg\} \Bigg\} \quad
\big\| \Big\| \bigg\| \Bigg\| \quad
\big\Downarrow \Big\Downarrow
\bigg\Downarrow \Bigg\Downarrow
```

$$\left(\left(\left(\left((x+1)(x-1)\right)^2\right)\right)\right)\right)$$

برای دیدن لیست کاملی از حائل‌ها جدول ۸.۳ در صفحه ۶۱ را ببینید.

## ۴.۳ تنظیم عمودی

## ۱.۴.۳ فرمول‌های چندگانه

برای فرمول‌هایی که در چند خط قرار می‌گیرند یا برای دستگاه معادلات، می‌توانید از محیط `align` و `align*` به جای `equation*` استفاده کنید.<sup>۹</sup> با `align` هر خط معادله یک شماره می‌گیرد. `align*` هیچ چیز را شماره‌گذاری نمی‌کند.

محیط `align` یک معادله را پیرامون علامت `&` گرد می‌کند. فرمان `\\` خط‌ها را می‌شکند. اگر می‌خواهید یک معادله را شماره‌گذاری نکنید از فرمان `\nonumber` برای حذف شماره آن استفاده کنید. این فرمان باید قبل از `\\` قرار داده شود:

```
\begin{align}
f(x) &= (a+b)(a-b) \label{1}\\
&= a^2-ab+ba-b^2 \\
&= a^2+b^2 \tag{wrong}
\end{align}
This is a reference to \eqref{1}.
```

$$f(x) = (a + b)(a - b) \quad (3.4)$$

$$= a^2 - ab + ba - b^2 \quad (3.5)$$

$$= a^2 + b^2 \quad (\text{wrong})$$

This is a reference to (3.4).

فرمول‌های طولانی به صورت خودکار شکسته نمی‌شوند. نویسنده باید مشخص کند کجا باید شکسته شوند و تورفتگی مناسب را مشخص کند:

```
\begin{align}
f(x) &= 3x^5 + x^4 + 2x^3 \\
&\quad \nonumber \\
&\quad \quad \quad \quad \quad \quad \\
&\quad \quad \quad \quad \quad \quad \quad \quad \\
&\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \\
&= g(x) - h(x)
\end{align}
```

$$f(x) = 3x^5 + x^4 + 2x^3$$

$$+ 9x^2 + 12x + 23 \quad (3.6)$$

$$= g(x) - h(x) \quad (3.7)$$

بسته `amsmath` چند محیط مفید دیگر را نیز در بر دارد: `multline`، `gather`، `flalign` و `split`. برای اطلاعات بیشتر به راهنمای این بسته مراجعه کنید.

## ۲.۴.۳ آرایه و ماتریس

برای حروف چینی آرایه‌ها از محیط `array` استفاده کنید. این محیط شبیه محیط `tabular` است. فرمان `\\` برای شکستن خط‌ها به‌کار می‌رود:

<sup>۹</sup> محیط `align` از بسته `amsmath` است. محیط مشابه به این محیط در خود لاتک با عنوان `eqnarray` وجود دارد، اما عموماً توصیه نمی‌شود زیرا مکان و برجسب آن پایدار نیست.



```
\begin{equation*}
\mathbf{X} = \left(
\begin{array}{ccc}
x_1 & x_2 & \dots \\
x_3 & x_4 & \dots \\
\vdots & \vdots & \ddots
\end{array}
\right)
\end{equation*}
```

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots \\ x_3 & x_4 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

از محیط array همچنین برای نوشتن توابع چندضابطه توسط یک . به عنوان یک حائل راست نامرئی استفاده می‌شود:<sup>۱۰</sup>

```
\begin{equation*}
|x| = \left\{
\begin{array}{rl}
-x & \text{if } x < 0 \\
0 & \text{if } x = 0 \\
x & \text{if } x > 0
\end{array}
\right.
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ x & \text{if } x > 0 \end{cases}$$

array را می‌توان برای نوشتن ماتریس‌ها نیز به‌کار برد، اما amsmath راه حل بهتری را توسط محیط matrix پیشنهاد می‌کند. شش نسخه از آن با حائل‌های مختلف وجود دارد: matrix (خالی)، pmatrix، bmatrix، Bmatrix، vmatrix و Vmatrix. با array لازم نیست تعداد ستون‌ها را مشخص کنید. بیشترین تعداد ستون ۱۰ است اما قابل تغییر است (هرچند معمولاً بیشتر از ۱۰ ستون لازم نیست!).

```
\begin{equation*}
\begin{matrix}
1 & 2 \\
3 & 4
\end{matrix} \quad
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\end{equation*}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

<sup>۱۰</sup> اگر می‌خواهید خیلی از این فرم استفاده کنید محیط cases از بسته amsmath کار را بسیار راحت می‌کند و بنابراین ارزش نگاه کردن را دارد.

## ۵.۳ فاصله در محیط ریاضی

اگر فاصله انتخاب شده توسط لاتک در فرمول‌ها مناسب نیست، می‌توان آن را با فرمان‌هایی تصحیح کرد:  $\int \ln x dx$  برای  $\int \ln x dx$ ،  $\int \ln x dx$  برای  $\int \ln x dx$  و  $\int \ln x dx$  برای  $\int \ln x dx$ . حرف فرار  $\backslash$  تولید یک فاصله بین  $\int$  و  $\ln x dx$  می‌کند. اندازه  $\quad$  متناظر با عرض حرف 'M' از این قلم جاری است.  $\int - \ln x dx$  تولید یک فاصله منفی به اندازه  $\quad$  می‌کند. توجه کنید 'd' در عملیات دیفرانسیل به خوبی در قلم ایستاده نوشته می‌شود:

```
\begin{equation*}
\int_1^2 \ln x \mathrm{d}x \quad
\int_1^2 \ln x \backslash, \mathrm{d}x
\end{equation*}
```

$$\int_1^2 \ln x dx \quad \int_1^2 \ln x dx$$

در مثال بعد، تابع جدید  $\ud$  را تعریف می‌کنیم که نماد  $d$  را تولید می‌کند (به فاصله قبل از  $d$  توجه داشته باشید)، بنابراین لازم نیست هر بار آن را بنویسیم. فرمان  $\newcommand$  در سرآغاز آورده می‌شود.

```
\newcommand{\ud}{\, \mathrm{d}}
```

```
\begin{equation*}
\int_a^b f(x) \ud x
\end{equation*}
```

$$\int_a^b f(x) dx$$

اگر می‌خواهید انتگرال چندگانه را بنویسید، خواهید دید که فاصله بین انتگرال‌ها نامطبوع است. می‌توانید این فاصله را با فرمان  $\!$  تغییر دهید، اما بسته  $amsmath$  راه حل ساده‌تری برای این کار دارد که عبارت است از  $\iint$ ،  $\iiint$ ،  $\iiint$  و  $\idotsint$ .

```
\newcommand{\ud}{\, \mathrm{d}}
```

```
\[\int \int f(x)g(y)
\ud x \ud y \]
\[\int \! \! \! \int
f(x)g(y) \ud x \ud y \]
\[\iiint f(x)g(y) \ud x \ud y \]
```

$$\int \int f(x)g(y) dx dy$$

$$\iint f(x)g(y) dx dy$$

$$\iiint f(x)g(y) dx dy$$

برای اطلاعات بیشتر به راهنمای الکترونیکی `testmath.tex` از  $AMS-TEX$  یا فصل ۸ از [۳] مراجعه کنید.

## ۱.۵.۳ اشباح

وقتی فرمول‌های مرتب عمودی شامل  $\sim$  و  $\_$  می‌نویسید، گاهی اوقات لاتک خیلی کمک نمی‌کند. با استفاده از فرمان `\phantom` می‌توانید فضایی برای حرفی که نمی‌خواهید در خروجی ظاهر شود ایجاد کنید. راحت‌ترین راه برای فهمیدن این موضوع مثال زیر است:

```
\begin{equation*}
{}^{\sim 14}_{6}\text{C}
\quad \text{versus} \quad
{}^{14}_{6}\text{C}
\end{equation*}
```

$${}^{\sim 14}_{6}\text{C} \quad \text{versus} \quad {}^{\phantom{1}14}_{\phantom{1}6}\text{C}$$

اگر می‌خواهید تعداد زیادی از ایزوتوپ‌ها را همانند مثال بالا بنویسید، بسته `mhchem` برای نوشتن فرمول‌های شیمی بسیار مفید است.

## ۶.۳ ریزه‌کاری با قلم‌های ریاضی

قلم‌های مختلف ریاضی را در جدول ۱۴.۳ در صفحه ۶۳ آورده‌ایم.

```
Re \quad
\mathcal{R} \quad
\mathfrak{R} \quad
\mathbb{R}
```

$$\Re \quad \mathcal{R} \quad \mathfrak{R} \quad \mathbb{R}$$

دوای آخر به `amssymb` یا `amsfonts` احتیاج دارند. گاهی اوقات باید به لاتک بگویید که اندازه را تصحیح کند. در سبک ریاضی، این‌کار را با فرمان زیر انجام می‌دهیم:

```
\displaystyle (۱۲۳), \textstyle (۱۲۳), \scriptstyle (۱۲۳) و
\scriptscriptstyle (۱۲۳).
```

اگر  $\sum$  در یک کسر قرار داشته باشد، به سبک متنی حروف چینی می‌شود مگر این که به لاتک اطلاع

دهید:

```
\begin{equation*}
P = \frac{\displaystyle
\sum_{i=1}^n (x_i - x)
(y_i - y)}
{\left[
\displaystyle
\sum_{i=1}^n (x_i - x)^2
\sum_{i=1}^n (y_i - y)^2
\right]^{1/2}}
\end{equation*}
```

$$P = \frac{\sum_{i=1}^n (x_i - x)(y_i - y)}{\left[ \sum_{i=1}^n (x_i - x)^2 \sum_{i=1}^n (y_i - y)^2 \right]^{1/2}}$$

تغییر سبک عموماً روی عملگرهای بزرگ و حدود آنها تاثیر می‌گذارد.

### ۱.۶.۳ حروف سیاه

نوشتن حروف سیاه در لاتک سخت است؛ یک حرف چینی آماتور ممکن است بخواهد بیش از حد از حروف سیاه استفاده کند. فرمان تغییر قلم `\mathbf` حروف سیاه را تولید می‌کند، اما این حروف ایستاده هستند و نمادهای ریاضی ایتالیک هستند، و یک فرمان `\boldmath` وجود دارد، این فرمان تنها باید در خارج از سبک ریاضی مورد استفاده قرار گیرد. با این وجود از آن می‌توان برای نمادها نیز استفاده کرد:

```
\mu, M \quad
\mathbf{\mu}, \mathbf{M}$
\quad \boldmath{\mu, M$}
```

|          |                   |                       |
|----------|-------------------|-----------------------|
| $\mu, M$ | $\mu, \mathbf{M}$ | $\boldsymbol{\mu}, M$ |
|----------|-------------------|-----------------------|

بسته `amsbsy` (توسط `amsmath` توزیع می‌شود) و همچنین `bm` از کلاف `tools` این کار را با ارائه فرمان `\boldsymbol` راحت‌تر می‌کنند:

```
\mu, M \quad
\boldsymbol{\mu}, \boldsymbol{M}$
```

|          |                       |
|----------|-----------------------|
| $\mu, M$ | $\boldsymbol{\mu}, M$ |
|----------|-----------------------|

### ۷.۳ قضیه‌ها، قانون‌ها

هنگام نوشتن نوشتار ریاضی، ممکن است به نوشتن ساختارهایی مانند قضیه، تعریف، اصل، و غیره احتیاج پیدا کنید.

```
\newtheorem{name}[counter]{text}[section]
```

آرگومان `name` کلمه کلیدی برای شناسایی `theorem` است. با آرگومان `text` نام واقعی قضیه را معرفی می‌کنید که در خروجی چاپ می‌شود. آرگومان‌های درون کروشه اختیاری هستند. از آنها برای مشخص کردن نوع شماره‌گذاری قضیه استفاده می‌شود. از آرگومان `counter` برای هم‌نوع شدن شماره‌گذاری با یک شماره‌گذاری تعریف شده استفاده می‌شود. آرگومان `section` اجازه می‌دهد در شماره قضیه شماره بخش نیز وارد شود. بعد از اجرای فرمان `\newtheorem` در سرآغاز مستندات، می‌توانید از محیط تعریف شده در نوشتار به شکل زیر استفاده کنید.

```
\begin{name}[text]
This is my interesting theorem
\end{name}
```

بسته `amsthm` (قسمتی از `AMS-TeX`) فرمان `\theoremstyle{style}` را ارائه می‌کند که توسط آن می‌توانید از محیط‌های از پیش تعریف شده مانند `definition` (تیترا بزرگ، بدنه رومن)، `plain` (تیترا بزرگ، بدنه ایتالیک) یا `remark` (تیترا ایتالیک، بدنه رومن) استفاده کنید. تئوری بس است. مثال‌های زیر هر نوع ابهامی را از بین می‌برد و مشخص می‌کند محیط `\newtheorem` کمی برای فهمیدن مشکل است. ابتدا قضیه‌ها را تعریف می‌کنیم:

```
\theoremstyle{definition} \newtheorem{law}{Law}
\theoremstyle{plain} \newtheorem{jury}[law]{Jury}
\theoremstyle{remark} \newtheorem*{marg}{Margaret}
```

```
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}.\end{jury}
\begin{marg}No, No, No\end{marg}
```

**Law 1.** Don't hide in the witness box

**Jury 2** (The Twelve). *It could be you! So beware and see law 1.*

*Margaret.* No, No, No

قضیه `Jury` دارای شماره‌گذاری `Law` است، بنابراین شماره‌ای را اخذ می‌کند که در دنباله شماره `Laws` است. آرگومان داخل کروشه برای معین کردن یک عنوان شبیه قضیه است.

```
\newtheorem{mur}{Murphy}[section]
```

```
\begin{mur} If there are two or
more ways to do something, and
one of those ways can result in
a catastrophe, then someone
will do it.\end{mur}
```

**Murphy 3.7.1.** *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

قضیه `Murphy` شماره‌ای وابسته به شماره بخش جاری اخذ می‌کند. می‌توانید به جای بخش از فصل و شبیه آن استفاده کنید. بسته `amsthm` دارای محیط `proof` نیز است.

```
\begin{proof}
Trivial, use
[E=mc^2]
\end{proof}
```

*Proof.* Trivial, use

$$E = mc^2$$

□

با فرمان `\qedhere` می‌توانید علامت انتهای اثبات را در مواقعی که به‌تنهایی در یک خط قرار دارد در مکان مناسبی درج کنید.

```
\begin{proof}
 Trivial, use
 \[E=mc^2 \qedhere\]
\end{proof}
```

*Proof.* Trivial, use

$$E = mc^2$$

□

اگر می‌خواهید تا محیط مناسبی برای خود طراحی کنید، بسته `ntheorem` گزینه‌های بسیار زیادی در اختیارتان قرار می‌دهد.

## ۸.۳ فهرست نمادهای ریاضی

جدول‌های زیر تمام نمادهایی را نشان می‌دهند که در سبک ریاضی وجود دارند. برای استفاده از نمادهای جدول‌های ۱۲.۳ الی ۱۱۹.۳ بستهٔ `amssymb` باید در سرآغاز فراخوانی شده باشد و قلم‌های `AMS` باید روی سیستم نصب شده باشند. اگر بستهٔ `AMS` و قلم‌های آن روی سیستم شما نصب نیست، نگاهی به `CTAN:macros/latex/required/amslatex` بیندازید. لیست کامل‌تری از نمادها را می‌توانید در `CTAN:info/symbols/comprehensive` بیابید.

جدول ۱.۳: لهجه‌های سبک ریاضی

|                |                           |             |                        |                   |                              |
|----------------|---------------------------|-------------|------------------------|-------------------|------------------------------|
| $\hat{a}$      | <code>\hat{a}</code>      | $\check{a}$ | <code>\check{a}</code> | $\tilde{a}$       | <code>\tilde{a}</code>       |
| $\grave{a}$    | <code>\grave{a}</code>    | $\dot{a}$   | <code>\dot{a}</code>   | $\ddot{a}$        | <code>\ddot{a}</code>        |
| $\bar{a}$      | <code>\bar{a}</code>      | $\vec{a}$   | <code>\vec{a}</code>   | $\widehat{AAA}$   | <code>\widehat{AAA}</code>   |
| $\acute{a}$    | <code>\acute{a}</code>    | $\breve{a}$ | <code>\breve{a}</code> | $\widetilde{AAA}$ | <code>\widetilde{AAA}</code> |
| $\mathring{a}$ | <code>\mathring{a}</code> |             |                        |                   |                              |

جدول ۲.۳: الفبای یونانی

بعضی از حروف مانند `\Alpha`، `\Beta` و غیره دارای شکل بزرگ نیستند، زیرا شکل بزرگ آنها شبیه حروف رومن `A`، `B` و ... هستند.

|               |                          |             |                        |             |                        |           |                       |
|---------------|--------------------------|-------------|------------------------|-------------|------------------------|-----------|-----------------------|
| $\alpha$      | <code>\alpha</code>      | $\theta$    | <code>\theta</code>    | $o$         | <code>o</code>         | $v$       | <code>\upsilon</code> |
| $\beta$       | <code>\beta</code>       | $\vartheta$ | <code>\vartheta</code> | $\pi$       | <code>\pi</code>       | $\phi$    | <code>\phi</code>     |
| $\gamma$      | <code>\gamma</code>      | $\iota$     | <code>\iota</code>     | $\varpi$    | <code>\varpi</code>    | $\varphi$ | <code>\varphi</code>  |
| $\delta$      | <code>\delta</code>      | $\kappa$    | <code>\kappa</code>    | $\rho$      | <code>\rho</code>      | $\chi$    | <code>\chi</code>     |
| $\epsilon$    | <code>\epsilon</code>    | $\lambda$   | <code>\lambda</code>   | $\varrho$   | <code>\varrho</code>   | $\psi$    | <code>\psi</code>     |
| $\varepsilon$ | <code>\varepsilon</code> | $\mu$       | <code>\mu</code>       | $\sigma$    | <code>\sigma</code>    | $\omega$  | <code>\omega</code>   |
| $\zeta$       | <code>\zeta</code>       | $\nu$       | <code>\nu</code>       | $\varsigma$ | <code>\varsigma</code> |           |                       |
| $\eta$        | <code>\eta</code>        | $\xi$       | <code>\xi</code>       | $\tau$      | <code>\tau</code>      |           |                       |
| $\Gamma$      | <code>\Gamma</code>      | $\Lambda$   | <code>\Lambda</code>   | $\Sigma$    | <code>\Sigma</code>    | $\Psi$    | <code>\Psi</code>     |
| $\Delta$      | <code>\Delta</code>      | $\Xi$       | <code>\Xi</code>       | $\Upsilon$  | <code>\Upsilon</code>  | $\Omega$  | <code>\Omega</code>   |
| $\Theta$      | <code>\Theta</code>      | $\Pi$       | <code>\Pi</code>       | $\Phi$      | <code>\Phi</code>      |           |                       |

<sup>۱۱</sup> این جدول‌ها از `symbols.tex` توسط David Carlisle انتخاب شده‌اند و طبق توصیهٔ Josef Tkadlec تغییر یافته‌اند.

## جدول ۳.۳: روابط دوتایی

نمادهای زیر را با افزودن `\not` در فرمان آنها می‌توانید نقیض کنید.

|               |                          |               |                          |             |                          |
|---------------|--------------------------|---------------|--------------------------|-------------|--------------------------|
| $<$           | <code>&lt;</code>        | $>$           | <code>&gt;</code>        | $=$         | <code>=</code>           |
| $\leq$        | <code>\leq or \le</code> | $\geq$        | <code>\geq or \ge</code> | $\equiv$    | <code>\equiv</code>      |
| $\ll$         | <code>\ll</code>         | $\gg$         | <code>\gg</code>         | $\dot{=}$   | <code>\doteq</code>      |
| $\prec$       | <code>\prec</code>       | $\succ$       | <code>\succ</code>       | $\sim$      | <code>\sim</code>        |
| $\preceq$     | <code>\preceq</code>     | $\succeq$     | <code>\succeq</code>     | $\simeq$    | <code>\simeq</code>      |
| $\subset$     | <code>\subset</code>     | $\supset$     | <code>\supset</code>     | $\approx$   | <code>\approx</code>     |
| $\subseteq$   | <code>\subseteq</code>   | $\supseteq$   | <code>\supseteq</code>   | $\cong$     | <code>\cong</code>       |
| $\sqsubset^a$ | <code>\sqsubset^a</code> | $\sqsupset^a$ | <code>\sqsupset^a</code> | $\bowtie^a$ | <code>\Join^a</code>     |
| $\sqsubseteq$ | <code>\sqsubseteq</code> | $\sqsupseteq$ | <code>\sqsupseteq</code> | $\bowtie$   | <code>\bowtie</code>     |
| $\in$         | <code>\in</code>         | $\ni$         | <code>\ni , \owns</code> | $\propto$   | <code>\propto</code>     |
| $\vdash$      | <code>\vdash</code>      | $\dashv$      | <code>\dashv</code>      | $\models$   | <code>\models</code>     |
| $ $           | <code>\mid</code>        | $\parallel$   | <code>\parallel</code>   | $\perp$     | <code>\perp</code>       |
| $\smile$      | <code>\smile</code>      | $\frown$      | <code>\frown</code>      | $\asymp$    | <code>\asymp</code>      |
| $:$           | <code>:</code>           | $\notin$      | <code>\notin</code>      | $\neq$      | <code>\neq or \ne</code> |

<sup>a</sup> از بسته `latexsym` برای دستیابی به این نماد استفاده کنید



## جدول ۴.۳: عملگرهای دوتایی

|                   |                                       |                    |                                          |                  |                             |
|-------------------|---------------------------------------|--------------------|------------------------------------------|------------------|-----------------------------|
| $+$               | <code>+</code>                        | $-$                | <code>-</code>                           |                  |                             |
| $\pm$             | <code>\pm</code>                      | $\mp$              | <code>\mp</code>                         | $\triangleleft$  | <code>\triangleleft</code>  |
| $\cdot$           | <code>\cdot</code>                    | $\div$             | <code>\div</code>                        | $\triangleright$ | <code>\triangleright</code> |
| $\times$          | <code>\times</code>                   | $\setminus$        | <code>\setminus</code>                   | $\star$          | <code>\star</code>          |
| $\cup$            | <code>\cup</code>                     | $\cap$             | <code>\cap</code>                        | $*$              | <code>\ast</code>           |
| $\sqcup$          | <code>\sqcup</code>                   | $\sqcap$           | <code>\sqcap</code>                      | $\circ$          | <code>\circ</code>          |
| $\vee$            | <code>\vee</code> , <code>\lor</code> | $\wedge$           | <code>\wedge</code> , <code>\land</code> | $\bullet$        | <code>\bullet</code>        |
| $\oplus$          | <code>\oplus</code>                   | $\ominus$          | <code>\ominus</code>                     | $\diamond$       | <code>\diamond</code>       |
| $\odot$           | <code>\odot</code>                    | $\oslash$          | <code>\oslash</code>                     | $\uplus$         | <code>\uplus</code>         |
| $\otimes$         | <code>\otimes</code>                  | $\bigcirc$         | <code>\bigcirc</code>                    | $\amalg$         | <code>\amalg</code>         |
| $\triangleleft$   | <code>\bigtriangleup</code>           | $\triangleright$   | <code>\bigtriangleright</code>           | $\dagger$        | <code>\dagger</code>        |
| $\triangleleft^a$ | <code>\bigtriangleup</code>           | $\triangleright^a$ | <code>\bigtriangleright</code>           | $\ddagger$       | <code>\ddagger</code>       |
| $\triangleleft^a$ | <code>\unlhd</code>                   | $\triangleright^a$ | <code>\unrhd</code>                      | $\wr$            | <code>\wr</code>            |

## جدول ۵.۳: عملگرهای بزرگ

|             |                        |              |                         |             |                        |
|-------------|------------------------|--------------|-------------------------|-------------|------------------------|
| $\sum$      | <code>\sum</code>      | $\bigcup$    | <code>\bigcup</code>    | $\bigvee$   | <code>\bigvee</code>   |
| $\prod$     | <code>\prod</code>     | $\bigcap$    | <code>\bigcap</code>    | $\bigwedge$ | <code>\bigwedge</code> |
| $\coprod$   | <code>\coprod</code>   | $\bigsqcup$  | <code>\bigsqcup</code>  | $\biguplus$ | <code>\biguplus</code> |
| $\int$      | <code>\int</code>      | $\oint$      | <code>\oint</code>      | $\bigodot$  | <code>\bigodot</code>  |
| $\bigoplus$ | <code>\bigoplus</code> | $\bigotimes$ | <code>\bigotimes</code> |             |                        |

## جدول ۶.۳: پیکان‌ها

|                      |                                               |                        |                                   |
|----------------------|-----------------------------------------------|------------------------|-----------------------------------|
| $\leftarrow$         | <code>\leftarrow</code> or <code>\gets</code> | $\longleftarrow$       | <code>\longleftarrow</code>       |
| $\rightarrow$        | <code>\rightarrow</code> or <code>\to</code>  | $\longrightarrow$      | <code>\longrightarrow</code>      |
| $\leftrightarrow$    | <code>\leftrightarrow</code>                  | $\longleftrightarrow$  | <code>\longleftrightarrow</code>  |
| $\Leftarrow$         | <code>\Leftarrow</code>                       | $\Lleftarrow$          | <code>\Lleftarrow</code>          |
| $\Rightarrow$        | <code>\Rightarrow</code>                      | $\Rrightarrow$         | <code>\Rrightarrow</code>         |
| $\Leftrightarrow$    | <code>\Leftrightarrow</code>                  | $\Llongleftrightarrow$ | <code>\Llongleftrightarrow</code> |
| $\mapsto$            | <code>\mapsto</code>                          | $\longmapsto$          | <code>\longmapsto</code>          |
| $\hookrightarrow$    | <code>\hookrightarrow</code>                  | $\hookleftarrow$       | <code>\hookleftarrow</code>       |
| $\lleftarrow$        | <code>\lleftarrow</code>                      | $\rrightarrow$         | <code>\rrightarrow</code>         |
| $\lleftarrow$        | <code>\lleftarrow</code>                      | $\rightarrow$          | <code>\rightarrow</code>          |
| $\rightleftharpoons$ | <code>\rightleftharpoons</code>               | $\iff$                 | <code>\iff</code> (bigger spaces) |
| $\uparrow$           | <code>\uparrow</code>                         | $\downarrow$           | <code>\downarrow</code>           |
| $\updownarrow$       | <code>\updownarrow</code>                     | $\Uparrow$             | <code>\Uparrow</code>             |
| $\Downarrow$         | <code>\Downarrow</code>                       | $\Updownarrow$         | <code>\Updownarrow</code>         |
| $\nearrow$           | <code>\nearrow</code>                         | $\searrow$             | <code>\searrow</code>             |
| $\swarrow$           | <code>\swarrow</code>                         | $\nwarrow$             | <code>\nwarrow</code>             |
| $\leadsto$           | <code>\leadsto</code> <sup>a</sup>            |                        |                                   |

<sup>a</sup> از بسته latexsym برای دستیابی به این نماد استفاده کنید

## جدول ۷.۳: پیکان‌ها به عنوان لهجه

|                           |                                      |                                       |                                                  |
|---------------------------|--------------------------------------|---------------------------------------|--------------------------------------------------|
| $\overrightarrow{AB}$     | <code>\overrightarrow{AB}</code>     | $\underline{\overrightarrow{AB}}$     | <code>\underline{\overrightarrow{AB}}</code>     |
| $\overleftarrow{AB}$      | <code>\overleftarrow{AB}</code>      | $\underline{\overleftarrow{AB}}$      | <code>\underline{\overleftarrow{AB}}</code>      |
| $\overleftrightarrow{AB}$ | <code>\overleftrightarrow{AB}</code> | $\underline{\overleftrightarrow{AB}}$ | <code>\underline{\overleftrightarrow{AB}}</code> |

جدول ۸.۳: حائل‌ها

|   |               |   |               |   |              |
|---|---------------|---|---------------|---|--------------|
| ( | (             | ) | )             | ↑ | \uparrow     |
| [ | [ or \lbrack  | ] | ] or \rbrack  | ↓ | \downarrow   |
| { | \{ or \lbrace | } | \} or \rbrace | ↕ | \updownarrow |
| < | \langle       | > | \rangle       | ⇧ | \Uparrow     |
|   | or \vert      |   | \  or \Vert   | ⇩ | \Downarrow   |
| / | /             | \ | \backslash    | ⇕ | \Updownarrow |
| ⌊ | \lfloor       | ⌋ | \rfloor       |   |              |
| ⌈ | \lceil        | ⌉ | \rceil        |   |              |

جدول ۹.۳: حائل‌های بزرگ

|   |             |   |            |   |             |
|---|-------------|---|------------|---|-------------|
| ( | \lgroup     | ) | \rgroup    | ∫ | \lmoustache |
|   | \arrowvert  |   | \Arrowvert |   | \bracevert  |
| } | \rmoustache |   |            |   |             |

جدول ۱۰.۳: نمادهای متفرقه

|     |               |     |            |   |                   |   |                       |
|-----|---------------|-----|------------|---|-------------------|---|-----------------------|
| ... | \dots         | ... | \cdots     | : | \vdots            | ⋯ | \ddots                |
| ℏ   | \hbar         | ℐ   | \imath     | ℐ | \jmath            | ℓ | \ell                  |
| ℜ   | \Re           | ℑ   | \Im        | ℵ | \aleph            | ∅ | \wp                   |
| ∀   | \forall       | ∃   | \exists    | ∅ | \mho <sup>a</sup> | ∂ | \partial              |
| '   | '             | '   | \prime     | ∅ | \emptyset         | ∞ | \infty                |
| ∇   | \nabla        | △   | \triangle  | □ | \Box <sup>a</sup> | ◇ | \Diamond <sup>a</sup> |
| ⊥   | \bot          | ⊤   | \top       | ∠ | \angle            | √ | \surd                 |
| ♠   | \diamondsuit  | ♥   | \heartsuit | ♣ | \clubsuit         | ♠ | \spadesuit            |
| ¬   | \neg or \lnot | ♭   | \flat      | ♮ | \natural          | ♯ | \sharp                |

<sup>a</sup> از بسته latexsym برای دستیابی به این نماد استفاده کنید

جدول ۱۱.۳: نمادهای غیر ریاضی

این نمادها را در سبک متنی نیز می‌توان به کار برد.

|   |       |   |    |   |            |   |                 |
|---|-------|---|----|---|------------|---|-----------------|
| † | \dag  | § | \S | © | \copyright | ® | \textregistered |
| ‡ | \ddag | ¶ | \P | £ | \pounds    | % | \%              |

جدول ۱۲.۳: حائل‌های  $AMS$ .

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| $\ulcorner$ | $\urcorner$ | $\llcorner$ | $\lrcorner$ |
| $\lvert$    | $\rvert$    | $\lVert$    | $\rVert$    |

جدول ۱۳.۳:  $AMS$  یونانی و عبری

|            |             |         |          |           |
|------------|-------------|---------|----------|-----------|
| $\digamma$ | $\varkappa$ | $\beth$ | $\gimel$ | $\daleth$ |
|------------|-------------|---------|----------|-----------|

جدول ۱۴.۳: الفبای ریاضی

جدول ۴.۶ در صفحه ۱۱۳ را برای دیگر قلم‌های ریاضی ببینید.

| نمونه                                                | فرمان                           | بسته مورد نیاز                                |
|------------------------------------------------------|---------------------------------|-----------------------------------------------|
| $ABCDEabcde1234$                                     | $\mathrm{ABCDE abcde 1234}$     |                                               |
| $\overline{ABCDEabcde1234}$                          | $\mathit{ABCDE abcde 1234}$     |                                               |
| $ABCDEFabcde1234$                                    | $\mathnormal{ABCDE abcde 1234}$ |                                               |
| $\mathcal{ABCDE}$                                    | $\mathcal{ABCDE abcde 1234}$    |                                               |
| $\mathscr{ABCDE}$                                    | $\mathscr{ABCDE abcde 1234}$    | <code>mathrsfs</code>                         |
| $\mathbb{ABCDEabcde1234}$                            | $\mathfrak{ABCDE abcde 1234}$   | <code>amsfonts</code> or <code>amssymb</code> |
| $\mathbb{ABCDE}\Omega\mathbb{K}\mathbb{H}\mathbb{Z}$ | $\mathbb{ABCDE abcde 1234}$     | <code>amsfonts</code> or <code>amssymb</code> |

جدول ۱۵.۳: عملگرهای دوتایی  $AMS$

|              |              |                |                |
|--------------|--------------|----------------|----------------|
| $\dot{+}$    | $\dot{+}$    | $\cdot$        | $\cdot$        |
| $\ltimes$    | $\ltimes$    | $\rtimes$      | $\rtimes$      |
| $\doublecup$ | $\doublecup$ | $\doublecap$   | $\doublecap$   |
| $\veebar$    | $\veebar$    | $\bar{\wedge}$ | $\bar{\wedge}$ |
| $\boxplus$   | $\boxplus$   | $\boxminus$    | $\boxminus$    |
| $\boxtimes$  | $\boxtimes$  | $\boxdot$      | $\boxdot$      |
| $\intercal$  | $\intercal$  | $\circledast$  | $\circledast$  |
| $\curlyvee$  | $\curlyvee$  | $\curlywedge$  | $\curlywedge$  |

جدول ۱۶.۳: روابط دوتایی  $AMS$ 

|                       |                |                  |
|-----------------------|----------------|------------------|
| $\lessdot$            | $\gtrdot$      | $\doteqdot$      |
| $\leqslant$           | $\geqslant$    | $\risingdotseq$  |
| $\eqslantless$        | $\eqslantgtr$  | $\fallingdotseq$ |
| $\leqq$               | $\geqq$        | $\eqcirc$        |
| $\lll$ or $\llless$   | $\ggg$         | $\circeq$        |
| $\lesssim$            | $\gtrsim$      | $\triangleq$     |
| $\lessapprox$         | $\gtrapprox$   | $\bumpeq$        |
| $\lessgtr$            | $\gtrless$     | $\Bumpeq$        |
| $\lesseqgtr$          | $\gtreqless$   | $\thicksim$      |
| $\lesseqqgtr$         | $\gtreqqless$  | $\thickapprox$   |
| $\preccurlyeq$        | $\succcurlyeq$ | $\approxeq$      |
| $\curlyeqprec$        | $\curlyeqsucc$ | $\backsim$       |
| $\precsim$            | $\succsim$     | $\backsimeq$     |
| $\precapprox$         | $\succapprox$  | $\vDash$         |
| $\subseteq$           | $\supseteq$    | $\Vdash$         |
| $\shortparallel$      | $\Supset$      | $\Vvdash$        |
| $\blacktriangleleft$  | $\sqsupset$    | $\backepsilon$   |
| $\vartriangleright$   | $\because$     | $\varpropto$     |
| $\blacktriangleright$ | $\Subset$      | $\between$       |
| $\trianglerighteq$    | $\smallfrown$  | $\pitchfork$     |
| $\vartriangleleft$    | $\shortmid$    | $\smallsmile$    |
| $\trianglelefteq$     | $\therefore$   | $\sqsubset$      |

جدول ۱۷.۳: پیکان‌های  $\text{AMS}$ 

|                      |                                 |                        |                                   |
|----------------------|---------------------------------|------------------------|-----------------------------------|
| $\dashleftarrow$     | <code>\dashleftarrow</code>     | $\dashrightarrow$      | <code>\dashrightarrow</code>      |
| $\Lleftarrow$        | <code>\Lleftarrow</code>        | $\Rrightarrow$         | <code>\Rrightarrow</code>         |
| $\twoheadleftarrow$  | <code>\twoheadleftarrow</code>  | $\twoheadrightarrow$   | <code>\twoheadrightarrow</code>   |
| $\leftarrowtail$     | <code>\leftarrowtail</code>     | $\rightarrowtail$      | <code>\rightarrowtail</code>      |
| $\leftrightharpoons$ | <code>\leftrightharpoons</code> | $\rightleftharpoons$   | <code>\rightleftharpoons</code>   |
| $\Lsh$               | <code>\Lsh</code>               | $\Rsh$                 | <code>\Rsh</code>                 |
| $\looparrowleft$     | <code>\looparrowleft</code>     | $\looparrowright$      | <code>\looparrowright</code>      |
| $\curvearrowleft$    | <code>\curvearrowleft</code>    | $\curvearrowright$     | <code>\curvearrowright</code>     |
| $\circlearrowleft$   | <code>\circlearrowleft</code>   | $\circlearrowright$    | <code>\circlearrowright</code>    |
| $\multimap$          | <code>\multimap</code>          | $\Uparrow$             | <code>\Uparrow</code>             |
| $\Downarrow$         | <code>\Downarrow</code>         | $\Uparpoonleft$        | <code>\Uparpoonleft</code>        |
| $\Uparpoonright$     | <code>\Uparpoonright</code>     | $\Downharpoonright$    | <code>\Downharpoonright</code>    |
| $\rightsquigarrow$   | <code>\rightsquigarrow</code>   | $\leftrightsquigarrow$ | <code>\leftrightsquigarrow</code> |

جدول ۱۸.۳: نقیض روابط دوتایی و پیکان‌های  $\text{AMS}$ 

|                 |                 |                     |
|-----------------|-----------------|---------------------|
| $\nless$        | $\ngtr$         | $\varsubsetneqq$    |
| $\lneq$         | $\gneq$         | $\varsupsetneqq$    |
| $\nleq$         | $\ngeq$         | $\subsetneqq$       |
| $\nleqslant$    | $\ngeqslant$    | $\supsetneqq$       |
| $\lneqq$        | $\gneqq$        | $\nmid$             |
| $\lvertneqq$    | $\gvertneqq$    | $\nparallel$        |
| $\nleqq$        | $\ngeqq$        | $\nshortmid$        |
| $\lnsim$        | $\gnsim$        | $\nshortparallel$   |
| $\lnapprox$     | $\gnapprox$     | $\nsim$             |
| $\nprec$        | $\nsucc$        | $\ncong$            |
| $\npreceq$      | $\nsucceq$      | $\nvdash$           |
| $\precneqq$     | $\succneqq$     | $\nvDash$           |
| $\precnsim$     | $\succnsim$     | $\nVdash$           |
| $\precnapprox$  | $\succnapprox$  | $\nVDash$           |
| $\subsetneq$    | $\supsetneq$    | $\ntriangleleft$    |
| $\varsubsetneq$ | $\varsupsetneq$ | $\ntriangleright$   |
| $\subsetneqq$   | $\supsetneqq$   | $\ntrianglelefteq$  |
| $\subsetneqq$   | $\supsetneqq$   | $\ntrianglerighteq$ |
| $\nleftarrow$   | $\rightarrow$   | $\nleftrightarrow$  |
| $\nLeftarrow$   | $\nrightarrow$  | $\nLeftrightarrow$  |



جدول ۱۹.۳: متفرقه  $\mathcal{AMS}$

|             |                            |                      |                                 |                   |                          |
|-------------|----------------------------|----------------------|---------------------------------|-------------------|--------------------------|
| $\hbar$     | <code>\hbar</code>         | $\hbar$              | <code>\hslash</code>            | $\mathbb{k}$      | <code>\Bbbk</code>       |
| $\square$   | <code>\square</code>       | $\blacksquare$       | <code>\blacksquare</code>       | $\textcircled{S}$ | <code>\circledS</code>   |
| $\triangle$ | <code>\vartriangle</code>  | $\blacktriangle$     | <code>\blacktriangle</code>     | $\complement$     | <code>\complement</code> |
| $\nabla$    | <code>\triangledown</code> | $\blacktriangledown$ | <code>\blacktriangledown</code> | $\Game$           | <code>\Game</code>       |
| $\lozenge$  | <code>\lozenge</code>      | $\blacklozenge$      | <code>\blacklozenge</code>      | $\bigstar$        | <code>\bigstar</code>    |
| $\angle$    | <code>\angle</code>        | $\sphericalangle$    | <code>\measuredangle</code>     |                   |                          |
| $\diagup$   | <code>\diagup</code>       | $\diagdown$          | <code>\diagdown</code>          | $\backprime$      | <code>\backprime</code>  |
| $\nexists$  | <code>\nexists</code>      | $\Finv$              | <code>\Finv</code>              | $\varnothing$     | <code>\varnothing</code> |
| $\eth$      | <code>\eth</code>          | $\sphericalangle$    | <code>\sphericalangle</code>    | $\mho$            | <code>\mho</code>        |



## فصل ۴

# ابزارهای ویژه

وقتی که در حال تهیه یک نوشتار بزرگ هستید، لاتک با ارائه ابزارهای ویژه‌ای مانند تولید نمایه، کتاب‌نامه، و غیره به شما کمک می‌کند. لیست کامل‌تری از ابزارهایی که در لاتک وجود دارد در [۱] و [۲] ارائه شده است.

### ۱.۴ الصاق بسته‌های پست‌اسکرپت

لاتک ابزارهای ابتدایی کار با اشیاء شناور مانند تصویر و گرافیک را با محیط‌های `table` و `figure` ارائه می‌کند. چندین راه برای تولید گرافیک واقعی توسط خود لاتک بوسیله بسته‌هایی وجود دارد که تعدادی از آنها در فصل ۵ بیان شده است. برای اطلاعات بیشتر به [۱] و [۳] مراجعه کنید.

یک راه ساده‌تر برای داشتن گرافیک در یک نوشتار این است که تصاویر را به وسیله نرم‌افزارهایی<sup>۱</sup> تولید کرد و آنگاه آنها را در نوشتار وارد کرد. لاتک راه‌های بسیاری برای انجام این کار در اختیار شما قرار می‌دهد، اما این مقدمه تنها استفاده از کپسول پست‌اسکرپت<sup>۲</sup> را شرح می‌دهد، زیرا کار با آن بسیار آسان و معمول است. برای این که تصاویر را به فرمت ای.پی.اس در بیاورید باید چاپگر پست‌اسکرپت داشته باشید.<sup>۳</sup>

چندین فرمان، مناسب الصاق یک تصویر به نوشتار در بسته `graphicx` موجود است که توسط D. P. Carlisle تهیه شده است. این بسته قسمتی از یک خانواده از بسته‌هاست که کلاف `graphics` نامیده می‌شود.<sup>۴</sup>

با فرض آنکه روی سیستمی کار می‌کنید که به چاپگر پست‌اسکرپت مجهز و بسته `graphicx` نصب شده است، گام‌های زیر شما را در الصاق تصویر به نوشتارتان یاری می‌کند:

---

<sup>۱</sup>مانند XFig، Gnuplot، ...

<sup>۲</sup>Encapsulated POSTSCRIPT

<sup>۳</sup>گزینه دیگر استفاده از نرم‌افزار GHOSTSCRIPT است که آن را می‌توانید از `support/ghostscript` تهیه کنید. کاربران ویندوز و OS/2 ممکن است نیاز داشته باشند به GSVIEW نگاهی بیندازند.

<sup>۴</sup>`macros/latex/required/graphics`

(۱) تصویر مورد نظر را از برنامه‌ی ای.پی.اس مربوطه به فرمت EPS خارج کنید.<sup>۵</sup>

(۲) بسته‌ی graphicx را در سرآغاز فایل به شکل زیر فراخوانی کنید،

```
\usepackage [driver] {graphicx}
```

که *driver* نام مبدل دی.وی.آی به پستاسکریپت است. مبدلی که بسیار مورد استفاده همگان قرار می‌گیرد مبدل dvips است. نام درایور مورد نیاز است، زیرا هیچ استاندارد برای الصاق یک تصویر در تک وجود ندارد. با دانستن نام درایور، بسته‌ی graphicx روش درست الصاق تصویر را در فایل dvi. به کار می‌بندد، و بنابراین چاپگر به شکل درست می‌تواند فایل eps. را تولید کند.

(۳) فرمان

```
\includegraphics [key=value,...] {file}
```

را به کار گیرید تا فایل تصویر را در نوشتار خود وارد کنید. پارامتر اختیاری لیستی از کلیدهای جداشده توسط ویرگول را قبول می‌کند و مقادیر مورد نظر را تنظیم می‌کند. کلیدها را می‌توان برای تغییر عرض و ارتفاع، و چرخاندن تصویر به کار برد. جدول ۱.۴ مهمترین کلیدها را نشان می‌دهد.

جدول ۱.۴: نام کلیدها برای بسته‌ی graphicx

|                         |        |
|-------------------------|--------|
| تنظیم عرض تصویر         | width  |
| تنظیم ارتفاع تصویر      | height |
| چرخش تصویر پاد ساعت‌گرد | angle  |
| تنظیم اندازه تصویر      | scale  |

مثال زیر به شرح مطالب گفته شده کمک می‌کند:

```
\begin{figure}
\centering
\includegraphics [angle=90,
width=0.5\textwidth] {test}
```

<sup>۵</sup> اگر از برنامه‌ی ای.پی.اس مربوطه نمی‌توانید تصویر را به فرمت ای.پی.اس خارج کنید، سعی کنید چاپگر ای.پی.اس (مانند Apple LaserWriter) را نصب کنید و خروجی آن را به فایل قرار دهید. اگر خوش‌شانس باشید تصویر به فرمت ای.پی.اس ذخیره خواهد شد. توجه داشته باشید که یک تصویر ای.پی.اس نباید بیش از یک صفحه باشد. بعضی از چاپگرها را می‌توان تنظیم کرد که خروجی خود را به فرمت ای.پی.اس تولید کنند.

```
\caption{This is a test.}
\end{figure}
```

این فرمان تصویر ذخیره شده در `test.eps` را به نوشتار الصاق می‌کند. تصویر در ابتدا به اندازه ۹۰ درجه چرخش می‌یابد و سپس در انتها به اندازه نصف عرض پاراگراف تنظیم می‌شود. نسبت تنظیم است زیرا هیچ ارتفاعی مشخص نشده است. پارامترهای عرض و ارتفاع را می‌توان به‌طور صریح مشخص کرد (نه بر حسب چیز دیگر مانند عرض پاراگراف). برای اطلاعات بیشتر به جدول ۵.۶ در صفحه ۱۱۷ مراجعه کنید. اگر می‌خواهید اطلاعات کاملی در این مورد داشته باشید [۹] و [۱۳] را مطالعه کنید.

## ۲.۴ کتاب‌نامه

کتاب‌نامه را می‌توان با محیط `thebibliography` تولید کرد. هر فقره را می‌توان با فرمان

```
\bibitem[label]{marker}
```

درست کرد. در این صورت از `marker` می‌توان برای ارجاع به یک کتاب یا مقاله در داخل نوشتار استفاده کرد.

```
\cite{marker}
```

اگر نمی‌خواهید از گزینه `label` استفاده کنید، هر فقره به‌طور خودکار شماره‌گذاری می‌شود. پارامتر بعد از `\begin{thebibliography}` مشخص می‌کند که چه مقدار فضا باید برای برچسب‌ها در نظر گرفته شود. در مثال زیر، {99} به لاتک می‌گوید که هیچ‌کدام از شماره‌های فقره‌ها گسترده‌تر از عدد 99 نیست.

```
Part1~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Part1:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Part1 [1] has proposed that ...

# Bibliography

[1] H. Part1: *German T<sub>E</sub>X*, TUGboat Volume 9, Issue 1 (1988)

برای پروژه‌های بزرگ‌تر، ممکن است مایل باشید برنامه BibTeX را ببینید. BibTeX با اغلب توزیع‌های تک ارائه می‌شود. این برنامه به شما اجازه می‌دهد که پایگاهی از مراجع را تهیه کنید و آنهایی را که لازم دارید در یک نوشتار وارد کنید. فرمی که BibTeX برای ذخیره مراجع ارائه می‌کند به صورتی است که می‌توانید انواع مختلف مرجع را به‌طور یکسان ذخیره کنید.

### ۳.۴ نمایه سازی

یکی از امکانات بسیار خوب اغلب کتاب‌ها نمایه است. به کمک برنامه `makeindex` لاتک قادر است به سادگی هرچه تمام‌تر نمایه تولید کند. این مقدمه تنها فرمان‌های ابتدایی نمایه‌سازی را شرح می‌دهد. برای شرح کامل‌تر به [۳] مراجعه کنید. برای این که لاتک را قادر به ساختن نمایه کنیم باید بسته `makeidx` را در سرآغاز به صورت زیر فراخوانی کنیم:

```
\usepackage{makeidx}
```

و فرمان ویژه نمایه‌سازی باید به صورت

```
\makeindex
```

در سرآغاز فعال شود.

محتویات یک نمایه با فرمان

```
\index{key}
```

مشخص می‌شود، که `key` فقره نمایه است. فرمان نمایه را در مکانی از متن وارد می‌کنید که می‌خواهید نمایه به آنجا ارجاع داشته باشد. جدول ۲.۴ شکل آرگومان `key` را با چندین مثال نشان می‌دهد. وقتی که فایل ورودی با لاتک پردازش می‌شود، هر فرمان `\index` فقره مربوطه را به همراه شماره صفحه جاری در یک فایل ویژه ذخیره می‌کند. این فایل دارای همان نام فایل ورودی است، اما پسوند آن `(.idx)` است. این فایل `idx` را سپس می‌توان با برنامه `makeindex` پردازش کرد.

```
makeindex filename
```

برنامه `makeindex` نمایه مرتب شده را در فایل هم‌نام با فایل ورودی ولی با پسوند `.ind` تولید می‌کند. بعد از این کار اگر فایل ورودی دوباره پردازش شود، نمایه مرتب شده در نقطه‌ای از نوشتار که

<sup>۶</sup>در سیستم‌هایی که نام یک فایل نمی‌تواند بیشتر از ۸ حرف باشد، نام این برنامه `makeidx` است.

جدول ۲.۴: مثال‌هایی از شکل کلیدها

| مثال                                  | فقرهٔ نمایه | توضیح               |
|---------------------------------------|-------------|---------------------|
| <code>\index{hello}</code>            | hello, 1    | فقرهٔ ساده          |
| <code>\index{hello!Peter}</code>      | Peter, 3    | زیرفقره زیر 'hello' |
| <code>\index{Sam@\textsl{Sam}}</code> | Sam, 2      | فقرهٔ شکیل          |
| <code>\index{Lin@\textbf{Lin}}</code> | Lin, 7      | همانند بالا         |
| <code>\index{Jenny textbf}</code>     | Jenny, 3    | شمارهٔ صفحهٔ شکیل   |
| <code>\index{Joe textit}</code>       | Joe, 5      | همانند بالا         |
| <code>\index{ecole@\'ecole}</code>    | école, 4    | اعمال لهجه          |

فرمان

```
\printindex
```

قرار داشته باشد ظاهر می‌شود.

بستهٔ `showidx` که به همراه لاتک عرضه می‌شود تمام فقره‌های نمایه را در حاشیهٔ سمت چپ متن ظاهر می‌کند. این کار برای اصلاح و بازدید مکان دقیق فقره‌های نمایه بسیار مفید است. توجه کنید که فرمان `\index` اگر به‌طور دقیق مورد استفاده قرار نگیرد ممکن است صفحه‌بندی را تحت تأثیر قرار دهد.

My Word `\index{Word}`. As opposed to `Word\index{Word}`. Note the position of the full stop.

My Word . As opposed to Word. Note the position of the full stop.

## ۴.۴ سربرگ‌های تجملی

بستهٔ `fancyhdr`<sup>۷</sup>، فرمان‌هایی ساده برای طراحی سربرگ و ته‌برگ برای نوشتار ارائه می‌کند. اگر به قسمت بالای این صفحه نگاه کنید، می‌توانید اثر این بسته را ببینید. مطلب اصلی در طراحی سربرگ و ته‌برگ این است که چگونه نام فصل و بخش جاری را ظاهر کنیم. لاتک این مشکل را با دو روش برطرف می‌کند. در تعریف سربرگ و ته‌برگ، می‌توانید از فرمان‌های `\leftmark` و `\rightmark` برای چاپ عنوان فصل و بخش استفاده کنید. مقدار این دو فرمان وقتی که فرمان‌های فصل جدید و بخش جدید قرار دارند دوباره‌سازی می‌شوند.

<sup>۷</sup>نوشته شده توسط Piet van Oostrum و قابل دریافت از `macros/latex/contrib/supported/fancyhdr`.

---

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% with this we ensure that the chapter and section
% headings are in lowercase.
\renewcommand{\chaptermark}[1]{%
 \markboth{#1}{} }
\renewcommand{\sectionmark}[1]{%
 \markright{\thesection\ #1} }
\fancyhf{} % delete current header and footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % space for the rule
\fancypagestyle{plain}{%
 \fancyhead{} % get rid of headers on plain pages
 \renewcommand{\headrulewidth}{0pt} % and the line
}
```

---

شکل ۱.۴: مثال بارگذاری fancyhdr



برای حداکثر انعطاف‌پذیری، فرمان `\chapter` و دوستانش به‌طور خودکار مقدمات `\rightmark` و `\leftmark` را تغییر نمی‌دهند. فرمان‌های

`\chaptermark`, `\sectionmark`, `\subsectionmark`

هستند که وظیفه تعریف دوباره `\rightmark` و `\leftmark` را دارند.

اگر می‌خواهید شکل قرارگرفتن عنوان فصل را در سربرگ تغییر دهید، کافی است تنها `\chaptermark` را به‌کار ببرید.

شکل ۱.۴ بارگذاری‌های ممکن بسته `fancyhdr` را نشان می‌دهد که شکل سربرگ و ته‌برگ همانند این مقدمه باشد. در هر حال، توصیه می‌کنم که راهنمای این بسته را که در پانویشت آمده است به‌طور کامل مطالعه کنید.

## ۵.۴ بسته Verbatim

در بخش‌های پیشین احتمالاً با محیط `verbatim` آشنا شده‌اید. در این بخش، با بسته `verbatim` آشنا می‌شوید. بسته `verbatim` اساساً گسترشی از محیط `verbatim` است که تعدادی از مشکلات این محیط را برطرف می‌کند. این به تنهایی کار خیلی خارق‌العاده‌ای نیست، اما این گسترش چندین ابزار جدید تعریف می‌کند، که به همین دلیل این بسته را در اینجا توضیح می‌دهم. بسته `verbatim` فرمان

```
\verbatiminput{filename}
```

را ارائه می‌کند، که شما را قادر به الصاق یک متن اسکی در نوشتار خود می‌کند که این متن اسکی باید در محیط `verbatim` قرار داشته باشد.

از آنجا که بسته `verbatim` قسمتی از کلاف ابزار است، باید روی سیستم شما نصب شده باشد. اگر می‌خواهید اطلاعات بیشتری در مورد این بسته بدست بیاورید حتماً [۱۰] را ببینید.

## ۶.۴ نصب بسته‌های اضافی

اکثر توزیع‌های لاتک شامل بسیاری از بسته‌ها است که هنگام نصب لاتک به‌طور خودکار نصب می‌شوند، با این حال تعداد بسیار بیشتری از بسته‌ها را می‌توان روی اینترنت پیدا کرد. مهم‌ترین مکان روی اینترنت برای دستیابی به این بسته‌ها CTAN (<http://www.ctan.org/>) است.

بسته‌هایی مانند `geometry`, `hyphenat`، و بسیاری بیشتر از این بسته‌ها به‌طور عمومی از دو فایل تشکیل شده‌اند: یکی با پسوند `.ins` و دیگری با پسوند `.dtx`. اغلب فایل‌ها با نام `readme.txt` نیز وجود دارد که شامل شرحی از بسته است. بهتر است همواره این فایل را مطالعه کنید.

اگر فردی فایل‌های یک بسته را در سیستم شما ذخیره کرده باشد، لازم است که آنها را پردازش کنید تا توزیع تک این بسته را بشناسد و راهنمای آن را در اختیار شما قرار دهد. اولین قدم به‌صورت زیر انجام می‌شود:

۱. لاتک را روی فایل `.ins` پردازش کنید. این کار باعث باز کردن فایل `.sty` می‌شود.

۲. فایل `.sty` را به مکانی انتقال دهید تا توزیع تک شما قادر به پیدا کردن آن باشد. معمولاً این مکان در `.../localtexmf/tex/latex` قرار دارد (کاربران ویندوز و OS/2 می‌توانند از بک‌اسلش به جای اسلش استفاده کنند).

۳. پایگاه نام - فایل توزیع خود را بروز کنید. فرمان انجام این کار به توزیع تک شما بستگی دارد: `texhash` در `TeX` و `fpTeX`؛ `mktexlsr` در `web2c`؛ و `update-fndb` - `initexmf` در `MikTeX` و یا از رابط گرافیکی کاربر مربوطه استفاده کنید.

حال می‌توانید راهنمای بسته را از فایل `.dtx` بدست آورید:

۱. لاتک را روی فایل `.dtx` پردازش کنید. این کار باعث تولید یک فایل `.dvi` می‌شود. توجه داشته باشید که باید لاتک را روی فایل چند بار اجرا کنید تا ارجاع‌های متن به‌درستی نمایش داده شوند.

۲. بررسی کنید که آیا لاتک فایل `.idx` را تولید کرده است یا نه. اگر این اتفاق نیفتاده بود به مرحله آخر <sup>۵</sup> بروید.

۳. برای تولید نمایه، عبارت زیر را وارد کنید:

```
makeindex -s gind.ist name
```

(که `name` همان نام فایل اصلی بدون هیچ پسوندی است).

۴. لاتک را دوباره روی فایل `.dtx` پردازش کنید.

۵. فایل `.ps` یا `.pdf` را برای لذت بیشتر از مطالعه ایجاد کنید.

گاهی اوقات می‌بینید که فایل `.glo`<sup>۸</sup> ایجاد شده است. فرمان زیر را بعد از مرحله <sup>۴</sup> و قبل از مرحله <sup>۵</sup> اجرا کنید:

```
makeindex -s gglo.ist -o name.gls name.glo
```

مطمئن شوید که لاتک را روی فایل `.dtx` یکبار دیگر اجرا کنید قبل از آنکه به مرحله <sup>۵</sup> بروید.

## ۷.۴ کار با پی.دی.اف لاتک

پی.دی.اف یک فرمت ابرمتن<sup>۹</sup> است. همانند صفحه‌های وب، بعضی از کلمات دارای ابرارجاع هستند. این کلمات به مکان‌های دیگری در نوشتار اشاره می‌کنند. اگر به این کلمه‌ها اشاره کنیم به مکان دیگری از متن انتقال می‌یابیم. به زبان لاتک، این موضوع به آن معنا است که هر ارجاع `\ref` و `\pageref`

<sup>۸</sup>glossary

<sup>۹</sup>hypertext

یک ابراراجاع می‌شود. به همین ترتیب تمام جدول‌ها، فهرست مطالب، فقره‌های نمایه و تمام اشیاء مانند اینها می‌توانند ابراراجاع باشند.

بیشتر صفحه‌های وب که امروزه نوشته می‌شوند به صورت HTML<sup>۱۰</sup> است. این فرمت دو ویژگی مهم برای نوشتن متن‌های علمی دارد:

۱. وارد کردن فرمول‌های ریاضی در متن‌های HTML عموماً پشتیبانی نمی‌شود. با این که استاندارد برای نوشتن فرمول در این فرمت وجود دارد، بسیاری از مرورگرهای امروزی از آن پشتیبانی نمی‌کنند، یا این که قلم‌های مورد نیاز را نمی‌شناسند.

۲. چاپ متن‌های HTML امکان‌پذیر است، اما نتیجه کار کاملاً به مرورگرها و سیستم‌عامل‌ها بستگی دارد. نتیجه چاپ بسیار با چیزی که در دنیای لاتک انتظار داریم متفاوت است.

تلاش‌های بسیاری برای تولید مترجم‌هایی از لاتک به HTML وجود دارد. بعضی از آنها حتی بسیار کارا هستند به این معنی که می‌توانند متن‌های مناسب وب از فایل‌های لاتک بسازند. اما همه آنها حاشیه‌های چپ و راست متن را می‌برند. همینکه شروع کنید متن‌های پیچیده با فراخوانی بسته‌های مختلف تولید کنید همه چیز خراب می‌شود. نویسندگانی که می‌خواهند نوشته آنها بدون تغییر در وب گذاشته شود، نوشته خود را ابتدا به صورت پی.دی.اف (PDF) تبدیل می‌کنند که به این ترتیب چهارچوب متن و ابرمتن بدون تغییر باقی می‌ماند. بعضی از مرورگرها به ابزار نمایش مستقیم صفحات پی.دی.اف مجهز هستند.

با وجود آنکه نمایشگر دی.وی.آی و پی.اس برای تقریباً تمام سیستم‌ها وجود دارد، نمایشگرهای Acrobat Reader و Xpdf برای مشاهده فایل‌های پی.دی.اف بسیار پیشرفته هستند. بنابراین تولید نسخه پی.دی.اف از فایل برای استفاده کنندگان بسیار مفید است.

#### ۱.۷.۴ نوشتارهای پی.دی.اف برای وب

تولید نسخه پی.دی.اف از کد لاتک توسط پی.دی.اف تک<sup>۱۱</sup> بسیار آسان است. پی.دی.اف تک برنامه‌ای است که توسط Hàn Thế Thành نوشته شده است. پی.دی.اف تک خروجی پی.دی.اف تولید می‌کند در حالی که تک خروجی دی.وی.آی تولید می‌کند.

هر دو برنامه پی.دی.اف تک و پی.دی.اف لاتک به‌طور خودکار توسط بسیاری از توزیع‌های تک نصب می‌شود، مانند  $\text{TeXLive}$ ،  $\text{MikTeX}$ ،  $\text{fpTeX}$ ،  $\text{teTeX}$  و  $\text{CMacTeX}$ .

برای تولید خروجی پی.دی.اف به جای دی.وی.آی، تنها باید فرمان `pdflatex file.tex` را به جای `latex file.tex` به کار برد. در سیستم‌هایی که لاتک را نمی‌توان از خط فرمان اجرا کرد، می‌توانید کلید مخصوص این کار را از مرکز فرمان تک پیدا کنید.

با لاتک می‌توانید اندازه صفحه را با گزینه‌هایی در نوشتار مشخص کنید مانند `a4paper` یا `letterpaper`. این روش در پی.دی.اف لاتک نیز کارساز است، قبل از این، پی.دی.اف لاتک باید اندازه واقعی صفحه را بداند. اگر از بسته `hyperref` استفاده می‌کنید (صفحه ۷۹ را ببینید)، اندازه صفحه به‌طور خودکار تعیین می‌شود. در غیر این صورت این کار را باید دستی به صورت زیر انجام دهید:

<sup>۱۰</sup>HyperText Markup Language

<sup>۱۱</sup>pdfTeX

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

بخش بعد به طور مفصل تر به تفاوت لاتک و پی.دی.اف لاتک می‌پردازد. مهمترین تفاوت‌ها عبارتند از قلم‌ها، نوع تصاویر الصاقی، و تنظیم دستی ابرمتن‌ها.

#### ۲.۷.۴ قلم‌ها

پی.دی.اف. لاتک می‌تواند با هر نوع قلم کار کند،<sup>۱۲</sup> اما قلم‌های نرمال لاتک، پی.کی بیت‌مپ‌ها، بعد از تبدیل به پی.دی.اف و هنگام مشاهده با آکروبات ریدر به صورت زشتی پدیدار می‌شوند. برای رفع این مشکل بهتر است از قلم‌های پی.کی بیت‌مپ نوع ۱ برای تولید نوشتار استفاده کرد. توزیع‌های جدید تک طوری نصب می‌شوند که این کار به صورت خودکار انجام شود. بهتر است این موضوع را بررسی کنید. اگر این گونه است تمام این بخش را نادیده بگیرید.

#### ۳.۷.۴ استفاده از گرافیک

الصاق تصاویر در یک نوشتار به شکل خوبی توسط بستهٔ `graphicx` انجام می‌شود (صفحهٔ ۶۹ را ببینید). با استفاده از گزینهٔ درایور `pdftex` این بسته با لاتک نیز کار می‌کند:

```
\usepackage[pdftex]{color,graphicx}
```

در مثال سادهٔ بالا گزینهٔ رنگ را نیز وارد کرده‌ام، زیرا استفاده از تصاویر رنگی در وب بسیار معمول است.

این خبر خوب بود. و حالا خبر بد این است که تصاویر به فرم ای.پی.اس با پی.دی.اف لاتک سازگار نیستند. اگر پسوند فایل را در فرمان `\includegraphics` اعلان نکنید، فرمان `graphicx` بدنبال فرمت مناسب خود، به ترتیب گزینه‌های درایور می‌گردد. برای پی.دی.اف تک فرمت‌های تصویر مناسب عبارتند از `.png`، `.pdf`، `.jpg` و `.mps` (۱۳) اما فرمت `.eps` از این نوع نیست.

راه سادهٔ رفع این مشکل این است که با استفاده از فرمان `epstopdf` تصاویر ای.پی.اس را به پی.دی.اف تبدیل کرد. برای تصاویر بُرداری این روش بسیار مناسب است. برای تصاویر بیت‌مپ، این روش ایده‌آل نیست، زیرا فرمت پی.دی.اف به طور طبیعی الصاق تصاویر پی.ان.جی و جی.پی.ای.جی را پشتیبانی می‌کند. پی.ان.جی برای تصاویر با تعداد کمی رنگ مناسب است و جی.پی.ای.جی برای تصاویر کامل‌تر مناسب است و بسیار کم حجم است.

حتی بسیار مناسب است که تصاویر هندسی را رسم نکرد و تنها با استفاده از فرمان‌هایی این تصاویر را در نوشتار قرار داد. زبان مناسب انجام این کار متا‌پست است، که در تمام توزیع‌های تک وجود دارد و دارای راهنمای کامل است.

<sup>۱۲</sup>مانند TrueType، PK bitmaps، PostScript type 1، ...

<sup>۱۳</sup>METAPOST

## ۴.۷.۴ ارجاع متنی

بستهٔ `hyperref` مسئولیت برگردان تمام ارجاعات داخلی متن را به ابرارجاع دارد. برای انجام این کار به کمی شعبده‌بازی احتیاج است، شما باید فرمان `\usepackage[pdf]{hyperref}` را به عنوان آخرین فرمان در سرآغاز نوشتار خود قرار دهید. چندین گزینه برای تغییر رفتار بستهٔ `hyperref` وجود دارد:

- به صورت تعدادی گزینه بعد از گزینهٔ `pdf` که با `pdf` ویرگول جدا می‌شوند

```
\usepackage[pdf]{hyperref}
```

- یا در یک خط جداگانه با استفاده از فرمان

```
\hypersetup{options}
```

تنها گزینهٔ اجباری `pdf` است؛ بقیهٔ گزینه‌ها اختیاری هستند و اجازهٔ تغییر رفتار ارجاعات را می‌دهند.<sup>۱۴</sup> در مثال زیر مقادیر پیش‌فرض به صورت عادی (غیر ایتالیک) نوشته شده‌اند.

`bookmarks (=true,false)` میلهٔ چوب الف را نمایش می‌دهد.

`unicode (=false,true)` اجازهٔ نمایش حروف غیر لاتین را در چوب الف آکروبات می‌دهد.

`pdftoolbar (=true,false)` میلهٔ ابزار آکروبات را فعال یا غیر فعال می‌کند.

`pdfmenubar (=true,false)` منوی آکروبات را نمایش می‌دهد.

`pdfwindow (=true,false)` اندازهٔ نمایش را تغییر می‌دهد.

`pdftitle (=text)` عنوانی را که هنگام نمایش فایل در قسمت اطلاعات آکروبات ظاهر می‌شود، نمایش می‌دهد.

`pdfauthor (=text)` عنوان نویسندهٔ فایل پی.دی.اف.

`pdfnewwindow (=true,false)` مشخص می‌کند که آیا باید یک صفحهٔ جدید هنگام نمایش فایل ظاهر شود.

`colorlinks (=false,true)` ارجاعات را در جعبه‌های رنگی محصور می‌کند (`false`) یا خود ارجاعات به صورت رنگی ظاهر می‌شوند (`true`). رنگ این ارجاعات را می‌توان بوسیلهٔ گزینه‌های زیر تنظیم کرد (مقادیر پیش‌فرض رنگی نشان داده می‌شوند):

`linkcolor (=red)` رنگ اتصال‌های داخلی (بخش‌ها، صفحه‌ها و غیره)

<sup>۱۴</sup> قابل ذکر است که بستهٔ `hyperref` در کار با پی.دی.اف تک دارای هیچ محدودیتی نیست. می‌توان آن را تنظیم کرد تا اطلاعات پی.دی.اف را در خروجی دی.وی.آی نیز هنگام پردازش لاتک ذخیره کند و هنگام تبدیل به پی.اس و در نهایت با مبدل آکروبات دیستایلر به فایل پی.دی.اف انتقال یابد.

رنگ ارجاعات `citecolor (=green)` (کتابنامه)

رنگ اتصالها `filecolor (=magenta)`

رنگ اتصالهای وب (ایمیل، وب) `urlcolor (=cyan)`

اگر تنظیمات پیش فرض مناسب کار شماست از فرمان زیر استفاده کنید

```
\usepackage[pdftex]{hyperref}
```

برای این که لیست چوب الف را باز کنید اتصالها را رنگی کنید ( مقدار `=true` اختیاری است):

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

وقتی که نوشتاری را برای چاپ آماده می کنید اتصالهای رنگی مناسب نیستند زیرا هنگام چاپ خاکستری چاپ می شوند که مناسب خواندن نیستند. می توانید از کادرهای رنگی استفاده کنید که هنگام چاپ ظاهر نمی شوند:

```
\usepackage{hyperref}
```

```
\hypersetup{colorlinks=false}
```

یا اتصالها را سیاه کنید:

```
\usepackage{hyperref}
```

```
\hypersetup{colorlinks,%
```

```
citecolor=black,%
```

```
filecolor=black,%
```

```
linkcolor=black,%
```

```
urlcolor=black,%
```

```
pdftex}
```

وقتی که تنها می خواهید اطلاعاتی را در قسمت اطلاعات نوشتار نمایش دهید:

```
\usepackage[pdftitle={Pierre Desproges},%
```

```
pdftitle={Des femmes qui tombent},%
```

```
pdftex]{hyperref}
```

اضافه بر ابرمتنهای خودکار می توانید اتصالهایی را به صورت دلخواه به صورت زیر تعیین کنید

```
\href{url}{text}
```

The `\href{http://www.ctan.org}{CTAN}` website.

متن “CTAN” را تولید می‌کند؛ اشاره به کلمه “CTAN” شما را به وبگاه CTAN راهنمایی می‌کند. اگر مقصد یک اتصال یک صفحه وب نباشد و تنها یک فایل باشد می‌توانید از فرمان `\href` استفاده کنید:

The complete document is `\href{manual.pdf}{here}`

که متن “The complete document is here” را تولید می‌کند. یک اشاره به کلمه “here” فایل `manual.pdf` را باز می‌کند. (مکان فایل وابسته به مکان فایل جاری است). نویسنده یک مقاله ممکن است بخواهد خوانندگان بوسیله ایمیل با او در تماس باشند که این کار با فرمان `\href` درون فرمان `\author` در صفحه اول نوشتار امکان‌پذیر است:

```
\author{Mary Oetiker %<\href{mailto:mary@oetiker.ch}%
{mary@oetiker.ch}$>$}
```

توجه داشته باشید که اتصال به ایمیل را طوری قرار داده‌ام که نه تنها در اتصال ظاهر شده است بلکه در خود صفحه نیز ظاهر می‌شود. این کار را کرده‌ام زیرا اتصال `\href{mailto:mary@oetiker.ch}{Mary Oetiker}`

با آکروبات به خوبی کار می‌کند ولی هنگامی که فایل را چاپ می‌کنیم آدرس ایمیل دیگر ظاهر نمی‌شود.

#### ۵.۷.۴ مشکلات اتصال‌ها

پیغامی همانند

```
! pdfTeX warning (ext4): destination with the same
 identifier (name{page.1}) has been already used,
 duplicate ignored
```

هنگامی ظاهر می‌شود که یک شمارنده از نو مقداردهی شود، به عنوان مثال هنگام استفاده از فرمان `\mainmatter` که توسط طبقه نوشتار کتاب تعریف می‌شود. این فرمان شمارنده صفحه را قبل از اولین فصل کتاب برابر با ۱ می‌کند. ولی از آنجا که اولین صفحه پیشگفتار نیز دارای شماره ۱ است، تمام اتصال‌ها به صفحه ۱ به‌طور یکتا مشخص نمی‌شود، بنابراین توجه داشته باشید شمارنده چندگانه بی‌تاثیر است.

اندازه‌گیر شمارنده‌ها را می‌توان با گزینه `plainpages=false` در گزینه‌های `hyperref` قرار داد. متأسفانه این کار تنها در شماره صفحه‌ها کمک می‌کند. حتی یک راه حل بنیادی می‌تواند استفاده از گزینه `false = hypertexnames` باشد، اما این کار باعث می‌شود اتصال‌های صفحات قابل استفاده نباشند.

## ۶.۷.۴ مشکلات چوب الف

متنی که در چوب الف نمایش داده می‌شود همواره آن چیزی نیست که انتظار آن را دارید. زیرا چوب الفها تنها متن هستند و حروف کمتری برای نمایش آنها نسبت به لاتک موجود است. Hyperref این مشکل را می‌شناسد و پیغام اخطار مناسب می‌دهد:

Package hyperref Warning:

Token not allowed in a PDFDocEncoded string:

می‌توانید این مشکل را با تخصیص یک متن برای چوب الف حل کنید، که جانشین متن مشکل‌دار می‌شود:

```
\texorpdfstring{ \TeX text}{Bookmark Text}
```

عبارات ریاضی به عنوان متن چوب الف دارای این مشکل هستند:

```
\section{\texorpdfstring{$E=mc^2$}%
{E=mc^2}}
```

که باعث می‌شود عبارت `\section{$E=mc^2$}` در چوب الف به صورت “E=mc2” ظاهر شود.

تغییرات رنگ‌ها نیز به خوبی در چوب الف ظاهر نمی‌شوند:

```
\section{\textcolor{red}{Red !}}
```

عبارت “redRed” در چوب الف ظاهر می‌شود. فرمان `\textcolor` نادیده گرفته می‌شود اما آرگومان آن (red) چاپ می‌شود. اگر از فرمان زیر استفاده کنید

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

نتیجه آن خواناتر خواهد بود.

اگر نوشتار خود را در یونیکد بنویسید و گزینه unicode را برای hyperref استفاده کنید آنگاه قادر خواهید بود حروف یونیکد را در چوب الف وارد کنید. این کار شما را قادر می‌سازد حروف بیشتری را موقع استفاده از فرمان `\texorpdfstring` در چوب الف ظاهر کنید.

سازگاری کد بین لاتک و پی.دی.اف لاتک

به‌طور نرمال کد شما با لاتک و پی.دی.اف لاتک پردازش می‌شود. اشکال عمده برای الصاق تصاویر وجود دارد. راه حل ساده این است که پسوند فایل را با فرمان `\includegraphics` تغییر داد. در این



صورت سیستم برای فایل مناسب در پرونده موجود جستجو می‌کند. تنها کاری که باید انجام دهید این است که نسخه مناسب از فایل تصویر را بسازید. در این صورت لاتک بدنبال فایل `.eps` می‌گردد و پی.دی.اف لاتک بدنبال `.png`، `.pdf`، `.jpg` یا `.mps` می‌گردد (به ترتیب).

در حالتی که می‌خواهید کدهای متفاوتی برای نسخه پی.دی.اف و حالت عادی داشته باشید، می‌توانید به راحتی از بسته `ifpdf`<sup>۱۵</sup> در سرآغاز نوشتار خود استفاده کنید. احتمالاً این بسته روی سیستم شما وجود دارد در غیر این صورت میکتک این بسته را برای شما نصب می‌کند. فرمان ویژه `\ifpdf` به شما امکان نوشتن فرمان‌های شرطی را می‌دهد. در این مثال می‌خواهیم نسخه پست‌اسکرپت سیاه و سفید را به خاطر سهولت چاپ بسازیم اما نسخه پی.دی.اف رنگی را برای وب داشته باشیم.

```
\RequirePackage{ifpdf} % running on pdfTeX?
\ifpdf
 \documentclass[a4paper,12pt,pdftex]{book}
\else
 \documentclass[a4paper,12pt,dvips]{book}
\fi

\ifpdf
 \usepackage{lmodern}
\fi

\usepackage[bookmarks, % add hyperlinks
 colorlinks,
 plainpages=false]{hyperref}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[english]{babel}
\usepackage{graphicx}
...
```

در کد بالا بسته `hyperref` را حتی در نسخه غیر پی.دی.اف به کار برده‌ام. تأثیر فرمان `\href` این است که زمان زیادی برای تعریف عبارات شرطی به کار نبریم.

توجه داشته باشید در توزیع‌های جدید تک (به عنوان مثال تک‌لیو) فرمان نرمال، پی.دی.اف لاتک است. این فرمان قادر است به راحتی بین پی.دی.اف و دی.وی.آی تغییر کند. اگر از کد بالا استفاده کنیم، فرمان `pdflatex` خروجی پی.دی.اف و فرمان `latex` خروجی دی.وی.آی را تولید می‌کند.

<sup>۱۵</sup><http://www.tex.ac.uk/cgi-bin/texfaq2html?label=ifpdf>

## ۸.۴ تولید اسلاید

می‌توانید نتایج کارهای علمی خود را با ترانسپارنت روی تخته سیاه نمایش دهید یا مستقیماً با نرم‌افزارهایی با لپ‌تاپ خود آنها را نمایش دهید.

pdf $\LaTeX$  به همراه طبقه beamer به شما امکان تولید اسلاید پی.دی.اف را می‌دهد که حاصل آن شبیه چیزی است که توسط پاورپوینت تولید می‌شود با این تفاوت که بسیار قابل حمل است، زیرا آکروبات ریدر روی اکثر سیستم‌ها وجود دارد.

طبقه beamer از بسته‌های color، graphicx، hyperref و به همراه گزینه‌هایی برای نمایش اسلاید استفاده می‌کند.

وقتی که کد ارائه شده در شکل ۲.۴ را با PDF $\LaTeX$  پردازش می‌کنید یک فایل پی.دی.اف بدست می‌آورد متشکل از یک صفحه عنوان و یک صفحه که در آن چندین آیتم می‌بینید که هر کدام از آنها با مرور فایل به ترتیب ظاهر می‌شوند.

یکی از دستاوردهای طبقه beamer این است که فایل پی.دی.اف تولید می‌کند که به صورت مستقیم قابل استفاده است و نیازی نیست مانند طبقه prosper به یک مرحله میانی پست‌اسکرپت رفت یا این که از بسته ppower4 استفاده کرد.

با استفاده از طبقه beamer می‌توانید نسخه‌های مختلفی از نوشتار خود بسازید. فایل ورودی می‌تواند شامل راه‌کارهایی برای انواع مختلف خروجی باشد که در گزینه طبقه در براکت قرار می‌گیرند. کارهای زیر امکان‌پذیر است.

beamer برای نمایش پی.دی.اف که در بالا توضیح داده شد.

trans برای اسلاید.

handout برای نسخه مناسب چاپ.

نوع پیش‌فرض beamer است، می‌توانید آن را با فعال کردن گزینه‌های دیگر غیر فعال کنید مانند `\documentclass[10pt,handout]{beamer}` که خروجی را مناسب چاپ طراحی می‌کند. شمای نوشتار شما وابسته به این است که چه نسخه‌ای را انتخاب کنید. می‌توانید یکی از شماهایی را که این طبقه فراهم کرده است استفاده کنید یا یک شما برای خودتان طراحی کنید. راهنمای طبقه را در `beameruserguide.pdf` ببینید.

اجازه دهید نگاهی دقیق‌تر به کد شکل ۲.۴ ببندیم. برای نسخه نمایشی `\mode<beamer>` شمای *Goettingen* را انتخاب کرده‌ایم تا پنل مرور را در فهرست مطالب وارد کرده باشیم. گزینه‌ها ما را قادر می‌سازند تا عرض پنل (۲۲ میلیمتر در این حالت) و مکان آن را تعیین کنیم (در سمت راست نوشتار). گزینه `hideothersubsections`، عنوان فصل را نمایش می‌دهد و تنها عنوان زیربخش جاری را نمایش می‌دهد. چیز ویژه‌ای برای تم‌های `\mode<trans>` و `\mode<handout>` وجود ندارد. آنها نوشتار را به شکل استاندارد خود نمایش می‌دهند.

فرمان‌های `\title{}`، `\author{}`، `\institute{}`، و `\titlegraphic{}` محتویات جلد را مشخص می‌کنند. گزینه‌های اختیاری `\title[ ]{}` و `\author[ ]{}` اجازه می‌دهند شکل ویژه‌ای از عنوان و نویسنده را در پنل *Goettingen* قرار دهید.

```
\documentclass[10pt]{beamer}
\mode<beamer>{%
 \usetheme[hideothersubsections,
 right,width=22mm]{Goettingen}
}

\title{Simple Presentation}
\author[D. Flipo]{Daniel Flipo}
\institute{U.S.T.L. \& GUTenberg}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
 \titlepage
\end{frame}

\section{An Example}

\begin{frame}
 \frametitle{Things to do on a Sunday Afternoon}
 \begin{block}{One could \ldots}
 \begin{itemize}
 \item walk the dog\ldots \pause
 \item read a book\pause
 \item confuse a cat\pause
 \end{itemize}
 \end{block}
 and many other things
\end{frame}
\end{document}
```

شکل ۲.۴: کد نمونه برای طبقه beamer

عنوان و زیرعنوان پنل با فرمان‌های نرمال `\section{}` و `\subsection{}` ایجاد می‌شوند که باید در خارج از محیط `frame` تعریف شوند.

کلیدهای مرورگر کوچک در پایین صفحه نمایش اجازه می‌دهند نوشتار را مرور کنید. حضور آنها ربطی به تم انتخابی ندارد.

محتویات هر اسلاید یا صفحه را باید در یک محیط `frame` قرار داد. هیچ گزینه انتخابی برای این محیط وجود ندارد و امکان انتخاب یک چهارچوب ویژه را برای نسخه‌ای ویژه ارائه می‌دهد. در مثال بالا صفحه اول به خاطر وجود فرمان `<handout:0>` در چاپ ظاهر نمی‌شود.

اکیداً توصیه می‌شود برای هر اسلاید یک عنوان به غیر از عنوان اسلاید تعریف کنید. این کار با فرمان `\frametitle{}` امکان‌پذیر است. اگر یک زیرعنوان لازم است می‌توانید از محیط `block` همانند مثال استفاده کنید. توجه داشته باشید که عنوان فرمان‌های `\section{}` و `\subsection{}` در خروجی ظاهر نمی‌شوند.

فرمان `\pause` در محیط شماره‌گذاری شده اجازه می‌دهد اجزاء را یک‌به‌یک نمایش دهید. برای افکت هر نمایش فرمان‌های `\only`، `\uncover`، `\alt` و `\temporal` را ببینید. در بسیاری از جاها می‌توانید از آکولاد برای تنظیم بیشتر استفاده کنید.

در هر حالت مطمئن شوید راهنمای طبقه `beameruserguide.pdf` را برای بیشترین استفاده مطالعه کنید. این بسته به سرعت در حال پیشرفت است، صفحه اینترنتی این بسته را ملاحظه کنید <http://latex-beamer.sourceforge.net/>.

## فصل ۵

# تولید شکل‌های ریاضی

بسیاری از افراد از لاتک برای حروف‌چینی متن استفاده می‌کنند. اما از آنجا که رهیافت ساختار یافته بسیار مناسب است، لاتک همچنین توانایی تولید تصاویر از فرمان‌های متنی را دارد. به علاوه، چندین گسترش از لاتک امکان انجام این کار را به بهترین شکل فراهم می‌کنند. در این فصل چند نوع از این گسترش‌ها را مطالعه می‌کنیم.

### ۱.۵ مرور

محیط `picture` امکان برنامه‌نویسی برای تولید شکل در لاتک را فراهم می‌کند. توضیح کامل را در [۱] ببینید. از یک طرف، چندین محدودیت وجود دارد که از آن جمله محدودیت شیب خط‌ها و شعاع دایره‌ها است. از طرف دیگر، محیط `picture` از لاتک به همراه فرمان `\qbezier` همراه است، “q” به معنای “quadratic”<sup>۱</sup> است. بسیاری از خم‌ها مانند دایره، بیضی، یا ترکیبی از این خم‌ها را می‌توان با تقریب خم‌های درجه دوم بزیه رسم کرد، هرچند که این کار نیازمند محاسبات ریاضی است. به علاوه، اگر یک زبان برنامه‌نویسی مانند جاوا برای تولید بلوک‌های `\qbezier` مورد استفاده قرار گیرد، محیط `picture` بسیار قدرتمند خواهد شد.

با وجود این که نوشتن کد تصاویر در لاتک بسیار محدود کننده و زمان‌بر است، کار با آن هنوز خواستگاه دارد زیرا نوشتار را بسیار کوچک می‌کند و به هیچ فایل تصویری احتیاج ندارد. بسته‌هایی مانند `epic` و `eeepic` (که به عنوان مثال در [۳] توضیح داده شده‌اند)، یا `pstricks` وجود دارند که محدودیت‌های محیط `picture` را ندارند و توان گرافیکی لاتک را به مقدار زیادی قدرت می‌بخشند.

درحالی که دو بسته اولیه تنها محیط `picture` را قدرت می‌بخشند، بسته `pstricks` دارای محیط منحصر به فرد `pspicture` است. قدرت سیستم `pstricks` در این است که این بسته از قابلیت‌های پست‌اسکریپت استفاده می‌کند. به علاوه بسته‌های مختلفی برای کارهای ویژه نوشته شده است. یکی از

<sup>۱</sup> مترجم: به معنای خم درجه دوم است.

این بسته‌ها Xy-pic است که در آخر این فصل توضیح داده شده است. توضیح مفصل‌تری بر این بسته در [۴] ارائه شده است (با [۳] اشتباه نشود).

شاید مهمترین ابزار گرافیکی مربوط به لاتک، متاپست است که به همراه متافونت دوقلوهای دونالد کنوت نام دارند. بر خلاف متافونت، که بیت‌مپ تولید می‌کند، متاپست فایل‌های پست‌اسکرپت تولید می‌کند که می‌توان آنها را به لاتک انتقال داد. برای مقدمه‌ای بر این موضوع به [۱۵]، یا راهنمای [۱۷] مراجعه کنید.

بحث کاملی از استراتژی‌های لاتک و تک برای گرافیک (و قلم‌ها) را می‌توانید در [۱۶] ببینید.

## ۲.۵ محیط تصویر

### ۱.۲.۵ فرمان‌های ابتدایی

یک محیط picture<sup>۲</sup> را می‌توان با دو فرمان زیر بوجود آورد

```
\begin{picture}(x,y)...\end{picture}
```

یا

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

اعداد  $x, y, x_0, y_0$  به `\unitlength` اشاره می‌کنند که می‌توان آنها را با فرمانی به شکل زیر دوباره بارگذاری کرد (ولی این کار را نمی‌توان با محیط picture انجام داد)

```
\setlength{\unitlength}{1.2cm}
```

مقدار پیش‌فرض `\unitlength` برابر 1pt است. زوج  $(x, y)$  اندازه چهارچوب دور تصویر را مشخص می‌کند. زوج اختیاری  $(x_0, y_0)$  مکان گوشه پایین سمت چپ چهارچوب رزرو شده را تعیین می‌کند.

<sup>۲</sup>قبول داشته باشید یا نه، محیط تصویر به‌طور هوشمندانه کار می‌کند، با لاتک استاندارد هیچ بسته‌ای لازم نیست.

بیشتر فرمان‌ها به یکی از دو شکل زیر است

$$\text{\put}(x, y)\{object\}$$

یا

$$\text{\multiput}(x, y)(\Delta x, \Delta y)\{n\}\{object\}$$

خم‌های بزیه از این قاعده مستثنی است. این خم‌ها را می‌توان با فرمان زیر رسم کرد

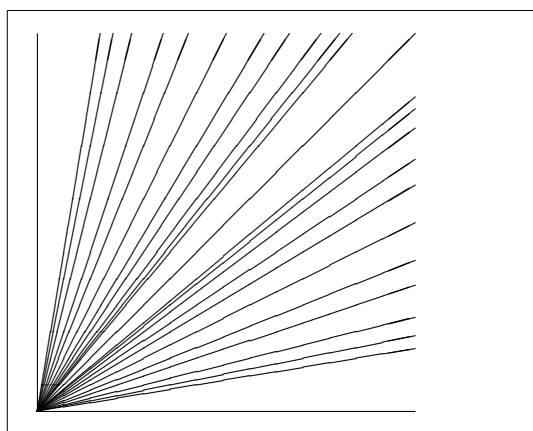
$$\text{\qBezier}(x_1, y_1)(x_2, y_2)(x_3, y_3)$$

## پاره‌خط ۲.۲.۵

```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
 \put(0,0){\line(0,1){1}}
 \put(0,0){\line(1,0){1}}
 \put(0,0){\line(1,1){1}}
 \put(0,0){\line(1,2){.5}}
 \put(0,0){\line(1,3){.3333}}
 \put(0,0){\line(1,4){.25}}
 \put(0,0){\line(1,5){.2}}
 \put(0,0){\line(1,6){.1667}}
 \put(0,0){\line(2,1){1}}
 \put(0,0){\line(2,3){.6667}}
 \put(0,0){\line(2,5){.4}}
 \put(0,0){\line(3,1){1}}
 \put(0,0){\line(3,2){1}}
 \put(0,0){\line(3,4){.75}}
 \put(0,0){\line(3,5){.6}}
 \put(0,0){\line(4,1){1}}
 \put(0,0){\line(4,3){1}}
 \put(0,0){\line(4,5){.8}}
 \put(0,0){\line(5,1){1}}
 \put(0,0){\line(5,2){1}}
 \put(0,0){\line(5,3){1}}
 \put(0,0){\line(5,4){1}}
 \put(0,0){\line(5,6){.8333}}
 \put(0,0){\line(6,1){1}}
 \put(0,0){\line(6,5){1}}
\end{picture}

```



پاره‌خط‌ها را می‌توان با فرمان زیر رسم کرد

```
\put(x,y){\line(x,y){length}}
```

فرمان `\line` دارای دو آرگومان است:

۱. یک بردار جهت‌دار،

۲. یک طول.



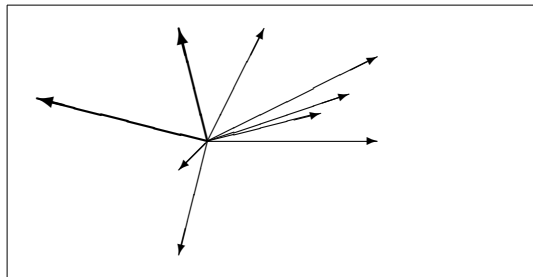
مؤلفه‌های بردار جهت‌دار به چند عدد محدود می‌شود

$$-۶, -۵, \dots, ۵, ۶,$$

و باید نسبت به هم اول باشند (یعنی دارای بزرگترین مقسوم علیه ۱ باشند). در شکل تمام ۲۵ شیب ممکن در یک چهارم اول نمایش داده شده است. طول بستگی به `\unitlength` دارد. آرگومان طول همان مؤلفه افقی است و تنها در حالتی که پاره‌خط عمودی باشد، این آرگومان همان مؤلفه عمودی است.

### ۳.۲.۵ پیکان‌ها

```
\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
 \put(30,20){\vector(1,0){30}}
 \put(30,20){\vector(4,1){20}}
 \put(30,20){\vector(3,1){25}}
 \put(30,20){\vector(2,1){30}}
 \put(30,20){\vector(1,2){10}}
 \thicklines
 \put(30,20){\vector(-4,1){30}}
 \put(30,20){\vector(-1,4){5}}
 \thinlines
 \put(30,20){\vector(-1,-1){5}}
 \put(30,20){\vector(-1,-4){5}}
\end{picture}
```



پیکان‌ها با فرمان زیر رسم می‌شوند

$$\text{\put}(x,y)\{\text{\vector}(x_1,y_1)\{length\}\}$$

برای پیکان‌ها، مؤلفه‌های بردارهای جهت‌دار حتی بیشتر از این محدود هستند و تنها به چند عدد محدود هستند

$$-۴, -۳, \dots, ۳, ۴.$$

این اعداد نیز باید نسبت به هم اول باشند. به تأثیر فرمان `\thicklines` روی دو بردار به سمت چپ توجه داشته باشید.

۴.۲.۵ دایره

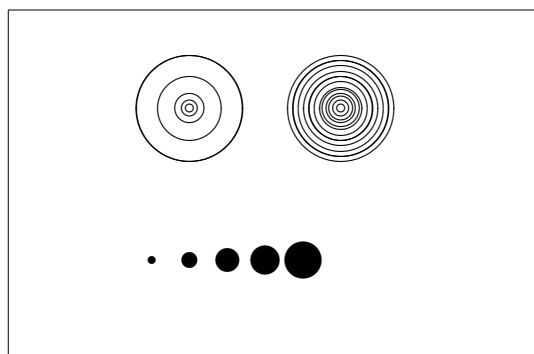
```

\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
 \put(20,30){\circle{1}}
 \put(20,30){\circle{2}}
 \put(20,30){\circle{4}}
 \put(20,30){\circle{8}}
 \put(20,30){\circle{16}}
 \put(20,30){\circle{32}}

 \put(40,30){\circle{1}}
 \put(40,30){\circle{2}}
 \put(40,30){\circle{3}}
 \put(40,30){\circle{4}}
 \put(40,30){\circle{5}}
 \put(40,30){\circle{6}}
 \put(40,30){\circle{7}}
 \put(40,30){\circle{8}}
 \put(40,30){\circle{9}}
 \put(40,30){\circle{10}}
 \put(40,30){\circle{11}}
 \put(40,30){\circle{12}}
 \put(40,30){\circle{13}}
 \put(40,30){\circle{14}}

 \put(15,10){\circle*{1}}
 \put(20,10){\circle*{2}}
 \put(25,10){\circle*{3}}
 \put(30,10){\circle*{4}}
 \put(35,10){\circle*{5}}
\end{picture}

```



فرمان

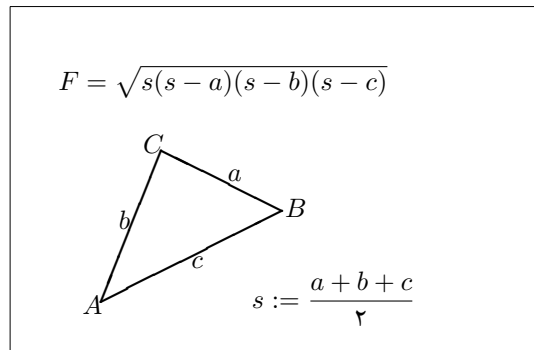
$\text{\put}(x, y)\{\text{\circle}\{diameter\}\}$

یک دایره به مرکز  $(x, y)$  و قطر (نه شعاع)  $diameter$  را رسم می‌کند. محیط `picture` تنها قطرهای تا حداکثر ۱۴ میلی‌متر را می‌پذیرد. فرمان `\circle*` قرص‌ها را تولید می‌کند (دایره‌های توپر). همانند پاره‌خط‌ها، باید از بسته‌های دیگری نیز استفاده کرد، مانند `eepic` یا `pstricks`. برای راهنمایی کامل در مورد این بسته‌ها به [۴] مراجعه کنید.

حالت دیگری نیز در محیط `picture` وجود دارد. اگر از انجام محاسبات ریاضی نمی‌ترسید، دایره‌ها و بیضی‌های دلخواه را می‌توان با خم‌های بزیه به هم چسباند. برای مثال‌هایی از کدهای جاوا به [۱۷] مراجعه کنید.

## ۵.۲.۵ متن و فرمول

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
 \thicklines
 \put(1,0.5){\line(2,1){3}}
 \put(4,2){\line(-2,1){2}}
 \put(2,3){\line(-2,-5){1}}
 \put(0.7,0.3){A}
 \put(4.05,1.9){B}
 \put(1.7,2.95){C}
 \put(3.1,2.5){a}
 \put(1.3,1.7){b}
 \put(2.5,1.05){c}
 \put(0.3,4){F=}
 \sqrt{s(s-a)(s-b)(s-c)}
 \put(3.5,0.4){\displaystyle
 s:=\frac{a+b+c}{2}}
\end{picture}
```



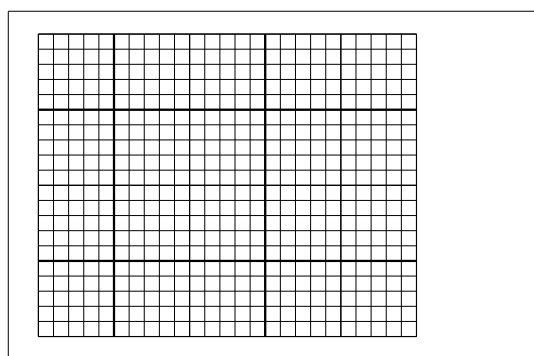
همان‌طور که این مثال نشان می‌دهد، متن و فرمول را می‌توان در محیط `picture` با فرمان `\put` به طریق عادی درج کرد.

## \linethickness و \multiput ۶.۲.۵

```

\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
 \linethickness{0.075mm}
 \multiput(0,0)(1,0){26}%
 {\line(0,1){20}}
 \multiput(0,0)(0,1){21}%
 {\line(1,0){25}}
 \linethickness{0.15mm}
 \multiput(0,0)(5,0){6}%
 {\line(0,1){20}}
 \multiput(0,0)(0,5){5}%
 {\line(1,0){25}}
 \linethickness{0.3mm}
 \multiput(5,0)(10,0){2}%
 {\line(0,1){20}}
 \multiput(0,5)(0,10){2}%
 {\line(1,0){25}}
\end{picture}

```



فرمان

$$\text{\multiput}(x,y)(\Delta x,\Delta y)\{n\}\{object\}$$

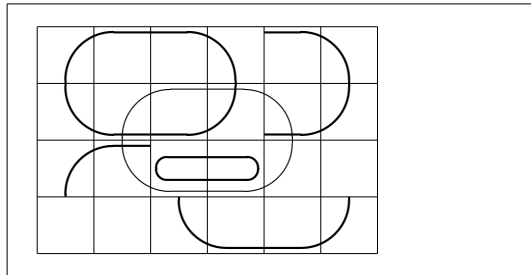
دارای چهار آرگومان است: نقطه شروع، نقطه پایان، بردار انتقال از یک شیء به شیء بعدی، تعداد اشیاء، و شیء که باید رسم شود. فرمان `\linethickness` به پاره‌خط‌های افقی و عمودی تأثیر دارد ولی روی خط‌های اریب و دایره‌ها بی‌تأثیر است. این فرمان مسلماً روی خم‌های بزیه تأثیر دارد!

## ۷.۲.۵ بیضی

```

\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
 \linethickness{0.075mm}
 \multiput(0,0)(1,0){7}%
 {\line(0,1){4}}
 \multiput(0,0)(0,1){5}%
 {\line(1,0){6}}
 \thicklines
 \put(2,3){\oval(3,1.8)}
 \thinlines
 \put(3,2){\oval(3,1.8)}
 \thicklines
 \put(2,1){\oval(3,1.8)[t]}
 \put(4,1){\oval(3,1.8)[b]}
 \put(4,3){\oval(3,1.8)[r]}
 \put(3,1.5){\oval(1.8,0.4)}
\end{picture}

```



فرمان

$$\boxed{\text{\put}(x,y)\{\text{\oval}(w,h)\}}$$

یا

$$\boxed{\text{\put}(x,y)\{\text{\oval}(w,h)[\textit{position}]\}}$$

یک بیضی به مرکز  $(x, y)$  و به عرض  $w$  و ارتفاع  $h$  تولید می‌کند. آرگومان‌های مکان  $position$  که عبارتند از  $r, l, t, b$  به "top" (بالا)، "bottom" (پایین)، "left" (چپ)، و "right" (راست) اشاره دارند و می‌توانند همانند مثال با هم ترکیب شوند.

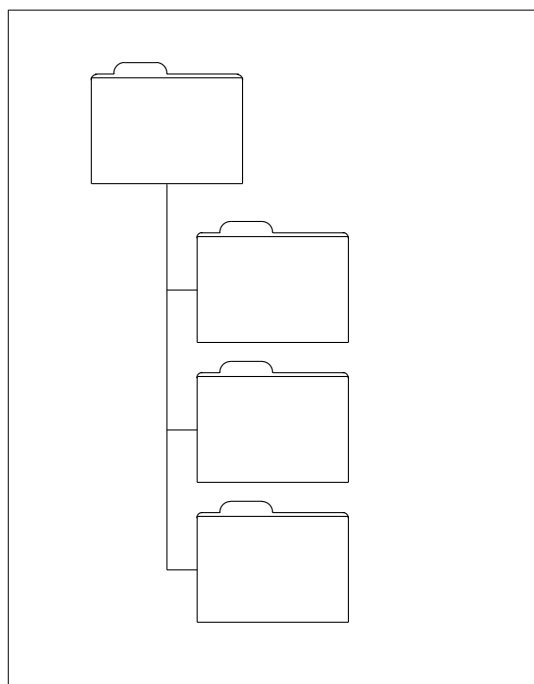
ضخامت خط را می‌توان با دو نوع فرمان کنترل کرد:  $\text{\linethickness}\{length\}$  از یک طرف، و  $\text{\thinlines}$  و  $\text{\thicklines}$  از طرف دیگر  $\text{\linethickness}\{length\}$  فقط به خط‌های افقی و عمودی (و خم‌های درجه دوم بزیه) تأثیر دارد، در حالی که  $\text{\thinlines}$  و  $\text{\thicklines}$  بر خط‌های اریب و دایره‌ها و بیضی‌ها نیز تأثیر دارند.

## ۸.۲.۵ استفاده چندباره از جعبه‌های تصویر پیش‌ساخته

```

\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
(40,32)[bl]{% definition
\multiput(0,0)(0,28){2}
{\line(1,0){40}}
\multiput(0,0)(40,0){2}
{\line(0,1){28}}
\put(1,28){\oval(2,2)[tl]}
\put(1,29){\line(1,0){5}}
\put(9,29){\oval(6,6)[tl]}
\put(9,32){\line(1,0){8}}
\put(17,29){\oval(6,6)[tr]}
\put(20,29){\line(1,0){19}}
\put(39,28){\oval(2,2)[tr]}
}
\newsavebox{\folderb}
\savebox{\folderb}
(40,32)[l]{% definition
\put(0,14){\line(1,0){8}}
\put(8,0){\usebox{\foldera}}
}
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
{\usebox{\folderb}}
\end{picture}

```



یک جعبه تصویر را می‌توان با فرمان

```
\newsavebox{name}
```

معرفی، و با فرمان

```
\savebox{name}(width,height)[position]{content}
```

تعریف، و نهایتاً با فرمان

```
\put(x,y)\usebox{name}
```

رسم کرد.

پارامتر اختیاری *position* نقش لنگر را برای جعبه بازی می‌کند. در مثال این پارامتر برابر `bl` تعریف شده است که لنگر را در گوشهٔ چپ پایین صفحه قرار می‌دهد. گزینه‌های دیگر `t` (بالا) و `r` (راست) هستند.

آرگومان *name* به یک جعبه در لاتک ارجاع می‌کند و بنابراین طبیعت فرمان دارد. تصاویر درون جعبه‌ها می‌توانند تودرتو باشند: در این مثال `\foldera` درون `\folderb` تعریف شده است.

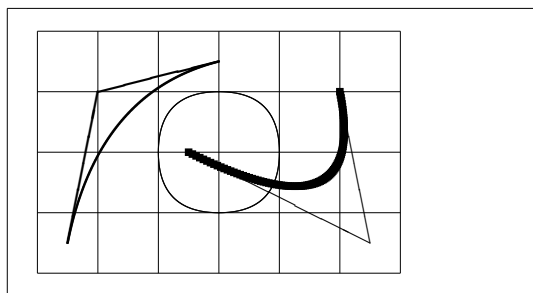
فرمان `\oval` که باید همانند `\line` استفاده شود به پاره‌خط‌های کمتر از ۳ میلیمتر بی‌تأثیر است.

## ۹.۲.۵ خم‌های درجهٔ دوم بزیه

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
 \linethickness{0.075mm}
 \multiput(0,0)(1,0){7}
 {\line(0,1){4}}
 \multiput(0,0)(0,1){5}
 {\line(1,0){6}}
 \thicklines
 \put(0.5,0.5){\line(1,5){0.5}}
 \put(1,3){\line(4,1){2}}
 \qbezier(0.5,0.5)(1,3)(3,3.5)
 \thinlines
 \put(2.5,2){\line(2,-1){3}}
 \put(5.5,0.5){\line(-1,5){0.5}}
 \linethickness{1mm}
 \qbezier(2.5,2)(5.5,0.5)(5,3)
 \thinlines
 \qbezier(4,2)(4,3)(3,3)
 \qbezier(3,3)(2,3)(2,2)
 \qbezier(2,2)(2,1)(3,1)
 \qbezier(3,1)(4,1)(4,2)
\end{picture}

```



همان‌طور که این مثال نشان می‌دهد، تقسیم یک دایره به چهار خم بزیه مطلوب نیست. حداقل ۸ قسمت مورد نیاز است. شکل، دوباره اثر فرمان `\linethickness` را روی خط‌های افقی و عمودی، و اثر `\thicklines` و `\thinlines` را روی خط‌های مورب نشان می‌دهد. این مثال همچنین نشان می‌دهد که همهٔ این فرمان‌ها روی خم‌های بزیه مؤثر هستند و اثر فرمان‌های قبلی را از بین می‌برند. فرض کنید  $P_1 = (x_1, y_1)$ ،  $P_2 = (x_2, y_2)$  نقاط انتهایی باشند، و  $m_1$ ،  $m_2$  به ترتیب شیب‌های خم‌های بزیه باشند. نقطهٔ کنترل کنندهٔ میانی  $S = (x, y)$  با رابطهٔ

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (1.5)$$

داده شده است. [۱۷] را برای دیدن یک برنامهٔ جاوا که خط فرمان لازم برای فرمان‌های `\qbezier` را ارائه می‌دهد ببینید.

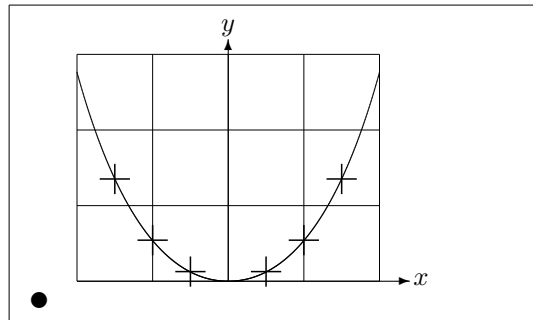


۱۰.۲.۵ تسبیح

```

\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){\mathit{x}}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){\mathit{y}}}
\qbezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{.2mm}
\put(.3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}

```



در این شکل، هر نیمه متقارن از تسبیح  $y = \cosh x - 1$  با یک خم بزیه تقریب زده شده است. نیمه سمت راست در نقطه  $(2, 2.7622)$  به پایان می‌رسد، که شیب خط در این نقطه  $m = 3.6269$  است. با استفاده دوباره از رابطه  $(1.5)$ ، می‌توانیم نقاط میانی کنترلی را بدست آوریم. این نقاط برابرند با  $(1.2384, 0)$  و  $(-1.2384, 0)$ . علامت‌های صلیب نقاط تسبیح را نشان می‌دهند. خطا قابل چشم‌پوشی است و کمتر از یک درصد است.

این مثال استفاده از آرگومان اختیاری فرمان `\begin{picture}` را نشان می‌دهد. تصویر به

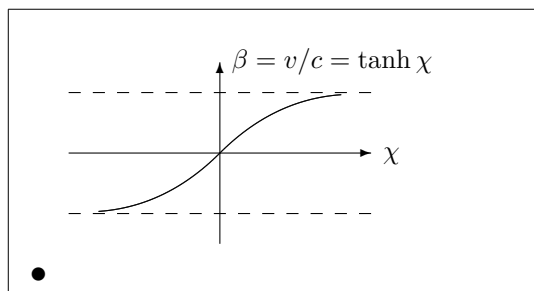
صورت مناسب مولفه‌های ریاضی تعریف شده است، با این وجود با فرمان

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

گوشه سمت چپ پایین (که با قرص سیاه مشخص شده است) با مختصات  $(-۲/۵, -۰/۲۵)$  تعریف شده است.

۱۱.۲.۵ سرعت در نظریه نسبیت عام

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)(-3,-2)
 \put(-2.5,0){\vector(1,0){5}}
 \put(2.7,-0.1){χ}
 \put(0,-1.5){\vector(0,1){3}}
 \multiput(-2.5,1)(0.4,0){13}
 {\line(1,0){0.2}}
 \multiput(-2.5,-1)(0.4,0){13}
 {\line(1,0){0.2}}
 \put(0.2,1.4)
 {$\beta=v/c=\tanh\chi$}
 \qbezier(0,0)(0.8853,0.8853)
 (2,0.9640)
 \qbezier(0,0)(-0.8853,-0.8853)
 (-2,-0.9640)
 \put(-3,-2){\circle*{0.2}}
\end{picture}
```



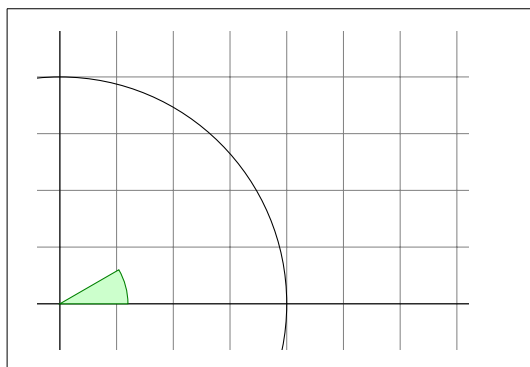
نقاط کنترلی خم‌های بزیه با فرمول‌های (۱.۵) محاسبه شده‌اند. شاخه مثبت با  $P_1 = (0, 0)$ ، شاخه منفی با  $P_2 = (2, \tanh 2)$ ،  $m_1 = 1$  و  $m_2 = 1/\cosh^2 2$  تعریف می‌شود. دوباره، تصویر به شکل مختصات مناسب مؤلفه‌ای ریاضی تعریف شده است و گوشه سمت چپ پایین با مختصات  $(-۳, -۲)$  تعریف شده است (دیسک سیاه).

### ۳.۵ بسته گرافیک TikZ & PGF

امروزه هر سیستم تولید خروجی  $\text{\LaTeX}$  توانایی تولید تصاویر برداری زیبا را دارد، تنها ابزار انجام این کار ممکن است تغییر کند. بسته PGF یک لایه رویی برای انجام این کار را در اختیار شما قرار می‌دهد و اجازه می‌دهد که این کار را با استفاده از فرمان‌های ساده به راحتی انجام دهید و تصاویر برداری پیچیده را دقیقاً از داخل نوشتار تولید کنید. بسته PGF دارای راهنمای +۵۰۰ صفحه‌ای است [۱۸]. بنابراین در این بخش کوتاه قصد داریم تنها جرحه‌ای از این چشمه بی‌کران را به شما بچشانیم.

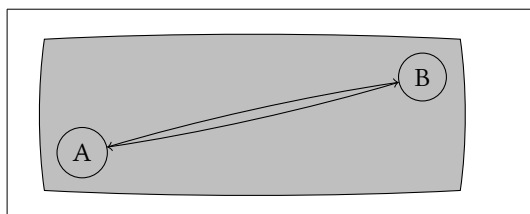
برای دسترسی سطح بالا به توابع PGF باید بسته tikz را فراخوانی کنید. با استفاده از بسته tikz می‌توانید فرمان‌های بسیار مؤثری را برای رسم تصاویر از داخل نوشتار خود استفاده کنید. از محیط tikzpicture برای این کار استفاده کنید.

```
\begin{tikzpicture}[scale=3]
 \clip (-0.1,-0.2)
 rectangle (1.8,1.2);
 \draw[step=.25cm,gray,very thin]
 (-1.4,-1.4) grid (3.4,3.4);
 \draw (-1.5,0) -- (2.5,0);
 \draw (0,-1.5) -- (0,1.5);
 \draw (0,0) circle (1cm);
 \filldraw[fill=green!20!white,
 draw=green!50!black]
 (0,0) -- (3mm,0mm)
 arc (0:30:3mm) -- cycle;
\end{tikzpicture}
```

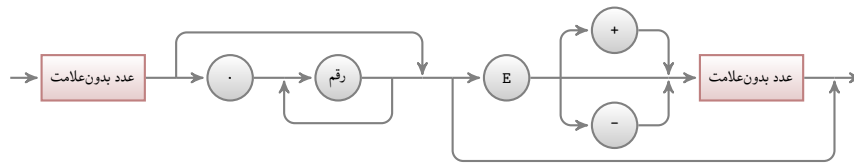


اگر به زبان‌های دیگر برنامه‌نویسی آشنا هستید، ممکن است به فرمان آشنای نیم‌نقطه (;) توجه کرده باشید که برای جداسازی فرمان‌های مختلف مورد استفاده قرار می‌گیرد. با استفاده از فرمان `\usetikzlibrary` در سرآغاز نوشتار خود می‌توانید امکانات بیشتری را برای رسم اشکال ویژه فعال کنید، مانند جعبه‌هایی که خم شده‌اند.

```
\usetikzlibrary{%
 decorations.pathmorphing}
\begin{tikzpicture}[
 decoration={bent,aspect=.3}]
 \draw [decorate,fill=lightgray]
 (0,0) rectangle (5.5,2);
 \node[circle,draw]
 (A) at (.5,.5) {A};
 \node[circle,draw]
 (B) at (5,1.5) {B};
 \draw[->,decorate] (A) -- (B);
 \draw[->,decorate] (B) -- (A);
\end{tikzpicture}
```



همچنین می‌توانید دیاگرام‌هایی را رسم کنید که مانند این است که دقیقاً از یک کتاب برنامه نویسی پاسکال برداشته شده است. کد این کار کمی پیچیده‌تر از مثال بالا است، بنابراین تنها اثر آن را نمایش می‌دهم. اگر به راهنمای بسته PGF نگاهی بیندازید، می‌توانید راهنمای مفصل رسم این دیاگرام‌ها را ببینید.



چیزهای بیشتری وجود دارد؛ اگر می‌خواهید نمودار داده‌های عددی را رسم کنید، باید نگاه دقیق‌تری به راهنمای بستهٔ pgfplot بیندازید. این راهنما شامل هر چیزی است که برای رسم این نمودارها لازم دارید. حتی می‌توانید فرمان gnuplot را استفاده کنید تا مقدار دقیق توابع مورد نظر خود را بدست آورید.

## Xy-pic ۴.۵

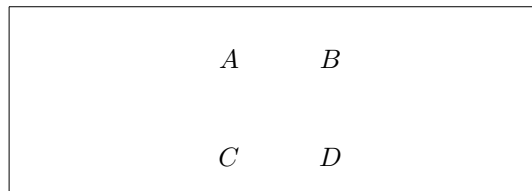
xy یک بسته برای طراحی دیاگرام‌هاست. برای استفاده از آن، فرمان زیر را در سرآغاز نوشتار خود قرار دهید:

```
\usepackage[options]{xy}
```

options لیستی از توابع Xy-pic است که می‌خواهید فراخوانی کنید. این گزینه‌ها برای غلط‌گیری بسیار مؤثر هستند. توصیه می‌کنم تمام گزینه‌ها را با گزینهٔ all فعال کنید تا لاتک تمام فرمان‌های Xy را فراخوانی کند.

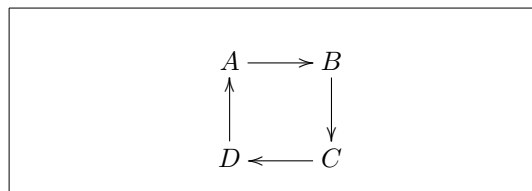
دیاگرام‌های Xy-pic روی یک طرح ماتریسی نمایش داده می‌شوند، که هر دیاگرام در یک خانهٔ ماتریس قرار می‌گیرد:

```
\begin{displaymath}
\xymatrix{A & B \\
 C & D }
\end{displaymath}
```



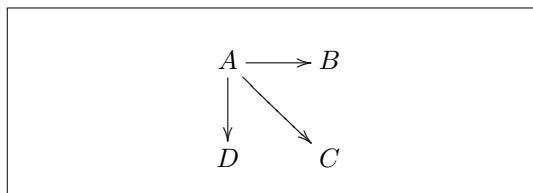
فرمان `\xymatrix` باید در محیط ریاضی مورد استفاده قرار بگیرد. در اینجا دو سطر و دو ستون مشخص کرده‌ایم. برای این که این ماتریس را به یک دیاگرام تبدیل کنیم باید جهت پیکان‌ها را با فرمان `\ar` مشخص کنیم.

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
 D \ar[u] & C \ar[l] }
\end{displaymath}
```



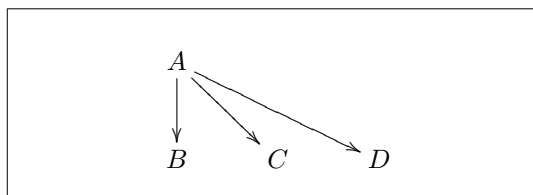
فرمان پیکان در سلول اصلی پیکان قرار داده می‌شود. آرگومان‌ها جهت پیکان هستند و باید به `right`، `down`، `left` یا `up` اشاره کنند.

```
\begin{displaymath}
\xymatrix{
A \ar[d] \ar[dr] \ar[r] & B \\
D & C }
\end{displaymath}
```



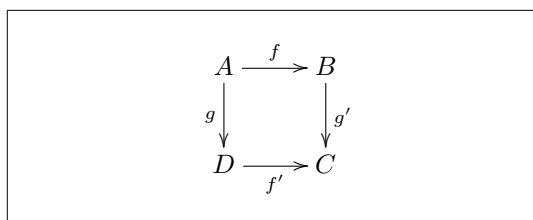
برای رسم قطرها، فقط کافی است جهت را معرفی کنیم. در حقیقت، می‌توانید جهت را تکرار کنید تا پیکان‌ها بزرگتر شوند.

```
\begin{displaymath}
\xymatrix{
A \ar[d] \ar[dr] \ar[dr] & & \\
B & & C & D }
\end{displaymath}
```



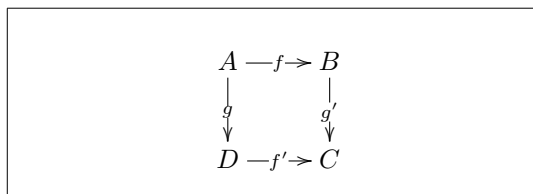
می‌توانیم حتی دیاگرام‌های جالب با افزودن برجسب به پیکان‌ها طراحی کنیم. برای این کار، از فرمان‌های زیرنویس و بالانویس استفاده می‌کنیم.

```
\begin{displaymath}
\xymatrix{
A \ar[r]^{f} \ar[d]_g & B \\
D \ar[r]_{f'} & C }
\end{displaymath}
```



همان‌طور که نشان داده شد، این کارها را همانند سبک ریاضی می‌توان انجام داد. تنها تفاوت در این است که بالانویس به معنای بالای پیکان و پایین‌نویس پایین پیکان است. عملگر سومی نیز وجود دارد: | این فرمان باعث می‌شود متنی در درون یک پیکان ظاهر شود.

```
\begin{displaymath}
\xymatrix{
A \ar[r]|f \ar[d]|g & B \\
D \ar[r]|{f'} & C }
\end{displaymath}
```

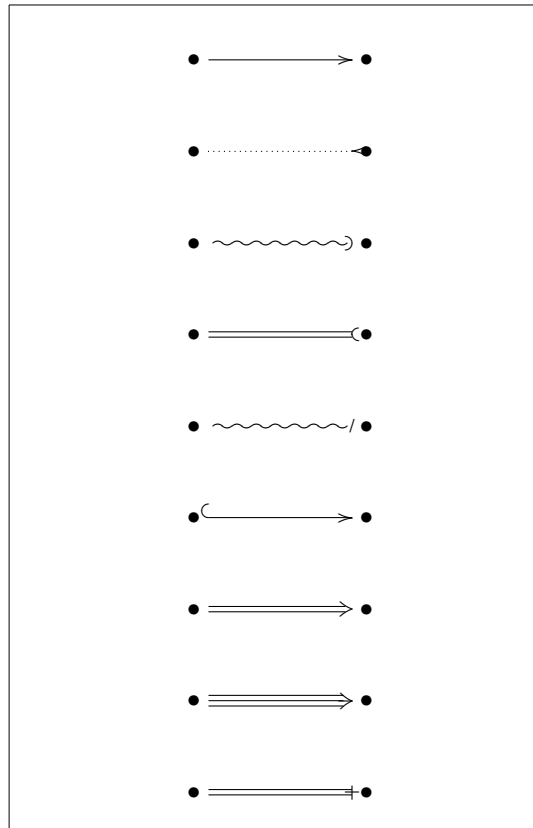


برای رسم یک پیکان با یک حفره درون آن از `\ar[...]|hole` استفاده کنید. در بعضی حالات، مهم است که تفاوت بین انواع پیکانها را بدانیم. این کار را می‌توان با قرار دادن برجسبی بر آنها یا تغییر ظاهر آنها انجام داد.

```

\begin{displaymath}
\mathrm{xymatrix}
{
\bullet \ar@{->}[rr] && \bullet \\
\bullet \ar@{.<}[rr] && \bullet \\
\bullet \ar@{-~}[rr] && \bullet \\
\bullet \ar@{=()}[rr] && \bullet \\
\bullet \ar@{-/}[rr] && \bullet \\
\bullet \ar@{^{}(->}[rr] && \bullet \\
\bullet \ar@{2{->}[rr] && \bullet \\
\bullet \ar@{3{->}[rr] && \bullet \\
\bullet \ar@{=+}[rr] && \bullet
}
\end{displaymath}

```

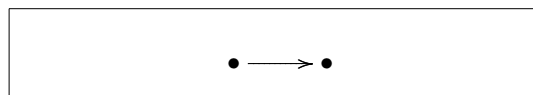


به تفاوت بین دو دیاگرام توجه کنید:

```

\begin{displaymath}
\mathrm{xymatrix}
{
\bullet \ar[r]
\ar@{.>}[r] &
\bullet
}
\end{displaymath}

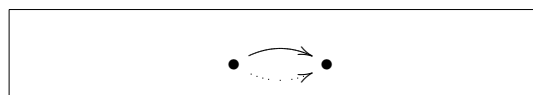
```



```

\begin{displaymath}
\mathrm{xymatrix}
{
\bullet \ar@/^/[r]
\ar@/_/@{.>}[r] &
\bullet
}
\end{displaymath}

```



تنظیم‌کننده‌های بین دو اسلش روش رسم خم‌ها را مشخص می‌کنند.  $X\text{-pic}$  روش‌های بسیاری را برای تغییر سبک رسم خم‌ها ارائه می‌کند: برای اطلاع بیشتر به راهنمای  $X\text{-pic}$  مراجعه کنید.



## فصل ۶

# تنظیم شخصی لاتک

فرمان‌هایی را که تا به حال آموخته‌اید مناسب نوشتاری برای بسیاری از افراد است. با این که ممکن است ظاهر خیلی شیک نداشته باشند ولی از اصول حروف‌چینی استاندارد پیروی می‌کنند که باعث سهولت خواندن آنها می‌شود. با این وجود شرایطی وجود دارد که لاتک فرمانی مناسب نیاز شما ندارد یا این که خروجی حاصل از فرمان‌های موجود مطلوب شما نیست. در این فصل، سعی می‌کنم روش راهنمایی لاتک برای تولید خروجی‌هایی را توضیح دهم که با روش پیش‌فرض آن متفاوت است.

### ۱.۶ فرمان‌ها، محیط‌ها، و بسته‌های جدید

شاید تا به حال توجه کرده‌باشید که تمام فرمان‌هایی را که در این مقدمه توضیح داده‌ام در یک جعبه قرار دارند و این فرمان‌ها در نمایه آخر کتاب قرار دارند. به جای این که از فرمان‌های استاندارد لاتک برای دستیابی این منظور استفاده کنم، بسته‌ای را تعریف کرده‌ام که در آن تعاریف و فرمان‌ها و محیط‌هایی را گنجانده‌ام. حالا به راحتی می‌توانم بنویسم:

```
\begin{lsccommand}
\ci{dum}
\end{lsccommand}
```

\dum

در این مثال، از یک محیط جدید `lsccommand`، که مسئولیت رسم یک کادر پیرامون فرمان را دارد، و یک فرمان `\ci`، که مسئولیت درج فرمان و قرار دادن مؤلفه متناظر را در نمایه دارد، استفاده کرده‌ام. می‌توانید این موضوع را با نگاه کردن به فرمان `\dum` در نمایه آخر کتاب ببینید، که در آنجا خواهید دید که شماره تمام صفحاتی را که در آن فرمان `\dum` آمده است مشخص شده است.

هرگاه بخواهم که دیگر فرمان‌ها در کادر نمایش داده نشوند به سادگی تنها باید تعریف محیط `\lscmd` را تغییر دهم. این کار به وضوح بسیار ساده‌تر از این است که تمام متن را برای تغییر فرمان‌ها بررسی کنم.

### ۱.۱.۶ فرمان‌های جدید

برای افزودن فرمان مناسب کار خودتان به شکل زیر عمل کنید

```
\newcommand{name}[num]{definition}
```

به طور پایه‌ای، فرمان نیاز به دو آرگومان دارد: نام فرمان (*name*) و تعریف فرمان (*definition*). آرگومان *num* که در براکت قرار می‌گیرد اختیاری است و تعداد آرگومان‌هایی را که فرمان می‌پذیرد مشخص می‌کند (حداکثر ۹ تا). حالت پیش فرض آن صفر است که هیچ آرگومانی را نمی‌پذیرد. دو مثال زیر کمک می‌کنند که این موضوع را بهتر درک کنید. مثال اول فرمان جدیدی به نام `\tnss` را مشخص می‌کند که اثر آن درج “The Not So Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$ ” است. چنین فرمانی موقعی مفید است که عنوان کتاب در نوشتار مکرراً تکرار می‌شود.

```
\newcommand{\tnss}{The not
so Short Introduction to
\LaTeXe}
This is ``\tnss'' \ldots{ }
``\tnss''
```

This is “The not so Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$ ” ... “The not so Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$ ”

مثال دوم فرمان دیگری را تعریف می‌کند که تنها یک آرگومان می‌پذیرد. مقدار #1 جایگزین آرگومان مشخص شده می‌شود. اگر می‌خواهید بیش از یک آرگومان داشته باشید از #2 و غیره استفاده کنید.

```
\newcommand{\txsit}[1]
{This is the \emph{#1} Short
Introduction to \LaTeXe}
% in the document body:
\begin{itemize}
\item \txsit{not so}
\item \txsit{very}
\end{itemize}
```

- This is the *not so* Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$
- This is the *very* Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$

لاتک به شما اجازه ساختن فرمانی را نمی‌دهد که قبلاً تعریف شده است. اما فرمان ویژه‌ای وجود دارد که با استفاده از آن می‌توانید یک فرمان از پیش تعریف شده را دوباره تعریف کنید: `\renewcommand`. این فرمان دقیقاً همان فرم فرمان `\newcommand` را دارد. در بعضی مواقع ممکن است بخواهید از فرمان `\providecommand` استفاده کنید. سبک این فرمان همانند فرمان `\newcommand` است، اما اگر فرمان مربوطه قبلاً تعریف شده باشد لاتک این فرمان را در نظر نمی‌گیرد.

چند نکته در مورد فاصله خالی بعد از یک فرمان لاتک باید در نظر داشته باشید. صفحه ۵ را برای اطلاعات بیشتر ببینید.

### ۲.۱.۶ محیط‌های جدید

مشابه فرمان `\newcommand`، فرمانی برای ساختن محیط‌ها وجود دارد `\newenvironment`. این فرمان فرم زیر را می‌پذیرد:

```
\newenvironment{name} [num] {before}{after}
```

دوباره `\newenvironment` می‌تواند یک آرگومان اختیاری داشته باشد. محتویات `before` قبل از متن محیط پردازش می‌شود. محتویات `after` بعد از فرمان `\end{name}` اجرا می‌شوند. در مثال زیر نحوه استفاده از فرمان `\newenvironment` شرح داده شده است.

```
\newenvironment{king}
{\rule{1ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{1ex}{1ex}}
```

■ My humble subjects ... ■

```
\begin{king}
My humble subjects \ldots
\end{king}
```

آرگومان `num` همانند آرگومان همنام فرمان `\newcommand` مورد استفاده قرار می‌گیرد. لاتک بررسی می‌کند که یک محیط از پیش تعریف شده را دوباره تعریف نکنید. اگر می‌خواهید یک محیط قبلی را از نو تعریف کنید از فرمان `\renewenvironment` استفاده کنید. روش استفاده از آن همانند `\newenvironment` است.

فرمان‌های استفاده شده در این مثال بعداً شرح داده خواهند شد. برای فرمان `\rule` صفحه ۱۲۳، برای `\stretch` صفحه ۱۱۷، و برای `\hspace` صفحه ۱۱۶ را ببینید.

### ۳.۱.۶ فاصله‌های اضافه

هنگام تعریف محیط‌های جدید ممکن است با فاصله‌های زیاد قبل و بعد از آن مشکل داشته باشید؛ به عنوان مثال وقتی که می‌خواهید یک محیط عنوان تعریف کنید که تورفتگی آن به اندازه پاراگراف بعدی باشد. فرمان `\ignorespaces` بلوک ابتدایی محیط را وادار می‌کند تا فاصله بعد از اجرای بلوک ابتدایی را نادیده بگیرد. بلوک انتهایی کمی پیچیده‌تر است زیرا این بلوک شامل پردازش‌های ویژه‌ای است. با فرمان `\ignorespacesafterend`، لاتک یک فرمان `\ignorespaces` را بعد از پایان پردازش اجرا می‌کند.

```
\newenvironment{simple}%
{\noindent}%
{\par\noindent}

\begin{simple}
See the space\\to the left.
\end{simple}
Same\\here.
```

See the space  
to the left.

Same  
here.

```
\newenvironment{correct}%
{\noindent\ignorespaces}%
{\par\noindent%
\ignorespacesafterend}

\begin{correct}
No space\\to the left.
\end{correct}
Same\\here.
```

No space  
to the left.

Same  
here.

#### ۴.۱.۶ خط فرمان لاتک

اگر روی سیستمی مانند لینوکس کار می‌کنید، ممکن است از Makefileها برای ساختن پروژه لاتک خود استفاده کنید. در این راستا جالب است که نسخه متفاوتی از نوشتار خود را با اجرای لاتک در خط فرمان درست کنید. اگر ساختار زیر را به نوشتار خود اضافه کنید:

```
\usepackage{ifthen}
\ifthenelse{\equal{\blackandwhite}{true}}{
% "black and white" mode; do something..
}{
% "color" mode; do something different..
}
```

حال می‌توانید لاتک را به شکل زیر فراخوانی کنید:

```
latex '\newcommand{\blackandwhite}{true}\input{test.tex}'
```

ابتدا فرمان `\blackandwhite` تعریف می‌شود و آنگاه فایل اصلی خوانده می‌شود. با قرار دادن `\blackandwhite` برابر `false` نسخه رنگی نوشتار تولید خواهد شد.

## ۵.۱.۶ بسته‌های شخصی

اگر فرمان‌ها و محیط‌های زیادی را تعریف کنید، سرآغاز فایل شما بسیار طولانی خواهد شد. در این حالت مناسب‌تر است که یک بسته لاتک شامل فرمان‌ها و محیط‌های شخصی خود را بسازید. آنگاه می‌توانید از فرمان `\usepackage` برای فراخوانی بسته خود در نوشتار استفاده کنید.

---

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{The not so Short Introduction
 to \LaTeXe}
\newcommand{\txsit}[1]{The \emph{#1} Short
 Introduction to \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

---

شکل ۱.۶: مثال بسته

نوشتن یک بسته شامل قرار دادن محتویات سرآغاز فایل در یک فایل با پسوند `.sty` است. یک فرمان ویژه وجود دارد

```
\ProvidesPackage{package name}
```

که در ابتدای بسته قرار می‌گیرد. فرمان `\ProvidesPackage` به لاتک نام بسته را می‌گوید و لاتک را قادر می‌سازد که پیغام خطایی را هنگام نوشتن یک بسته از پیش تعریف شده بدهد. شکل ۱.۶ یک مثال کوچک از یک بسته را نشان می‌دهد که شامل فرمان‌های تعریف شده در مثال‌های بالا است.

## ۲.۶ قلم‌ها و اندازه آنها

## ۱.۲.۶ فرمان تغییر قلم

لاتک قلم و اندازه مناسب را بسته به ساختار منطقی نوشتار انتخاب می‌کند (بخش، پانویس، ...). گاهی اوقات نیاز است که قلم و اندازه آن را به صورت دستی تغییر دهیم. برای این کار از فرمان‌های ارائه شده در جدول‌های ۱.۶ و ۲.۶ استفاده کنید. اندازه واقعی هر قلم به طبقه نوشتار و گزینه‌های آن بستگی دارد. جدول ۳.۶ مقدار دقیق را برای هر کدام از طبقه‌های استاندارد نشان می‌دهد.

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

```
The small and bold Romans ruled all of
great big Italy.
```

یک امکان مهم لاتک این است که شکل قلم‌ها مستقل هستند. یعنی این که می‌توانید اندازه قلم را تغییر دهید و همزمان شکل سیاه و خوابیده را داشته باشید. در سبک ریاضی می‌توانید فرمان‌های تغییر قلم را با خروج اضطراری از سبک ریاضی به صورت متن عادی بنویسید. اگر می‌خواهید از قلم دیگری برای نوشتن فرمول‌ها استفاده کنید باید از فرمان‌های دیگری استفاده کنید؛ به جدول ۴.۶ مراجعه کنید. در مورد فرمان‌های اندازه قلم، آکولاد نقش مهمی دارد. از آنها برای ساختن یک گروه استفاده می‌شود. یک گروه تاثیر بیشتر فرمان‌های لاتک را محدود می‌کند.

He likes {\LARGE large and  
{\small small} letters}.

He likes large and small letters.

فرمان‌های اندازه قلم روی فاصله خالی نیز تاثیر دارند اما تنها در موقعی که پایان پاراگراف قبل از پایان تاثیر فرمان تغییر قلم باشد. بنابراین توجه داشته باشید که } مربوط به پایان فرمان تغییر قلم زودتر

جدول ۱.۶: قلم‌ها

|                           |                   |                               |                  |
|---------------------------|-------------------|-------------------------------|------------------|
| <code>\textrm{...}</code> | roman             | <code>\textsf{...}</code>     | sans serif       |
| <code>\texttt{...}</code> | typewriter        |                               |                  |
| <code>\textmd{...}</code> | medium            | <code>\textbf{...}</code>     | <b>bold face</b> |
| <code>\textup{...}</code> | upright           | <code>\textit{...}</code>     | <i>italic</i>    |
| <code>\textsl{...}</code> | <i>slanted</i>    | <code>\textsc{...}</code>     | SMALL CAPS       |
| <code>\emph{...}</code>   | <i>emphasized</i> | <code>\textnormal{...}</code> | document font    |

جدول ۲.۶: اندازه قلم

|                            |                  |                     |                 |
|----------------------------|------------------|---------------------|-----------------|
| <code>\tiny</code>         | tiny font        | <code>\Large</code> | larger font     |
| <code>\scriptsize</code>   | very small font  | <code>\LARGE</code> | very large font |
| <code>\footnotesize</code> | quite small font | <code>\huge</code>  | huge            |
| <code>\small</code>        | small font       | <code>\Huge</code>  | largest         |
| <code>\normalsize</code>   | normal font      |                     |                 |
| <code>\large</code>        | large font       |                     |                 |

جدول ۳.۶: اندازه واقعی قلم در طبقه استاندارد

| size                       | 10pt (default) | 11pt option | 12pt option |
|----------------------------|----------------|-------------|-------------|
| <code>\tiny</code>         | 5pt            | 6pt         | 6pt         |
| <code>\scriptsize</code>   | 7pt            | 8pt         | 8pt         |
| <code>\footnotesize</code> | 8pt            | 9pt         | 10pt        |
| <code>\small</code>        | 9pt            | 10pt        | 11pt        |
| <code>\normalsize</code>   | 10pt           | 11pt        | 12pt        |
| <code>\large</code>        | 12pt           | 12pt        | 14pt        |
| <code>\Large</code>        | 14pt           | 14pt        | 17pt        |
| <code>\LARGE</code>        | 17pt           | 17pt        | 20pt        |
| <code>\huge</code>         | 20pt           | 20pt        | 25pt        |
| <code>\Huge</code>         | 25pt           | 25pt        | 25pt        |

جدول ۴.۶: قلم‌های ریاضی

|                               |                          |
|-------------------------------|--------------------------|
| <code>\mathrm{...}</code>     | Roman Font               |
| <code>\mathbf{...}</code>     | <b>Boldface Font</b>     |
| <code>\mathsf{...}</code>     | Sans Serif Font          |
| <code>\mathtt{...}</code>     | Typewriter Font          |
| <code>\mathit{...}</code>     | <i>Italic Font</i>       |
| <code>\mathcal{...}</code>    | <i>CALLIGRAPHIC FONT</i> |
| <code>\mathnormal{...}</code> | <i>Normal Font</i>       |

از پایان پاراگراف ظاهر نشود. به مکان فرمان `\par` در دو مثال زیر توجه کنید.<sup>۱</sup>

```
{\Large Don't read this!
 It is not true.
 You can believe me!\par}
```

Don't read this! It is not true.  
You can believe me!

```
{\Large This is not true either.
 But remember I am a liar.}\par
```

This is not true either. But remember I am a liar.

اگر می‌خواهید یک فرمان تغییر اندازه قلم را برای کل یک پاراگراف یا کل یک نوشتار فعال کنید، می‌توانید از محیط مناسب آن استفاده کنید.

```
\begin{Large}
 This is not true.
 But then again, what is these
 days \ldots
\end{Large}
```

This is not true. But then again,  
what is these days ...

این کار شما را از نوشتن تعداد زیادی آکولاد بی‌نیاز می‌کند.

## ۲.۲.۶ خطر، ویل رابینسون، خطر

همان‌طور که در ابتدای این فصل گفته شد، شلوغ کردن فایل خود با فرمان‌هایی از این دست خطرناک است زیرا با روح لاتک در تناقض است که می‌گوید ساختار منطقی را از تغییرات بصری جدا کنید. یعنی اگر می‌خواهید از یک فرمان تغییر اندازه قلم چندین بار در نوشتار خود استفاده کنید باید از `\newcommand` برای تعریف یک فرمان منطقی تغییر قلم استفاده کنید.

```
\newcommand{\oops}[1]{%
 \textbf{#1}}
Do not \oops{enter} this room,
it's occupied by \oops{machines}
of unknown origin and purpose.
```

Do not **enter** this room, it's occupied by **machines** of unknown origin and purpose.

این رهیافت دارای این دستاورد است که در مراحل بعد برای تغییر این نمایش بصری کافی است که تعریف آن را تغییر دهید تا این که در کل فایل خود بدنبال متن `\textbf` بگردید و برای هر کدام از آنها تصمیم بگیرید که باید تغییر کند یا نه.

<sup>۱</sup> `\par` معادل با یک خط خالی است.



## ۳.۲.۶ توصیه

به عنوان پایان سفر به دنیای قلم‌ها و اندازه آنها، توصیه‌ای را بیان می‌کنیم:

**Remember!** The **MO RE** fonts **you** use in a document, the more readable and *beautiful* it becomes.

به یاد داشته باشید! هر چقدر از قلم‌های بیشتری در نوشتار استفاده کنید نوشتار شما زیباتر و خواناتر خواهد شد.

## ۳.۶ فاصله‌گذاری

## ۱.۳.۶ فاصله خط‌ها

اگر می‌خواهید فاصله بین خط‌ها بیشتر از حالت معمولی باشد می‌توانید این کار را با قرار دادن فرمان زیر در سرآغاز فایل انجام دهید

```
\linespread{factor}
```

از `\linespread{1.3}` برای فاصله یک‌ونیم برابر و از `\linespread{1.6}` برای فاصله دو برابر استفاده کنید. فاصله نرمال یک برابر است. توجه داشته باشید که اثر فرمان `\linespread` شدید است و مناسب چاپ نیست. بنابراین اگر دلیل قانع‌کننده دارید می‌توانید از این فرمان استفاده کنید:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
{1.5\baselineskip}}
```

This paragraph is typeset with the baseline skip set to 1.5 of what it was before. Note the par command at the end of the paragraph. `\par`

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

This paragraph is typeset with the baseline skip set to 1.5 of what it was before. Note the par command at the end of the paragraph.

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

## ۲.۳.۶ شکل پاراگراف

در لاتک دو پارامتر وجود دارند که شکل پاراگراف را تغییر می‌دهند. با قرار دادن تعریفی شبیه به

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

در سرآغاز فایل ورودی می‌توانید شکل پاراگراف‌ها را تغییر دهید. این دو فرمان فاصله بین دو پاراگراف را بیشتر می‌کنند و تورفتگی پاراگراف را صفر می‌کنند. قسمت plus و minus از طول به لاتک می‌گوید فاصله بین پاراگراف‌ها را می‌تواند برای قرار گرفتن درست در صفحه کم یا زیاد کند.

در قاره اروپا، پاراگراف‌ها با فاصله از هم نوشته می‌شوند ولی تورفتگی ندارند. اما توجه داشته باشید که این فرمان بر فهرست مطالب نیز تاثیر دارد. فاصله بین خط‌های فهرست مطالب نیز تغییر می‌کند. برای اجتناب از این کار، می‌توانید این دو فرمان را از سرآغاز حذف کنید و به بعد از `\tableofcontents` انتقال دهید، یا این که اصلاً از آنها استفاده نکنید زیرا کتاب‌های حرفه‌ای از تورفتگی به جای فاصله برای مشخص کردن پاراگراف‌ها استفاده می‌کنند.

اگر می‌خواهید پاراگرافی را که تورفتگی ندارد دارای تورفتگی کنید از فرمان

```
\indent
```

در ابتدای پاراگراف استفاده کنید.<sup>۲</sup> به وضوح این کار موقعی موثر است که `\parindent` برابر صفر تعریف نشده باشد.

برای نوشتن یک پاراگراف بدون تورفتگی از فرمان

```
\noindent
```

در ابتدای پاراگراف استفاده کنید. این کار موقعی که می‌خواهید یک متن را بدون داشتن بخش بنویسید مفید است.

## ۳.۳.۶ فاصله افقی

لاتک فاصله بین کلمه‌ها و جمله‌ها را به طور خودکار تنظیم می‌کند. برای افزایش فاصله افقی از فرمان

```
\hspace{length}
```

استفاده کنید. اگر می‌خواهید این فاصله حتی در ابتدا و انتهای خط باقی بماند از `\hspace*` به جای `\hspace` استفاده کنید. مقدار `length` در ساده‌ترین حالت تنها یک عدد به اضافه یک کمیت است. مهمترین کمیت‌ها در جدول ۵.۶ ارائه شده‌اند.

<sup>۲</sup> برای تورفته کردن اولین پاراگراف هر بخش از بسته `indentfirst` که جزئی از کلاف `tools` است استفاده کنید.

جدول ۵.۶: کمیت‌های تک

---

|    |                                             |                                       |   |
|----|---------------------------------------------|---------------------------------------|---|
| mm | millimetre                                  | $\approx 1/25$ inch                   | □ |
| cm | centimetre                                  | = 10 mm                               | □ |
| in | inch                                        | = 25.4 mm                             | □ |
| pt | point                                       | $\approx 1/72$ inch $\approx 1/36$ mm | □ |
| em | approx width of an 'M' in the current font  |                                       | □ |
| ex | approx height of an 'x' in the current font |                                       | □ |

---

This\hspace{1.5cm}is a space  
of 1.5 cm.

This is a space of 1.5 cm.

فرمان

`\stretch{n}`

یک فاصله کشیده تولید می‌کند. این فاصله کل فاصله باقیمانده خط را پر می‌کند. اگر چند فرمان `\hspace{\stretch{n}}` در یک خط قرار بگیرند، هرکدام مقداری متناسب با فاکتور کشیدگی خود اشغال می‌کند.

`x\hspace{\stretch{1}}`  
`x\hspace{\stretch{3}}x`

x x x

وقتی که فاصله افقی را به همراه متن به کار می‌برید، مناسب است که فاصله را متناسب با اندازه قلم تعیین کنید. این کار را می‌توان با کمیت وابسته به قلم `em` و `ex` تعیین کرد:

`{\Large}big\hspace{1em}y\`  
`{\tiny}tin\hspace{1em}y`

big y  
tin y

### ۴.۳.۶ فاصله عمودی

فاصله بین پاراگراف‌ها، بخش‌ها، زیربخش‌ها، ... به صورت خودکار توسط لاتک تعیین می‌شود. هر وقت که لازم است، فاصله عمودی بین دو پاراگراف را می‌توان با فرمان زیر تولید کرد:

`\vspace{length}`

این فرمان به طور نرمال با یک خط فاصله خالی بین دو پاراگراف قرار می‌گیرد. اگر می‌خواهید

این فاصله در ابتدا یا انتهای صفحه محفوظ بماند، از شکل ستاره‌دار این فرمان، `\vspace*`، به جای `\vspace` استفاده کنید.

از فرمان `\stretch`، به همراه `\pagebreak` برای نوشتن متن در آخرین سطر یک صفحه یا وسط صفحه استفاده کنید.

Some text \ldots

`\vspace{\stretch{1}}`

This goes onto the last line of the page.\pagebreak

فاصله اضافی بین دو سطر از یک پاراگراف یا یک جدول با فرمان زیر تولید می‌شود.

`\[length]`

با `\bigskip` و `\smallskip` می‌توانید یک فاصله عمودی از پیش تعریف شده را بدون نگرانی از مقدار دقیق آنها تولید کنید.

## ۴.۶ طرح صفحه

لاتک اجازه می‌دهد اندازه صفحه را با فرمان `\documentclass` تعیین کنید. در این صورت لاتک حاشیه مناسب را به طور خودکار تعیین می‌کند، اما گاهی اوقات اندازه پیش‌فرض مطلوب شما نیست. به طور طبیعی می‌توان آنها را تغییر داد. شکل ۲.۶ تمام پارامترهای قابل تغییر را نشان می‌دهد. این شکل با بسته `layout` از کلاف `tools` تولید شده است.<sup>۳</sup>

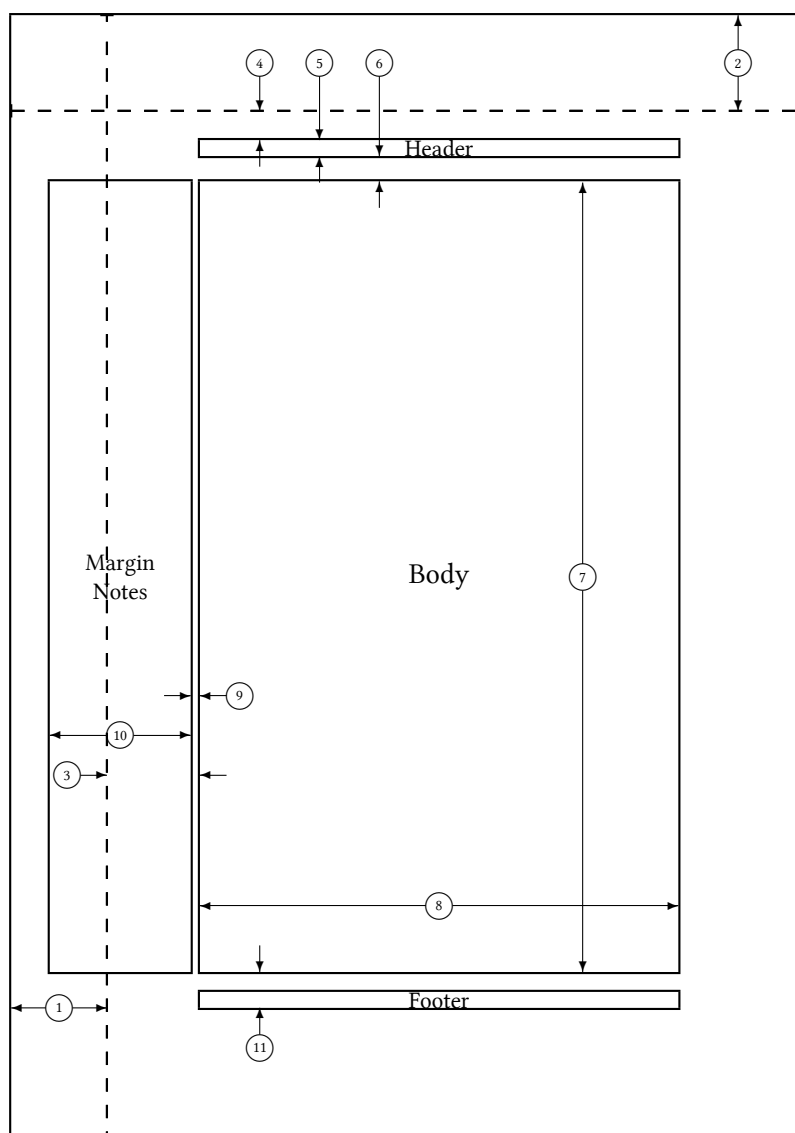
دست نگهدارید! ... قبل از این که اندازه صفحه را کوچک یا بزرگ کنید کمی فکر کنید. همانند دیگر چیزها در لاتک، دلایل قانع‌کننده‌ای برای تغییر ندادن اندازه پیش‌فرض وجود دارد.

مطمئناً، نسبت به صفحه MS Word، صفحه پیش‌فرض لاتک باریک‌تر است. اما نگاهی به یک کتاب مورد علاقه خود ببینید<sup>۴</sup> و تعداد حروف موجود در یک سطر را بشمارید. خواهید دید که این تعداد حدود ۶۶ است. حال همین تعداد را در صفحه لاتک محاسبه کنید. خواهید دید که این تعداد هم حدود ۶۶ است. تجربه نشان داده است که اگر این تعداد بیش از ۶۶ باشد خواندن سطر مشکل است. دلیل این موضوع این است که رفتن دید از انتهای یک سطر به ابتدای سطر دیگر در سطرهای با بیش از ۶۶ حرف سخت است. به همین دلیل است که روزنامه‌ها هم چند ستونی چاپ می‌شوند. بنابراین توجه داشته باشید که اگر اندازه صفحه را تغییر دهید، زندگی را برای خوانندگان مقاله یا کتاب سخت کرده‌اید. ولی روش تغییر را به شما خواهم گفت.

لاتک دو فرمان برای این کار دارد. این فرمان‌ها در سرآغاز ظاهر می‌شوند.

<sup>۳</sup> `macros/latex/required/tools`

<sup>۴</sup> منظورم یک کتاب واقعی است که توسط یک انتشارات معتبر چاپ شده باشد.



- |    |                                             |    |                                  |
|----|---------------------------------------------|----|----------------------------------|
| 1  | one inch + \hoffset                         | 2  | one inch + \voffset              |
| 3  | \oddsidemargin = 22pt<br>or \evensidemargin | 4  | \topmargin = 22pt                |
| 5  | \headheight = 12pt                          | 6  | \headsep = 19pt                  |
| 7  | \textheight = 595pt                         | 8  | \textwidth = 360pt               |
| 9  | \marginparsep = 7pt                         | 10 | \marginparwidth = 106pt          |
| 11 | \footskip = 27pt                            |    | \marginparpush = 5pt (not shown) |
|    | \hoffset = 0pt                              |    | \voffset = 0pt                   |
|    | \paperwidth = 597pt                         |    | \paperheight = 845pt             |

شکل ۲.۶: پارامترهای طرح صفحه

اولین فرمان به هرکدام از پارامترها مقدار ثابتی نسبت می‌دهد:

```
\setlength{parameter}{length}
```

فرمان دوم مقداری را به هرکدام از پارامترها اضافه می‌کند.

```
\addtolength{parameter}{length}
```

فرمان دوم مفیدتر از `\setlength` است، زیرا می‌توانید نسبت به مقادیر پیش فرض تغییر دهید. برای افزودن یک سانتیمتر به عرض کل متن، فرمان زیر را در سرآغاز قرار می‌دهیم:

```
\addtolength{\hoffset}{-0.5cm}
```

```
\addtolength{\textwidth}{1cm}
```

در این راستا بهتر است به بسته `calc` نیز نگاهی بیندازید. این بسته به شما امکان انجام تغییرات تابعی بر آرگومان‌های `\setlength` را می‌دهد.

## ۵.۶ بازی بیشتر با طول‌ها

هر جا که ممکن باشد، از قرار دادن مقدار دقیق طول‌ها در نوشتار خودداری کنید. در عوض، سعی کنید از مقادیر تعریف شده استفاده کنید. برای قرار دادن یک تصویر به گونه‌ای که عرض آن به اندازه عرض نوشتار باشد از `\textwidth` استفاده کنید.

سه فرمان زیر اجازه می‌دهد شما عرض، ارتفاع و عمق یک رشته را تعیین کنید.

```
\settoheight{variable}{text}
```

```
\settodepth{variable}{text}
```

```
\settowidth{variable}{text}
```

مثال زیر کاربردی از این فرمان‌ها را نشان می‌دهد.

```
\flushleft
\newenvironment{vardesc}[1]{%
 \settowidth{\parindent}{#1:\ }
 \makebox[0pt][r]{#1:\ }}{}
```

```
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}
```

```
\begin{vardesc}{Where}a,
b -- are adjoin to the right
angle of a right-angled triangle.
```

```
c -- is the hypotenuse of
the triangle and feels lonely.
```

```
d -- finally does not show up
here at all. Isn't that puzzling?
\end{vardesc}
```

$a^2 + b^2 = c^2$

Where:  $a, b$  – are adjoin to the right angle of a right-angled triangle.

$c$  – is the hypotenuse of the triangle and feels lonely.

$d$  – finally does not show up here at all. Isn't that puzzling?

## ۶.۶ جعبه‌ها

لاتک با قراردادن جعبه‌هایی طرح صفحه را مشخص می‌کند. در ابتدا هر حرف یک جعبه کوچک دارد که از چسبیدن این جعبه‌ها کلمه‌ها درست می‌شوند. اینها هم به همدیگر می‌چسبند تا سطرها را تشکیل دهند ولی روش چسباندن کلمه‌ها کمی پیچیده است تا انعطاف لازم را برای پرکردن سطرها داشته باشند.

قبول دارم که این توضیح ساده‌ای است از آنچه اتفاق می‌افتد، اما نکته این است که تک مسئولیت چسباندن را دارد. می‌توانید هر چیزی، از جمله جعبه‌های دیگر را در یک جعبه قرار دهید. هر جعبه در این صورت همانند یک حرف عمل می‌کند.

در فصل‌های پیشین با جعبه‌های واقعی روبرو شده‌اید، هرچند به شما نگفتم. محیط `tabular` و `\includegraphics` از این نوع هستند که جعبه تعریف می‌کنند. این به آن معنی است که می‌توانید جدول‌ها را در کنار هم قرار دهید. فقط باید مواظب باشید مجموع عرض آنها از عرض متن بیشتر نباشد.

همچنین می‌توانید یک پاراگراف را به شکل زیر در یک جعبه قرار دهید.

```
\parbox[pos]{width}{text}
```

یا به طریق زیر این کار را انجام دهید.

```
\begin{minipage}[pos]{width} text \end{minipage}
```

پارامتر `pos` می‌تواند یکی از مقادیر `c`، `t` یا `b` را بپذیرد که جهت چیدن جعبه را نسبت به متن پیرامون آن مشخص می‌کند. `width` یک مقدار طول مربوط به عرض جعبه را می‌پذیرد. مهمترین تفاوت بین `\parbox` و `minipage` این است که نمی‌توانید تمام فرمان‌ها و محیط‌ها را داخل `\parbox` استفاده کنید درحالی که این کار در `minipage` امکان‌پذیر است.

درحالی که `\parbox` تمام امکانات شکستن خط را پشتیبانی می‌کند، تعدادی از فرمان‌های جعبه هستند که تنها در متن‌های افق‌چین امکان‌پذیرند. یکی از آنها را می‌شناسیم؛ `\mbox` که تعدادی از جعبه‌ها را درون هم قرار می‌دهد و برای جلوگیری از شکستن کلمه‌ها مورد استفاده قرار می‌گیرد. از آنجا که می‌توانید جعبه‌ها را درون هم قرار دهید، این ویژگی انعطاف زیادی به کار شما می‌دهد.

```
\makebox[width][pos]{text}
```

`width` عرض جعبه را از بیرون نشان می‌دهد<sup>۵</sup>. به جز طول عبارت، می‌توانید عرض (`\width`)، ارتفاع (`\height`)، عمق (`\depth`)، و ارتفاع کلی (`\totalheight`) را در پارامتر عرض تغییر دهید. این مقادیر با مقایسه متن تعیین می‌شوند. پارامتر `pos` یک مقدار تک‌حرفی را می‌پذیرد: `c` برای وسط، `l` برای چپ، `r` برای راست، یا `s` برای توزیع متن در جعبه.

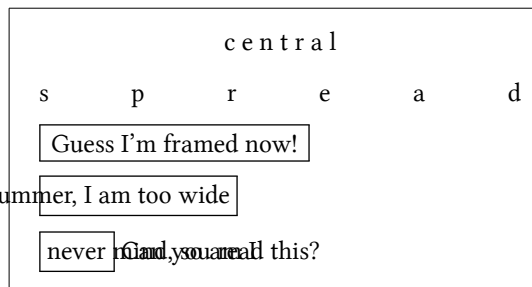
فرمان `\framebox` دقیقاً همانند `\makebox` استفاده می‌شود، اما کادری پیرامون جعبه رسم می‌کند.

مثال زیر چند کار را نشان می‌دهد که با `\makebox` و `\framebox` می‌توان انجام داد.

<sup>۵</sup> این به آن معنی است که می‌تواند کوچک‌تر از متن پیرامونش باشد. حتی می‌توانید عرض را برابر صفر پوینت تعریف کنید تا متن داخل جعبه بدون اثر جانبی روی جعبه محیطی قرار داده شود.



```
\makebox[\textwidth]{%
 c e n t r a l}\par
\makebox[\textwidth][s]{%
 s p r e a d}\par
\framebox[1.1\width]{Guess I'm
 framed now!} \par
\framebox[0.8\width][r]{Bummer,
 I am too wide} \par
\framebox[1cm][l]{never
 mind, so am I}
Can you read this?
```



حال که حالت افقی را کنترل کردیم، قدم بعدی کنترل حالت عمودی است. ۶

```
\raisebox{lift}[extend-above-baseline][extend-below-baseline]{text}
```

این فرمان به شما اجازهٔ تعریف خواص عمودی جعبه را می‌دهد. دوباره می‌توانید عرض، ارتفاع، عمق، و ارتفاع کلی را در سه پارامتر اول تعیین کنید.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}}
he shouted but not even the next
one in line noticed that something
terrible had happened to him.
```

## \strut و \rule ۷.۶

چند صفحهٔ قبل ممکن است به فرمان زیر توجه کرده باشید.

```
\rule[lift]{width}{height}
```

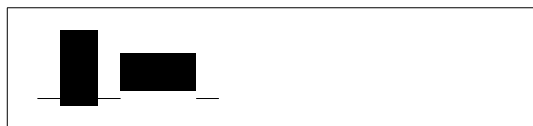
در حالت نرمال این فرمان یک جعبهٔ سیاه تولید می‌کند.

کنترل واقعی با کنترل همزمان افقی و عمودی بدست می‌آید.

```

\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}

```



این کار برای رسم خط‌های افقی و عمودی مناسب است. خط سیاه در عنوان این مقدمه با فرمان `\rule` رسم شده است.

یک حالت ویژه این است که یک خط بدون عرض ولی با یک ارتفاع مشخص رسم کنیم. در حروف چینی حرفه‌ای به چنین چیزی `strut` می‌گویند. کاربرد آن برای این است که شیئی ویژه‌ای دارای حداقل مشخصی از ارتفاع باشد. می‌توانید آن را در یک محیط `tabular` به‌کار برید تا مطمئن شوید یک سطر دارای یک حداقل ارتفاع مشخص باشد.

```

\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\
\hline
\rule{0pt}{4ex}Strut\
\hline
\end{tabular}

```



پایان.

## کتاب نامه

- [1] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The TeXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The LaTeX Companion, (2nd Edition)*. Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [5] Each LaTeX installation should provide a so-called *LaTeX Local Guide*, which explains the things that are special to the local system. It should be contained in a file called `local.tex`. Unfortunately, some lazy sysops do not provide such a document. In this case, go and ask your local LaTeX guru for help.
- [6] LaTeX3 Project Team. *LaTeX 2<sub>ε</sub> for authors*. Comes with the LaTeX 2<sub>ε</sub> distribution as `usrguide.tex`.
- [7] LaTeX3 Project Team. *LaTeX 2<sub>ε</sub> for Class and Package writers*. Comes with the LaTeX 2<sub>ε</sub> distribution as `clsguide.tex`.
- [8] LaTeX3 Project Team. *LaTeX 2<sub>ε</sub> Font selection*. Comes with the LaTeX 2<sub>ε</sub> distribution as `fntguide.tex`.
- [9] D. P. Carlisle. *Packages in the 'graphics' bundle*. Comes with the 'graphics' bundle as `grfguide.tex`, available from the same source your LaTeX distribution came from.

- [10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of  $\text{\LaTeX}$ 's verbatim Environments*. Comes with the 'tools' bundle as `verbatim.dtx`, available from the same source your  $\text{\LaTeX}$  distribution came from.
- [11] Vladimir Volovich, Werner Lemberg and  $\text{\LaTeX}$ 3 Project Team. *Cyrillic languages support in  $\text{\LaTeX}$* . Comes with the  $\text{\LaTeX}$ 2 $\epsilon$  distribution as `cyrguide.tex`.
- [12] Graham Williams. *The TeX Catalogue* is a very complete listing of many  $\text{\TeX}$  and  $\text{\LaTeX}$  related packages. Available online from [CTAN:/tex-archive/help/Catalogue/catalogue.html](http://ctan.org/help/Catalogue/catalogue.html)
- [13] Keith Reckdahl. *Using EPS Graphics in  $\text{\LaTeX}$ 2 $\epsilon$  Documents*, which explains everything and much more than you ever wanted to know about EPS files and their use in  $\text{\LaTeX}$  documents. Available online from [CTAN:/tex-archive/info/epslatex.ps](http://ctan.org/info/epslatex.ps)
- [14] Kristoffer H. Rose. *X<sub>y</sub>-pic User's Guide*. Downloadable from CTAN with X<sub>y</sub>-pic distribution
- [15] John D. Hobby. *A User's Manual for METAPOST*. Downloadable from <http://cm.bell-labs.com/who/hobby/>
- [16] Alan Hoenig.  *$\text{\TeX}$  Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [17] Urs Oswald. *Graphics in  $\text{\LaTeX}$ 2 $\epsilon$* , containing some Java source files for generating arbitrary circles and ellipses within the `picture` environment, and *METAPOST - A Tutorial*. Both downloadable from <http://www.ursoswald.ch>
- [18] Till Tantau. *TikZ&PGF Manual*. Download from [CTAN:/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf](http://ctan.org/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf)