

به نام هستی بخش



آشنایی با $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

و

فارسی‌تک

دکتر بهناز عمومی

ویرایش و طرح روی جلد

رامین جوادی

دانشکده علوم ریاضی

دانشگاه صنعتی اصفهان

فهرست مندرجات

۷	متن‌های راسته	۱
۷ ساختار کلی فایل‌های $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$	۱.۱
۷ پیش درآمد	۲.۱
۸ متن راسته	۳.۱
۱۱ انتخاب قلم	۴.۱
۱۲ نقش آکولادها	۵.۱
۱۳ فاصله و واحدهای اندازه‌گیری	۶.۱
۱۵ قسمت بندی	۷.۱
۱۵ فهرست‌ها	۸.۱
۱۷	متن ریاضی	۲
۱۷ زیرنویس و بالانویس	۱.۲
۱۸ سه نقطه‌ها	۲.۲
۱۹ رادیکال‌ها	۳.۲
۱۹ کسرها	۴.۲
۲۰ خط بالا و پایین	۵.۲
۲۱ اکسانت‌ها	۶.۲
۲۳ حروف تحریری	۷.۲
۲۴ نفی رابطه‌ها	۸.۲
۲۴ نمادهای \int و \sum	۹.۲
۲۵ شماره‌گذاری فرمول‌ها	۱۰.۲
۲۶ نمادها و توابع ریاضی	۱۱.۲
۲۷ ماتریس‌ها	۱۲.۲
۳۰ جدول‌ها	۱۳.۲
۳۱ مراجعه متقابل (cross reference)	۱۴.۲
۳۳	محیط‌ها	۳
۳۴ محیط <code>varbatim</code>	۱.۳
۳۵ محیط <code>tabbing</code>	۲.۳
۳۵ محیط‌های <code>eqnarray</code> و <code>array</code>	۳.۳

۳۷ picture	محیط	۴.۳
۳۹ theorem	محیط	۵.۳
۴۰ عناصر شناور		۶.۳
۴۲ thebibliography	محیط	۷.۳
۴۳		سبک‌ها	۴
۴۳ سبک نوشتار		۱.۴
۴۴ سبک‌های فرعی		۲.۴
۴۵ bezier	سبک فرعی	۳.۴
۴۶ استفاده از شکل یا عکس در متن		۴.۴
۴۷		فارستیک	۵
۴۷ مقدمه		۱.۵
۴۸ FarsiTeX	فارستیک	۲.۵
۴۸ سبک کلی فایل‌های فارستیک		۳.۵
۵۱		جدول‌ها	۶

تاریخچه

پیدایش زبان حروفچینی TeX از آنجا آغاز شد که چاپ اثر چند جلدی Donald E. Knuth با عنوان «هنر برنامه نویسی کامپیوتر» مشکل بزرگی برای نویسنده‌اش ایجاد کرد، زیرا جلدها از نظر چاپی یکسان نبودند. برای حل این مشکل، Knuth تصمیم گرفت یک زبان حروفچینی به وجود آورد. Knuth در یکی از کتابهای خود می‌نویسد: «من متوجه شدم که جنبه اصلی چاپ در حد دست‌کاری بیت‌ها تنزل مقام داشته، من به عنوان یک دانشمند علوم کامپیوتر نمی‌توانم در مقابل چالش‌های مربوط به بهبود کیفیت چاپ به واسطه دست‌کاری بیت‌ها، مقاومت نمایم.» او معتقد بود که حروفچینی تنها نیمی از مشکل نوشتن نسخه‌های خطی را حل نموده است.

مقدمه

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ چیست؟

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ چیزی جز مجموعه‌ای از تعدادی تعریف و فرمان اضافه شده به TeX نیست. این تعریف‌ها و فرمان‌های جدید، استفاده از TeX را بسیار راحت‌تر می‌کنند و اسلوب منظم‌تری به آن می‌بخشند، به طوری که اولاً برای تهیه بسیاری از نوشتارها فرمان‌های کمتر و راحت‌تری لازم است، ثانیاً شکل فرمان‌ها برای کاربران معمولی قابل هضم‌تر می‌شود.

البته $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ معایبی نیز به همراه خود می‌آورد که مهمترین عیب آن پنهان کردن بسیاری از تواناییهای مختلف TeX از دید کاربران است، به طوری که برخی از کاربران به اشتباه تصور می‌کنند که منظور از TeX همان فرمان‌های راحت در $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ است. لیزلی لمپرت، خالق اصلی $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ، در مورد آن می‌گوید: « TeX همانند یک ماشین مسابقه قدرتمند و فوق‌العاده است که من این ماشین مسابقه را به اتومبیلی خانوادگی تبدیل کرده‌ام تا شما به راحتی بتوانید با آن در داخل شهر خرید کنید». الگوی لمپرت برای ساختن $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ، مرجع [۳] بوده است و اکثر فرمان‌ها همانند قراردادهای موجود در این مرجع تعریف شده‌اند. پدید آمدن $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ نتیجه یکی از قابلیت‌های بسیار جالب TeX است. این قابلیت به طراحان کتاب اجازه می‌دهد که شکل‌بندی دلخواه خود را به جای شکل‌بندی ساده TeX ، یعنی Plain، به کار برند. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ اغلب مورد استفاده ریاضی‌دانان، فیزیک‌دانان، مهندسين، دانشمندان و ماشین‌نویسان فنی است که مجبور به یادگیری چگونگی حروفچینی مقالات محتوی فرمول‌های ریاضی هستند.

فصل ۱

متن‌های راسته

۱.۱ ساختار کلی فایل‌های $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

یک متن در $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ در قالب فایل با TEX ، Extention ، TEX تدوین می‌شود. این فرمت را می‌توان در ادیتورهای Winedit یا PCTex ایجاد کرد. هر فایل $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ از دو قسمت اصلی تشکیل شده است: الف) پیش درآمد (Preamble): در این قسمت مشخصات کلی قالب (Format) متن آورده می‌شود. ب) متن (document): که همان متنی است که می‌خواهیم چاپ کنیم. پیش درآمد همواره پیش از متن می‌آید. متن با $\backslash\text{begin}\{\text{document}\}$ شروع و با $\backslash\text{end}\{\text{document}\}$ ختم می‌شود.

```

: Preamble
\begin{document}
:
\end{document}
```

۲.۱ پیش درآمد

یک مثال از یک پیش درآمد در زیر آمده است:

مثال .

```
\documentstyle[12pt]{article}
\textwidth = 6in
\textheight = 8.7in
\renewcommand{\baselinestretch}{1.3}
```

دستور اول که باید در ابتدای هر فایل $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ بیاید، سبک (style) متن را مشخص می‌کند. سبک‌های مجاز در $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ عبارتند از: article، book، letter و report. سبک article برای مقاله‌ها و سبک letter برای نامه‌ها به کار می‌روند. دو سبک دیگر برای جزوه‌های مفصل و کتاب‌ها استفاده می‌شوند. شناسه $[12\text{pt}]$ اندازه حروف

مقاله را تعیین می‌کند. این شناسه اختیاری است و می‌توان آن را حذف کرد، که در آن صورت مقاله با حروف ۱۰ پوینت چاپ می‌شود. شناسه‌های [11pt] و [12pt] برای چاپ متن با حروف ۱۱ و ۱۲ پوینت منظور شده‌اند. دو دستور بعد که اختیاری هستند به ترتیب عرض و طول یک صفحه متن چاپی را مشخص می‌کنند. در این مثال عرض و طول متن به ترتیب ۶ و ۸.۷ اینچ مشخص شده‌اند. می‌توان از واحدهای میلیمتر و سانتیمتر نیز به ترتیب با کدهای mm و cm استفاده کرد. باید تذکر داد که به هنگام چاپ، این اندازه‌ها دقیقاً برابر آن چیزی که مشخص شده‌اند از کار در نخواهند آمد، بلکه در ضریبی که به نوع دستگاه چاپگر بستگی دارد ضرب می‌شوند. دستور آخر برای کنترل فاصله بین سطرها به کار رفته و صورت کلی آن چنین است:

```
\renewcommand{\baselinestretch}{d}
```

این دستور باعث می‌شود فاصله بین سطرها d برابر فاصله‌ای شود که در حالت معمولی در نظر می‌گیرد. مثلاً اگر d برابر 2 قرار گیرد، متن به صورت double space چاپ می‌شود. در مثال فوق، فاصله سطرها ۱.۳ برابر فاصله معمولی می‌شود. این دستور نیز اختیاری است.

حاشیه‌ها (margins)

به طور کلی برای افزایش یا کاهش طول، عرض، ارتفاع و حاشیه صفحه می‌توان از دستور زیر استفاده کرد:

```
\addtolength{length command}{length}
```

این افزایش یا کاهش با اختصاص مقدار طول مورد نظر به آرگومان length صورت می‌گیرد (مقدار طول می‌تواند منفی باشد). آرگومان length command اجباری است و به عنوان یک دستور طول مانند `\textheight` است. اما در صورتی که این دستور قبلاً موجود باشد خطا رخ می‌دهد. قابل ذکر است که دستور `\addtolength` در قسمت پیش درآمد آورده می‌شود.

مثال .

```
\addtolength{\textwidth}{2cm}
\addtolength{\textheight}{2cm}
\addtolength{\topmargin}{0.5cm}
\addtolength{\oddsidemargin}{-2cm}
```

همچنین با دستورات زیر می‌توان اندازه حاشیه‌ها را تعیین کرد.

```
\textwidth=...cm
\textheight=...cm
\topmargin=...cm
\oddsidemargin=...cm
\evensidemargin=...cm
```

۳.۱ متن راسته

همان‌طور که گفتیم متن با `\begin{document}` شروع و با `\end{document}` ختم می‌شود. آن بخش‌هایی از متن را که شامل فرمول‌ها و نمادهای ریاضی هستند متن ریاضی می‌گوییم و بخش‌های دیگر را که شامل حروف معمولی لاتین و ارقام هستند متن راسته (LR text) می‌نامیم. برای نوشتن یک فرمول ریاضی درون یک متن

راسته، آن را بین دو علامت $\$$ قرار می‌دهیم و برای نوشتن یک فرمول در یک سطر جداگانه آن را بین دو علامت $\$\$$ قرار می‌دهیم. در متن راسته می‌توان هر کاراکتری را که به ازای آن کلیدی روی صفحه کلید کامپیوتر وجود دارد چاپ کرد. برای این کار کافی است آن کاراکتر را عیناً در فایل `tex` در قسمت متن وارد کرد. البته ۱۰ کاراکتر زیراستثنا هستند:

% \$ & _ { } ^ ~ \

این کاراکترها در دستورهایی $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ به منظورهایی خاصی به کار می‌روند. به عنوان مثال هر عبارتی که با کاراکتر `\` شروع شود از نظر $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ یک دستور است و اگر این دستور قبلاً تعریف نشده باشد، در زمان اجرا خطا محسوب می‌شود. برای چاپ هشت کاراکتر اول باید به ترتیب از دستورهایی زیر استفاده کرد:

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

به عنوان مثال برای چاپ عبارت $\{A \backslash B\}$ باید تایپ کنیم:

$\$ \{ A \backslash B \} \$$

همان‌طور که گفتیم به جز ده کاراکتر فوق بقیه کاراکترهای موجود در خانواده حروف در صورتی که فرمان نباشند عیناً در خروجی ظاهر می‌شوند. به متن قضیهٔ تیلور توجه کنید:

```
{\bf 5.15 Theorem}\hspace 0.5cm {\it Suppose
\bf} is a real function on  $[a,b]$ ,  $n$  is a
positive integer,  $f^{(n-1)}$  is continuous
on  $[a,b]$ ,  $f^{(n)}$  exists for every  $t \in (a,b)$ .
Let  $\alpha, \beta$  be distinct points of  $[a,b]$ 
and define  $p(t) = \sum_{k=0}^{n-1} \frac{f^{(k)}(\alpha)}{k!} (t-\alpha)^k$ .
Then there exists a point  $x$  between  $\alpha$ 
and  $\beta$  such that
 $f(\beta) = p(\beta) + \frac{f^{(n)}(x)}{n!} (\beta - \alpha)^n$ .
}

```

5.15 Theorem Suppose f is a real function on $[a, b]$, n is a positive integer, $f^{(n-1)}$ is continuous on $[a, b]$, $f^{(n)}$ exists for every $t \in (a, b)$. Let α, β be distinct points of $[a, b]$ and define

$$(23) \quad p(t) = \sum_{k=0}^{n-1} \frac{f^{(k)}(\alpha)}{k!} (t - \alpha)^k.$$

Then there exists a point x between α and β such that

$$(24) \quad f(\beta) = p(\beta) + \frac{f^{(n)}(x)}{n!} (\beta - \alpha)^n.$$

• دستور `\hspace` در این مثال فاصله‌ای افقی به اندازهٔ داده شده ایجاد می‌کند. در مواردی مانند تعیین طول سطرها، فاصلهٔ متن از چپ و راست و بالا، مقدار تورفتگی ابتدای هر بند، بلندی متن و غیره که در حروفچینی

اهمیت دارند، مجموعهٔ فرمان‌هایی در $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ تعبیه شده‌است که می‌توان آن‌ها را به اجرا درآورد و متن ورودی را مطابق این دستورات حروفچینی کرد.

دو دستور مفید $\backslash\text{hspace}\{d\}$ و $\backslash\text{vspace}\{d\}$ برای تعیین کنترل عادی فاصله بین سطرها و کلمات به کار می‌روند. در این دستورها d با یک واحد بیان می‌شود. مثلاً نویسه‌های 2.5 یا 7mm یا 1.2cm می‌تواند به جای d قرار گیرد. دستور اول باعث ایجاد فاصله‌ای عمودی و دستور دوم باعث ایجاد فاصله‌ای افقی به اندازهٔ d خواهد شد.

- در متن راسته یک یا چند فاصله (کاراکتر خالی blank) معادلند:

مثال .

Command	typestyle
Example for Seeing Blanks.	Example for Seeing Blanks.
Example for Seeing Blanks.	Example for Seeing Blanks.

همچنین یک یا بیش از یک سطر خالی نشان‌دهندهٔ انتهای یک پاراگراف و شروع پاراگراف بعدی است. دستورهایی $\backslash\backslash$ یا $\backslash\text{newline}$ شروع خط را در ابتدای سطر بعد قرار می‌دهد. همچنین دستور $\backslash\text{newpage}$ باعث شکسته شدن یک صفحه و شروع مطالب از صفحهٔ بعدی می‌شود.

- فرمان $\backslash\text{centerline}\{text\}$ جمله را وسط سطر قرار می‌دهد:

مثال .

Command: $\backslash\text{centerline}\{\text{This } \{\textit{information}\}\text{should be } \{\text{bf centered}\}$

typestyle:

This *information* should be **centered**

علامت $\backslash/$ بعد از متن داخل آکولاد در حالتی که متن ایتالیک باشد فاصله‌ای مناسب بین این حروف و حروف بعدی که ایتالیک نیستند و به دنبال آن خواهند آمد ایجاد می‌کند. در صورت نبود این علامت، به دلیل کج بودن حروف ایتالیک ممکن است فاصله به طور نامناسبی کم شود.

۴.۱ انتخاب قلم

در حروفچینی اندازه حروف معمولاً با واحد پوینت (برابر با ۰.۰۳۵ سانتیمتر) تعریف می‌شود. قبل از انتخاب اندازه حروف، \LaTeX آن را ۱۰ پوینت فرض می‌کند. برای استفاده از حروف بزرگتر یا کوچکتر، دستورهای زیر تعریف شده‌اند:

Command	\LaTeX sample text	font size
<code>\tiny</code>	sample text	5pt
<code>\scriptsize</code>	sample text	7pt
<code>\footnotesize</code>	sample text	8pt
<code>\small</code>	sample text	9pt
<code>\normalsize</code>	sample text	10pt
<code>\large</code>	sample text	12pt
<code>\Large</code>	sample text	13.14pt
<code>\LARGE</code>	sample text	17.28pt
<code>\huge</code>	sample text	20.74pt
<code>\Huge</code>	sample text	24.88pt

به طور کلی برای فرمول نویسی در \TeX از سه اندازه قلم استفاده می‌شود:

T	textsize	(Like This)
S	scriptsize	(Like This)
SS	scriptscriptsize	(Like This)

اندازه SSS یا اندازه `scriptscriptsize` وجود ندارد زیرا در این اندازه نمادها به قدری نازک می‌شوند که دیگر قابل خواندن نخواهند بود. بنابراین در این حالت نیز از اندازه قلم سبک SS استفاده می‌شود. برای مثال اگر $x^{\{a_b\}}$ در فرمول باشد، در این صورت قلم x از نوع T، a از نوع S، b از نوع SS است و نتیجه خروجی x^{a_b} است. در $a^{b^{c^d}}$ با قلم T، b با قلم S و c و d و e همگی با قلم SS هستند. در متن ریاضی دوسبک نمایشی و داخلی وجود دارد. در حالت پیش فرض فرمول‌هایی که درون متن هستند با سبک داخلی و فرمول‌هایی که در یک سطر جدا هستند با سبک نمایشی نشان داده می‌شوند. در هر دوسبک قلم از نوع T است به جز در نمایش کسرها که در سبک نمایشی از نوع T و در سبک داخلی از نوع S است. مثلاً $a \over b$ به صورت $\frac{a}{b}$ نمایش داده شود و در $a \over b$ نتیجه به صورت زیر است:

$$\frac{a}{b}$$

ذکر این نکته لازم است که دسته بندی فوق صورت ساده شده دسته بندی اصلی در \TeX است. در \TeX این کار با ظرافت بیشتری انجام می‌شود که شرح آن را می‌توانید در [۴] ببینید. در صورتی که نخواهید \TeX به طور خودکار قلم را انتخاب کند، می‌توانید با کمک فرمان‌های `\displaystyle`، `\textstyle`، `\scriptstyle` و `\scriptscriptstyle` سبک مورد نظر را انتخاب کنید. مانند:

$\$n+\scriptstyle n+\scriptscriptstyle n\$\$$

$$n + n + n$$

شکل قلم‌ها را نیز می‌توان به وسیلهٔ دستورهای `\rm`، `\bf`، `\it`، `\sl`، `\sc`، `\sf` و `\tt` تغییر داد. مثال‌های زیر به خوبی نحوهٔ استفاده از این دستورها و نمونهٔ خط آن‌ها را نشان می‌دهد:

مثال .

Command	type style
<code>{\rm roman type style}</code>	roman type style
<code>{\sl slanted type style}</code>	<i>slanted type style</i>
<code>{\it italic type style}</code>	<i>italic type style</i>
<code>{\bf bold type style}</code>	bold type style
<code>{\sf sans serif type style}</code>	sans serif type style
<code>{\tt type writer type style}</code>	type writer type style
<code>{\sc small capital type style}</code>	SMALL CAPITAL TYPE STYLE
<code>{\em italic type style}</code>	<i>italic type style</i>

دقت کنید که دستورهای `\em` و `\it` معادل‌اند. در متن راسته پیش فرض قلم، `roman` است و نیازی به استفاده از `\rm` نیست. در واقع کاربرد دستور `\rm` در متن‌های ریاضی و در مواردی است که قلم متن غیر از `Roman` است. گاهی به دنبال یک کلمه، عدد یا حرفی یا کلمه‌ای می‌آید که بهتر است در موقع حروفچینی در سطر متفاوتی نباشد، مثلاً 'Donald E. Knuth'؛ در این موارد از کاراکتر `~` برای جدا نشدن کلمه از حرف یا کلمهٔ بعدی استفاده می‌شود:

مثال .

Command:

`\TeX\` was introduced by Donald~E. Knuth for the first time.

Typestyle:

`TEX` was introduced by Donald E. Knuth for the first time.

Command:

`\TeX\` was introduced by Donald E. Knuth for the first time.

Typestyle:

`TEX` was introduced by Donald

E. Knuth for the first time.

۵.۱ نقش آکولادها

در اینجا مطلب مهمی که لازم است به آن اشاره شود، نقش آکولادها در پرونده‌های ورودی `LATEX` یا `TEX` است. به طور کلی آکولادها در `TEX` دو وظیفهٔ اصلی را به عهده دارند. اولین وظیفهٔ آن‌ها «دسته بندی» است، که عبارت است از جداکردن پارامترهای فرمان‌ها از یکدیگر و از متن معمولی. مثلاً در فرمان

`\documentstyle{article}` کلمه 'article' که پارامتر فرمان است توسط آکولادها از بقیه ورودی جدا شده است.

به وجود آوردن «محدوده» دومین وظیفه اصلی آکولادها است. محدوده یا به قول برنامه‌نویسان «بلوک»، باعث می‌شود تعریف‌ها و تغییراتی که در داخل آن محدوده اعمال می‌شوند در خارج از محدوده اثری نداشته باشند. مثلاً اگر در داخل یک محدوده اندازه یا شکل قلم را تغییر دهیم، هنگامی که محدوده به پایان می‌رسد اندازه یا شکل قلم به حالت اولیه (حالتی که قبل از محدوده وجود داشته است) برمی‌گردد. مثلاً به کار بردن فرمان `\large` به صورت `\large{text}` برای استفاده از حروف `large` در متن درست نیست و باید به صورت `{\large text}` نوشته تا فقط اندازه قلم متن داخل آکولادها تغییر کند.

مثال.

Command:

D.E. Knuth. `{\it The TeXbook.}`
Amer. Math. Soc., 1984.

Typestyle:

D.E. Knuth. *The TeXbook*. Amer. Math. Soc., 1984.

۶.۱ فاصله و واحدهای اندازه گیری

اغلب نرم افزارهای کامپیوتری نشر هنگام صفحه‌بندی متن، هر سطر را با طول معین حروفچینی می‌کنند و سپس با قرار دادن سطرها به دنبال هم، صفحه‌های خروجی را آماده می‌کنند. `TEX` نیز این عمل را با اضافه کردن فاصله‌هایی جزئی بین کلمه‌های موجود در هر سطر انجام می‌دهد و این فاصله‌ها را طوری تنظیم می‌کند تا بهترین و زیباترین حالت ممکن برای هر پاراگراف به دست آید.

سطرها نیز معمولاً در فاصله‌های بین کلمه‌ها شکسته می‌شوند. یادآور می‌شویم اگر تمایلی برای شکسته شدن خط در مکان معینی نداشته باشیم، به جای فاصله از نویسه `~` استفاده می‌کنیم. ضمناً پس از نویسه‌های `،` `:` `؛` `؛` و `،` به طور خودکار فاصله بیشتری قرار داده می‌شود تا به خوانایی متن اضافه شود؛ اگر می‌خواهید این فاصله اضافی حذف شود قبل از این نویسه‌ها فرمان `\@` را قرار دهید یا به جای فاصله از `~` استفاده کنید.

قبلاً اشاره کردیم که وجود یک یا چند فاصله بین دو کلمه همانند یک فاصله است. همچنین فاصله‌هایی که پس از یک فرمان و برای جدا کردن آن از بقیه متن قرار می‌گیرند نادیده گرفته می‌شوند.

مثال.

Command:

How `\LaTeX` treats spaces and blank lines?

Typestyle:

How `\LaTeX` treats spaces and blank lines?

اشکالی که در این مثال وجود دارد نبودن فاصله بین دو کلمه `\LaTeX` و `treats` است. در مواردی که می‌خواهیم در یک محل بیش از یک فاصله قرار دهیم، یا به طور قطع فاصله‌ای قرار دهیم، پس از هر فرمان از فرمان `\ \` (برای نمایش فاصله به کار می‌رود) استفاده می‌کنیم. بنابراین راه پیشگیری از اشتباه به وجود آمده قرار دادن `\ \` پس از فرمان `\LaTeX` است.

قبلاً اشاره کردیم که در متن اصلی وظیفهٔ فرمان `\` پایان دادن به سطرهاست و متن پس از این فرمان، از ابتدای سطر بعدی آغاز می‌شود. ضمناً اگر بخواهیم فاصلهٔ بین دو سطر را افزایش دهیم، می‌توانیم مقدار این فاصلهٔ اضافی را در بین یک جفت کروشه پس از `\` بنویسیم. همچنین اگر نخواهیم بلافاصله پس از این فرمان صفحهٔ جدید شروع شود، باید نویسهٔ * را پس از این فرمان قرار دهیم.

ذکر این مطلب لازم است که در TeX سه دستهٔ مختلف واحد اندازه‌گیری وجود دارد:

(۱) واحد اندازه‌گیری مطلق،

(۲) واحد اندازه‌گیری وابسته به حروف،

(۳) واحد اندازه‌گیری نامحدود.

برخی از واحدهای اندازه‌گیری مطلق را در جدول زیر آورده‌ایم. این واحدها در هر شرایطی مقدار ثابت دارند.

	پوینت	اینچ	سانتیمتر	میلیمتر
pt	1	0.014	0.035	0.351
In	72.27	1	2.54	25.4
cm	28.453	0.393	1	10
mm	2.845	0.093	0.1	1

دومین دسته از واحدهای اندازه‌گیری وابسته به اندازهٔ حروف است. به عنوان مثال `em` و `ex` دو واحد از این دسته‌اند. واحد `em` به اندازهٔ پهنای حرف `M` و `ex` به اندازهٔ بلندی حرف `X` است. مثلاً برای حروف `۱۰` پوینت، هر واحد `em` برابر `10.00002pt` است. از این دسته، `mu` واحد اندازه‌گیری است که در حالت ریاضی استفاده می‌شود. توضیح بیشتر راجع به این واحدها و واحدهای اندازه‌گیری دستهٔ سوم را در [۴] ببینید.

قبلاً اشاره شد که اگر زمانی بخواهیم بین دو کلمه فاصلهٔ افقی با اندازهٔ معین قرار دهیم، از فرمان `\hspace{اندازه}` استفاده می‌کنیم. برخی مواقع ممکن است این فاصله اضافه نشود، مثلاً زمانی که قبل یا بعد از این فرمان چیزی نباشد. در این مواقع از `\hspace*{اندازه}` استفاده می‌کنیم تا این فاصله حتماً گذاشته شود. به طور مشابه برای اضافه کردن فاصلهٔ بین خط جاری تا خط بعدی فرمانهای `\vspace{اندازه}` و `\vspace*{اندازه}` به کار می‌روند.

مثال .

Command:

```
This is \hspace{1.2cm} horizontal space.\\
\hspace{1cm} Command doesn't work.\\[4mm]
\hspace*{1cm} And this work properly.
```

Typestyle:

```
This is horizontal space.
Command doesn't work.
```

And this work properly.

البته در LaTeX بسیاری از کارهای مربوط به رعایت فاصله به صورت خودکار انجام می‌شود و در صورت استفاده از امکانات LaTeX به ندرت نیاز به استفاده از این فرمانها برای صفحه‌آرایی پیدا می‌شود.

۷.۱ قسمت بندی

فرمان‌های زیر برای تقسیم متن به کار می‌روند.

<code>\part</code>	
<code>\chapter</code>	<code>\section</code>
<code>\subsection</code>	<code>\subsubsection</code>
<code>\paragraph</code>	<code>\subparagraph</code>

پس از این فرمان‌ها نام قسمت مورد نظر، در داخل آکولاد آورده می‌شود. بعضی از این فرمان‌ها عنوان را نیز شماره‌گذاری می‌کنند. البته می‌توان عمق شماره‌گذاری و نحوه شماره‌گذاری قسمت‌ها را تغییر داد که توضیح آن از حوصله این نوشتار خارج است. با شروع هر قسمت، تمام زیرقسمت‌ها نیز از ابتدا شماره‌گذاری می‌شوند، به جز هنگامی که از `\part` استفاده می‌شود. این فرمان تأثیری بر شماره‌گذاری زیرقسمت‌ها نمی‌گذارد. قابل ذکر است که در سبک `article` فرمان `\chapter` وجود ندارد.

در صورتی که نوشتار شامل ضمیمه باشد، بهتر است که شماره‌گذاری ضمیمه‌ها الفبایی باشد. مانند حروف A, B, C، و غیره. برای این منظور از فرمان `\appendix` استفاده می‌شود. از جایی که این فرمان نوشته می‌شود در سبک `book` شماره‌هایی که `\chapter` تولید می‌کند و در سبک `article` شماره‌هایی که `\section` تولید می‌کند، به صورت الفبایی خواهند بود.

۸.۱ فهرست‌ها

معمولاً در ابتدای نوشتار فهرست مطالب، شکلها و جدولهای موجود در آن نوشتار می‌آید. در `LaTeX` برای تهیه این فهرست‌ها به صورت خودکار فرمان‌های `\tableofcontents`، `\listoffigures` و `\listoftables` در نظر گرفته شده‌اند. ذکر این نکته ضروری است که در هنگام حروفچینی این فهرست‌ها از اطلاعات پردازش قبلی استفاده می‌شود. بنابراین برای اطمینان از صحت اطلاعات این فهرست‌ها بهتر است که پرونده ورودی دو بار پردازش شود.

چند فرمان دیگر

برخی دیگر از فرمانهای سودمند `LaTeX` را در زیر فهرست می‌کنیم. البته هدف درج تمام فرمانهای موجود در `LaTeX` نیست.

`\footnote{متن}`: برای درج متن زیرنویس در پایین صفحه به کار می‌رود.

`\indent`: در ابتدای پاراگراف تورفتگی ایجاد می‌کند.

`\noindent`: تورفتگی ابتدای پاراگراف را حذف می‌کند.

`[عدد]\linebreak`: میزان تمایل برای شکسته شدن خط در نقطه جاری را تعیین می‌کند. (0 تا 4)

`[عدد]\nolinebreak`: میزان تمایل برای شکسته نشدن خط در نقطه جاری را تعیین می‌کند. (0 تا 4)

`[عدد]\pagebreak`: میزان تمایل برای شروع شدن صفحه در نقطه جاری را تعیین می‌کند. (0 تا 4)

`[عدد]\nopagebreak`: میزان تمایل برای شروع نشدن صفحه در نقطه جاری را تعیین می‌کند. (0 تا 4)

در صورتی که بخواهیم یک دستور را تعریف (یا تعریف مجدد) کنیم از دستورات زیر در قسمت پیش درآمد استفاده می‌کنیم.


```
\newcommand{cmd}[args]{definition}
\newcommand{cmd}[args][default]{definition}
\renewcommand{cmd}[args][default]{definition}
```

- `cmd`: نام دستور مورد نظر است که با `\` شروع می‌شود. این دستور نباید قبلاً تعریف شده باشد و نیز نباید با `\end` شروع شود (برای دستور `\renewcommand` دستور `cmd` باید قبلاً تعریف شده باشد).
- `args`: یک عدد صحیح از ۱ تا ۹ که نشان‌دهنده تعداد آرگومانهای دستور تعریف شده می‌باشد. مقدار پیش فرض این آرگومان برای یک دستور صفر است.
- `def`: اگر این پارامتر اختیاری موجود باشد به این معنی است که اولین آرگومان دستور تعریف شده اختیاری است. مقدار پیش فرض این آرگومان اختیاری "def" می‌باشد.
- `definition`: متن یا دستوری که در هر بار استفاده از دستور `cmd` در متن جانشین این دستور می‌شود. اگر دستور شما آرگومان دارد، باید در تعریف دستور به جای آرگومان شماره `i` نویسه `#i` را قرار دهیم. دستور هنگام بارگذاری، هر `#i` را با آرگومان داده شده در دستور جایگذاری می‌کند.

مثال .

```
\newcommand{\name}{1}{my name is #1}
```

با تعریف دستور فوق می‌توانیم از دستور `\name` در متن استفاده کنیم. مثلاً با نوشتن `\name{Ali}` عبارت زیر ظاهر می‌شود:

My name is Ali.

تعریف دستورات جدید با دستور `\def` نیز ممکن است. به عنوان مثال با تعریف `\def\la{\leftarrow}` در متن می‌توانیم به جای دستور `\leftarrow` از `\la` استفاده کرد.

فصل ۲

متن ریاضی

همان طور که گفتیم متن ریاضی، که شامل فرمول‌ها و نمادهای ریاضی و ماتریس‌ها است، باید در داخل یک جفت $\$$ یا $\$\$$ یا در داخل $\[...\]$ بیاید. فرمولی که در داخل یک جفت $\$$ بیاید در درون سطری که متن راسته قرار دارد ظاهر می‌شود (سبک ریاضی داخلی) و فرمولی که در داخل $\[...\]$ یا یک جفت $\$\$$ قرار دارد در سطری جداگانه و وسط سطر چاپ می‌شود (سبک ریاضی نمایشی)، مگر آن‌که با دستوری از نوع \rm یا \bf و ... نوع حروف تغییر داده شود.

در متن ریاضی کاراکترها بدون فاصله چاپ می‌شوند و فاصله‌های موجود در فایل ورودی در نظر گرفته نمی‌شوند. در صورتی که بخواهیم فاصله‌ای در فرمول ظاهر شود باید از $\ \$ (نشان‌دهنده فاصله است) استفاده شود.

مثال .

Command	type style
$\$x+2y\$$	$x + 2y$
$\$x +2y\$$	$x + 2y$
$\$x\ +2y\$$	$x + 2y$
$\$x\ +2y\$$	$x + 2y$
$\$x\ \ +2y\$$	$x + 2y$

۱.۲ زیرنویس و بالانویس

در متن ریاضی برای چاپ زیرنویس یا بالانویس باید به ترتیب از دستورهای $\{...\}$ یا $\^{\{...\}}$ استفاده کرد. در صورتیکه زیرنویس یا بالانویس تنها از یک کاراکتر تشکیل شده باشد، می‌توان آکولادها را حذف کرد.

مثال .

Command	type style
<code>\$x^2\$</code>	x^2
<code>\$x_2\$</code>	x_2
<code>\$x^{2y}\$</code>	x^{2y}
<code>\$2^{2^x}\$</code>	2^{2^x}
<code>\$2^{2^{2^x}}\$</code>	$2^{2^{2^x}}$
<code>\$y_{x_2}\$</code>	y_{x_2}
<code>\$y_{x^2}\$</code>	y_{x^2}

توجه کنید که x_yz یا x^y^z دارای ابهام است و باید با آکولاد دسته بندی شود. بنابر آنچه مورد نظر است می توان از x_{yz} یا x^{yz} یا x^{y^z} را می توان به کاربرد. استفاده هم زمان از اندیس و توان امکان پذیر است و ترتیب آن ها اهمیت ندارد:

مثال .

Command	type style
<code>\$x^2_3\$</code>	x^2_3
<code>\$x_3^2\$</code>	x_3^2
<code>\$x^{y^a_b}_{z^c_d}\$</code>	$x^{y^a_b}_{z^c_d}$

توجه کنید که دستورهای زیر نویس و بالانویس و به طور کلی دستورهای این فصل، تنها در متن ریاضی قابل قبول هستند، مگر آن که خلاف آن ذکر شود.

Command: the j^{th} row of the matrix

type style: the j^{th} row of the matrix

۲.۲ سه نقطه ها

دستورهای `\dots`، `\vdots`، `\cdots` و `\ldots` برای قراردادن سه نقطه به صورت های مختلف به کار می روند:

مثال .

Command	type style
<code>\$a_1\ldots a_n\$</code>	$a_1 \dots a_n$
<code>\$a_1+\cdots +a_n\$</code>	$a_1 + \dots + a_n$

دستورهای `\dots` و `\ldots` به ترتیب برای تولید سه نقطه های میانی (`\cdots`) و پایینی (`\ldots`) به کار می روند. سه نقطه های میانی معمولاً بین عملگرهای دودویی و سه نقطه های پایینی بیشتر بین کاماها به کار می روند. دستورهای `\vdots` و `\ddots` نقطه هایی به صورت \cdot و $\cdot\cdot$ تولید می کنند. دستورهای `\cdots` و `\ldots` در متن راسته نیز قابل قبول هستند.

۳.۲ رادیکال‌ها

دستور $\sqrt[n]{\dots}$ ریشه n ام را تولید می‌کند. در صورتیکه $[n]$ حذف شود، جذر معمولی به نشان داده می‌شود.

مثال .

Command	type style
$\sqrt{x+y}$	$\sqrt{x+y}$
$\sqrt[3]{a}$	$\sqrt[3]{a}$
$\sqrt[n+m]{b}$	$\sqrt[n+m]{b}$
$\sqrt[q^2]{i+x}$	$\sqrt[q^2]{i+x}$

با کمک دستور \root نیز می‌توان $\sqrt[n]{\dots}$ را تولید کرد:

مثال .

Command	type style
$\root 3 \of 2$	$\sqrt[3]{2}$
$\root n \of {x^n + y^n}$	$\sqrt[n]{x^n + y^n}$

۴.۲ کسرها

برای نمایش کسر می‌توان از دستور \over استفاده کرد. استفاده از فرمان \over نیاز به دقت زیادی دارد. این فرمان تمام نمادهای قبل و بعد را به صورت کسری نمایش می‌دهد؛ استفاده نادرست از این فرمان موجب غافلگیری شما می‌شود:

مثال .

$$x = (y^2 \over k+1)$$

$$x = \frac{y^2}{k+1}$$

شیوه صحیح برای عبارت فوق به صورت زیر است:

$$x = \{y^2 \over k+1\}$$

$$x = \frac{y^2}{k+1}$$

در متن راسته برای کسر بهتر است از $\frac{\dots}{\dots}$ استفاده شود. در صورتی که کسری به صورت $\frac{\dots}{\dots}$ مورد نظر باشد، باید از دستور $\frac{\dots}{\dots}$ استفاده شود، آکولاد اول صورت کسر و آکولاد دوم مخرج کسر را در بردارد:

مثال .

Command	type style
$\frac{1+x}{y^2}$	$\frac{1+x}{y^2}$
$1+\frac{1-x}{2^x}$	$1 + \frac{1-x}{2^x}$
$\frac{x}{1+\frac{1+x}{z^2}}$	$\frac{x}{1 + \frac{1+x}{z^2}}$

تمرین ۱.

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \dots}}}}$$

۵.۲ خط بالا و پایین

دستورهای $\overline{\{...\}}$ و $\underline{\{...\}}$ برای کشیدن خط در بالا و پایین فرمول‌ها و عبارات‌ها است (این دو دستور استثنائاً در متن راسته نیز عمل می‌کنند):

مثال .

Command	type style
$\overline{x+y}$	$\overline{x+y}$
$\underline{x+y}$	$\underline{x+y}$
$\overline{\overline{x}+\underline{y}}$	$\overline{\overline{x}+\underline{y}}$
The $\underline{\text{value}}$ of x	The <u>value</u> of x

به طور مشابه، با استفاده از دستورهای $\overbrace{\{...\}}$ و $\underbrace{\{...\}}$ می‌توان در بالا یا پایین فرمول‌ها آکولاد قرار داد:

مثال .

Command	type style
$\overbrace{a+b+c}$	$\overbrace{a+b+c}$
$\overbrace{a+\underbrace{b+c}+d}$	$\overbrace{a+\underbrace{b+c}+d}$

با استفاده از بالانویس یا زیرنویس می‌توان برای آکولادها برچسب قرار داد:

مثال .

Command	type style
$\overbrace{a+\cdots+a}^{10}$	$\overbrace{a+\cdots+a}^{10}$
$\underbrace{a+\overbrace{b+\cdots+b}^m}_m$	$\underbrace{a+\overbrace{b+\cdots+b}^m}_m$

تمرین ۲.

$$\underbrace{\overbrace{a, \dots, a}^{k \text{ times}}, \overbrace{b, \dots, b}^{l \text{ times}}}_{k+l \text{ times}}$$

۶.۲ اکسانت‌ها

اکسانت‌ها علائمی هستند که بر روی حروف الفبا قرار می‌گیرند و به آنها معانی ویژه‌ای را می‌دهند. فرمانهای \vec{a} , (\bar{a}) , (\hat{a}) , (\dot{a}) , (\ddot{a}) , (\tilde{a}) , (\hat{a}) , (\tilde{a}) و ... (بقیه این اکسانت‌ها در جدول ۱ درج شده است). بعضی مواقع هنگام استفاده از اکسانت‌ها لازم است که i و j بدون نقطه باشند. برای این منظور کافی است که به جای i یا j به ترتیب از \imath یا \jmath استفاده شود که به ترتیب i و j در خروجی ظاهر می‌شود. دستورهای $\hat{}$ و $\tilde{}$ برای قراردادن \sim و $\tilde{}$ در بالای فرمول‌ها نیز به کار می‌روند. در صورتی که بخواهیم این اکسانت‌ها همه فرمول را بپوشانند از دستورهای $\widehat{}$ و $\widetilde{}$ استفاده می‌کنیم.

مثال .

Command	type style
$\widehat{x+y}$	$\widehat{x+y}$
$\widetilde{x+y}$	$\widetilde{x+y}$

همچنین می‌توان با دستور $\stackrel{\dots}{\dots}$ فرمول‌ها را بالای هم قرارداد.

مثال .

Command	type style
$A \stackrel{f}{\longrightarrow} B$	$A \xrightarrow{f} B$
$A \stackrel{\text{def}}{=} \{x \mid x > 0\}$	$A \stackrel{\text{def}}{=} \{x \mid x > 0\}$

چند فرمان دیگر

• فرمان \prime نماد " ' " را ظاهر می‌کند:

مثال .

Command	type style
$y_1 \prime$	y_1'
$y_2 \prime \prime$	y_2''
$y_3 \prime \prime \prime$	y_3'''

این امکان با به کارگیری \prime به جای \prime و $\prime \prime$ به جای $\prime \prime$ و غیره نیز انجام می‌شود:

مثال .

Command	type style
$y_1 \prime + y_2 \prime \prime - y_3 \prime \prime \prime$	$y_1' + y_2'' - y_3'''$
$y_1 \prime + y_2 \prime \prime - y_3 \prime \prime \prime$	$y_1' + y_2'' - y_3'''$

• نویسه‌های + و - و * معنای خاص خود را دارند؛ اگر علامت ضرب را در خروجی بخواهیم نویسه * این کار را ممکن نمی‌سازد:

$\$x+y*z\$$ $x + y * z$

برای این منظور از فرمان `\times` استفاده می‌شود. مانند:

$\$x+y\times z\$$ $x + y \times z$

• برای نمایش علامت تقسیم به صورت‌های زیر توجه کنید:

Command	type style
$\$a/b\$$	a/b
$\$a:b\$$	$a : b$
$\$a\div b\$$	$a \div b$
$\$a\over b\$$	$\frac{a}{b}$

یادآوری می‌کنیم که { } برای دسته‌بندی استفاده می‌شود:

مثال .

Command	type style
$\$7\{1\over 2\}\$$	$7\frac{1}{2}$
$\$e^{i\pi\over 2}=i\$$	$e^{\frac{i\pi}{2}} = i$

• فرمان `\atop` مشابه `\over` کار می‌کند اما خط کسری را نمی‌کشد:

مثال .

$\$x \atop y+2\$$

$$\frac{x}{y+2}$$

• استفاده از فرمانهای `\left` و `\right` قبل از کاراکترهایی مانند () یا [] و هجده علامت دیگر که در زیر آن‌ها را فهرست کرده‌ایم، باعث می‌شود تا در فرمول‌های ریاضی این کارکترها با اندازه مناسب در فرمول قرار گیرند:

{ } [] [] < > / \ | || ↑ ⇑ ↓ ⇓ ⇔ ⇕

برای هر فرمان `\left` (یا `\right`) باید یک `\right` (یا `\left`) آمده باشد. اگر نیاز به فرمان دوم نباشد، در آن صورت به جای آن از `\right.` (یا `\left.`) استفاده می‌کنیم. مثال زیر صرفاً با امکاناتی که تاکنون معرفی شده نوشته شده است:

مثال .

$\$f(x) = \left\{ \begin{array}{l} \frac{1}{q} \\ 0 \end{array} \right. \atop 0 \hspace{1cm} \right. \right.$

$$f(x) = \begin{cases} \frac{1}{q} & x = \frac{p}{q} \\ 0 & \text{otherwise} \end{cases}$$

چنانچه بخواهیم اندازه قلم را به انتخاب خود تعیین کنیم از فرمان `\big` یا `\Big` یا `\bigg` یا `\Bigg` برای چهار مرحله بزرگ کردن قلم استفاده می‌کنیم:

`$$\Bigg(\bigg(\Big(\big(() \big) \Big) \bigg) \Bigg)$$`

$$\left(\left(\left(\left(\left(\right)\right)\right)\right)\right)$$

مثال .

`$$\bigg({n \atop k} \bigg)$$`

$$\binom{n}{k}$$

البته برای نماد «انتخاب k عضو از n عضو» فرمان `\choose` مناسب‌تر از مثال فوق است:

`$$n \choose k$$`

$$\binom{n}{k}$$

برای استفاده مرکب از فرمان‌های `\over`، `\atop` و `\choose` باید با کمک آکولاد دسته‌بندی مناسب را ایجاد کرد؛ برای مثال `$$n \choose k \over 2$$` خطاست:

مثال .

Command	type style
<code>\$\$n \choose k \over 2\$\$</code>	$\frac{\binom{n}{k}}{2}$
<code>\$\$n \choose {k \over 2}\$\$</code>	$\binom{n}{\frac{k}{2}}$

البته بهتر است عبارت اخیر به یکی از دو صورت زیر نمایش داده شود:

Command	type style
<code>\$\$n \choose k/2\$\$</code>	$\binom{n}{k/2}$
<code>\$\$n \choose {1 \over 2}k\$\$</code>	$\binom{n}{\frac{1}{2}k}$

۷.۲ حروف تحریری

با استفاده از دستور `\cal` می‌توان ۲۶ حرف بزرگ تحریری را تولید کرد.

مثال .

`$$\{\cal B\}=\cal C\cup \{\cal E\}$$`

$$B = C \cup E$$

۸.۲ نفی رابطه‌ها

برای به دست آوردن نفی رابطه‌هایی که در جدول ۵ آمده‌اند کافی است دستور `\not` را بر دستور تولید آن رابطه‌ها مقدم کنیم:

مثال .

Command: `\not \in A, \ B \not \subseteq X, \ a \not \equiv b`

type style: `\notin A, B \not\subseteq X, a \not\equiv b`

۹.۲ نمادهای \int و \sum

دستورهای `\int` و `\sum` به ترتیب نمادهای جمع بندی \sum و انتگرال \int را تولید می‌کنند. با استفاده از دستورهای تولید زیرنویس و بالانویس می‌توان حدود بالا و پایین این نمادها را معرفی کرد:

مثال .

Command	type style
<code>\sum_{i=1}^k a_i</code>	$\sum_{i=1}^k a_i$
<code>\int_0^1 \! f(x) \, dx</code>	$\int_0^1 f(x) dx$

درحالت ریاضی تمام فاصله‌های وارد شده نادیده گرفته می‌شوند و فواصل به طور اتوماتیک تنظیم می‌شوند تا بهترین حالت به دست آید. اما در برخی موارد ممکن است که فاصله‌های کوچکی لازم باشند تا خروجی واضح‌تر شود. به عنوان مثال در دستور انتگرال فوق دو دستور جدید به کار رفته‌است: `\,` و `\!`؛ دستور `\,` برای ایجاد فاصله‌ای کوچک بین $f(x)$ و dx به کار رفته است. این فاصله حدود یک چهارم فاصله‌ای است که دستور `\quad` تولید می‌کند. دستور `\!` برای ایجاد «فاصله منفی» است زیرا در حالت معمولی دستور `\int` فاصله‌ای بعد از نماد \int به وجود می‌آورد و دستور `\!` برای حذف این فاصله استفاده شده است. فرمان‌های ایجاد فواصل جزئی عبارتند از:

command	space length
<code>\!</code>	
<code>\/</code>	
<code>\,</code>	
<code>\:</code>	
<code>\;</code>	
<code>\quad</code>	
<code>\quad</code>	
<code>\quad</code>	

فرمان `\quad` و `\quad` نیز فاصله‌هایی متناسب با اندازه قلم ایجاد می‌کنند.

مثال .

Command:

the summation $\sum_{i=1}^k a_i$ and the integral

$\int_0^1 f(x) dx$

type style:

the summation $\sum_{i=1}^k a_i$ and the integral $\int_0^1 f(x) dx$

همه نمادهایی که در جدول ۱۱ آمده‌اند از این قاعده پیروی می‌کنند. وقتی در سبک داخلی قرار داریم، حدود بالا و پایین این نمادها در سمت راست آن‌ها قرار می‌گیرند. اما در سبک نمایشی (در وسط سطر بعدی) حدود در بالا و پایین نماد قرار می‌گیرند. اگر بخواهیم حدود در وسط متن نیز مثل سبک نمایشی ظاهر شوند باید از دستور \displaystyle استفاده کنیم.

مثال .

Command:

the summation $\displaystyle \sum_{i=1}^k a_i$ and the

integral $\displaystyle \int_0^1 f(x) dx$

type style:

the summation $\sum_{i=1}^k a_i$ and the integral $\int_0^1 f(x) dx$

۱۰.۲ شماره گذاری فرمول‌ها

در \LaTeX با استفاده از دستور زیر می‌توان به سادگی فرمول‌هایی را که در سطرهای جداگانه می‌آیند شماره گذاری کرد.

$\begin{equation}$

:

$\end{equation}$

آنچه که در این دستور ظاهر می‌شود متن ریاضی محسوب می‌شود و لزومی به استفاده از دستورهای $\$$ یا \backslash نیست.

مثال .

$\begin{equation}$

$a=b$

$\end{equation}$

$\begin{equation}$

$a+c=b+d$

$\end{equation}$

and the relation

$\begin{equation}$

$e=f$

`\end{equation}`

$$a = b \tag{۱}$$

$$a + c = b + d \tag{۲}$$

and the relation

$$e = f \tag{۳}$$

دقت کنید که شماره‌ها را خود برنامهٔ L^AT_EX تولید می‌کند. شمارهٔ معادله در سمت راست فرمول چیده می‌شود و بسته به اینکه از سبک مقاله یا کتاب استفاده شود به صورت یک یا ترکیب دو شماره نوشته می‌شود.

مثال .

`\begin{equation}`

$$x_i^{2n} + y_j^{2m} = z_{ij}^p$$

`\end{equation}`

$$x_i^{2n} + y_j^{2m} = z_{ij}^p \tag{۲.۶}$$

برنامهٔ L^AT_EX معادله‌ها را به ترتیب صعودی شماره گذاری می‌کند. در صورتیکه بخواهیم این ترتیب را برهم زنیم مثلاً بخواهیم بعد از فرمول (۴) فرمول بعدی با شمارهٔ (۶) ظاهر شود چنین عمل می‌کنیم:

`\begin{equation}`

$$x=y^2$$

`\end{equation}`

`\setcounter{equation}{5}`

`\begin{equation}`

$$z=x+1$$

`\end{equation}`

$$x = y^2 \tag{۴}$$

$$z = x + 1 \tag{۶}$$

۱۱.۲ نمادها و توابع ریاضی

در جدول‌های ۳ تا ۱۲ دستورهای تولید نمادهای مختلف ریاضی، حروف یونانی و نیز اسامی خاص و برخی توابع متداول ریاضی آمده است.

مثال .

$$\sin^2 x + \cos^2 x = 1$$

$$\lim_{n \rightarrow \infty} \frac{n}{n+1}$$

$$\int_0^{+\infty} f(x) dx$$

$$A \otimes B = C \cap \{x \mid x \sqcup y \sqsubseteq z\} = \emptyset$$

$$\sin^2 x + \cos^2 x = 1$$

$$\lim_{n \rightarrow \infty} \frac{n}{n+1}$$

$$\int_0^{+\infty} f(x) dx$$

$$A \otimes B = C \cap \{x \mid x \sqcup y \sqsubseteq z\} = \emptyset$$

۱۲.۲ ماتریس‌ها

دستور $\begin{array}{...} \dots \end{array}$ برای تولید ماتریس‌ها به کار می‌رود. این دستورها در متن ریاضی قابل استفاده است. اگر ماتریس n ستون داشته باشد دستور این چنین شروع می‌شود:

$$\begin{array} \{x_1 \ x_2 \dots x_n\}$$

که در آن هر x_i یکی از نمادهای c, d, r یا r, c, d است (به ترتیب به معنای «چپ»، «وسط» و «راست»). پارامتر x_i نشان می‌دهد که درایه‌های (entry) ستون i ام باید در سمت چپ، وسط یا سمت راست فضائی که برای این ستون در نظر گرفته شده است ظاهر شوند. درایه‌های یک سطر را با دستور $\&t$ و سطرهای متوالی را با دستور $\&A$ از هم جدا می‌کنیم:

مثال .

$$\begin{array} \{cc\}$$

$$a \ \& \ b \ \backslash \backslash$$

$$c \ \& \ d$$

$$\end{array}$$

$$a \quad b$$

$$c \quad d$$

دقت کنید که پس از سطر آخر نیازی به دستور $\&A$ نیست. در صورتی که بخواهیم در طرفین این ماتریس پرانتز، خط راست، کروشه و ... قرار دهیم از دستورهایی $\&left$ و $\&right$ استفاده می‌کنیم.

مثال .

$$\left(\begin{array}{cc} a & b+f \\ c+d & e \end{array} \right)$$

$$\left(\begin{array}{cc} a & b+f \\ c+d & e \end{array} \right)$$

$$\left| \begin{array}{cc} a & b+f \\ c+d & e \end{array} \right|$$

$$\left(\begin{array}{cc} a & b+f \\ c+d & e \end{array} \right)$$

همان طور که در مثال آخر دیده می شود لزومی ندارد که در سمت راست و چپ یک نوع پرانتز آورده شود؛ ولی همواره `\left` و `\right` باید هر دو ظاهر شوند، در صورتیکه بخواهیم تنها از یکی استفاده کنیم در انتها از دستور `\left` یا `\right` استفاده می کنیم:

مثال .

$$f(x) = \begin{cases} x & \text{if } x < 0, \\ \left| \begin{array}{cc} a & b \\ c & d \end{array} \right| & \text{if } 0 \leq x < 1, \\ x^2 & \text{if } x \leq 1. \end{cases}$$

همان طور که می بینید درایه های یک ماتریس می توانند هر مجموعه مجازی از کاراکترها (خواه از نوع ریاضی یا از نوع متن راسته) باشند.

مثال .

$$(x+y)^n = \sum_{i=0}^n \left(\begin{array}{c} n \\ i \end{array} \right) x^i y^{n-i}$$

$$(x+y)^n = \sum_{i=0}^n \binom{n}{i} x^i y^{n-i}$$

تمرین ۳

$$\begin{aligned} a &= b \\ b + c &= e + f \end{aligned}$$

تمرین ۴

$$\sum_{\substack{i \in A \\ i \neq \circ}} a_i = b$$

تمرین ۵

$$\begin{aligned} a + b &= ۱, & c + d &= ۲, & e + f &= ۰, \\ n &= ۵, & i + j + l &= -۱, & m &= \sqrt{۲}. \end{aligned}$$

تمرین ۶

$$\begin{array}{ccccccc} & & & O & & & \\ & & & \downarrow & & & \\ O & \longrightarrow & B & \xrightarrow{h} & C & \longrightarrow & D \longrightarrow O \\ & & \downarrow f & & \downarrow \phi & & \\ O & \longrightarrow & E & \longrightarrow & F & \longrightarrow & G \longrightarrow O \\ & & & & \downarrow & & \\ & & & & O & & \end{array}$$

برای کشیدن سریع ماتریس‌ها از دستور $\pmatrix{\dots}$ نیز می‌توان استفاده کرد. برای جدا کردن درایه‌های یک سطر از شناسه & و برای جدا کردن یک سطر از سطر بعد از دستور \cr استفاده می‌کنیم.

مثال .

$$\pmatrix{a & b \cr c & d} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

هم‌چنین دستور $\bordermatrix{\dots}$ ماتریس‌های با حاشیه ایجاد می‌کند:

مثال .

$$\bordermatrix{y_1 & y_2 \cr x_1 & a & b \cr x_2 & c & d}}$$

$$\begin{matrix} y_1 & y_2 \\ x_1 & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\ x_2 & \end{matrix}$$

۱۳.۲ جدول‌ها

دستور `\begin{tabular}{...} ... \end{tabular}` برای تولید جدول‌ها به کار می‌رود. قواعد این دستور بسیار شبیه قواعد تولید ماتریس‌ها است. دو تفاوت عمده وجود دارد؛ یکی آنکه این دستور هم در متن راسته و هم در متن ریاضی به کار می‌رود و دیگر آن که در آن می‌توان خط‌های افقی و عمودی رسم کرد. در صورتی که بخواهیم درایه‌ای ریاضی در جدول قرار دهیم باید از یک جفت $\$$ استفاده کنیم. یک جدول n ستونی با دستور `\begin{tabular}{x_1 x_2 \dots x_n}` شروع می‌شود که $x_i \in \{l, c, r\}$. در صورتی که بخواهیم بین ستون‌های i و $i+1$ یک خط عمودی قرار دهیم، در دستور فوق بین x_i و x_{i+1} علامت `|` را قرار می‌دهیم. دستور `\hline` قبل یا بعد از هر سطر یک خط افقی رسم می‌کند.

مثال .

```
\begin{tabular}{|cc|c|} \hline
```

```
name & a & 0 \\
```

```
1 & & $x^2$ \\ \hline
```

```
x & y & z \\ \hline
```

```
\end{tabular}
```

name	a	0
1		x^2
x	y	z

دستور `\cline{i-j}` خط افقی رسم می‌کند که از ستون i شروع و به ستون j ختم می‌شود.

مثال .

```
\begin{tabular}{||l|c|c||} \hline
```

```
abc & b & c \\ \cline{2-3}
```

```
& xy & c \\ \hline
```

```
zz & g & 1 \\ \hline
```

```
\end{tabular}
```

abc	b	c
	xy	c
zz	g	1

تمرین ۷

$$\left(\begin{array}{c|c|c|c} x_1 & & & \\ x_2 & & & \\ \vdots & A & \dots & \circ \\ x_n & & & \\ \hline y & \circ & A & \vdots \\ \hline & \circ & \dots & \circ \end{array} \right)$$

به جای نویسه‌های c ، d ، r می‌توانیم از $\{اندازه\} p$ استفاده کنیم که در آن صورت هر قسمت از ستون متناظر، در پاراگرافی با طول سطر مشخص شده حروفچینی می‌شود. اگر بخواهیم بین هر دو ستون نوشته‌ای را قرار دهیم بین نویسه‌های معرف آن دو ستون، $\@$ را قرار می‌دهیم تا نوشته داخل آکولادها بین دو ستون قرار بگیرد.

دستور مفید دیگر در تولید جدول‌ها $\multicolumn{n}{pos}{text}$ است. این دستور برای قرار دادن یک درایه در یک یا چند ستون به کار می‌رود. آرگومان اول، n ، تعداد ستون‌هایی است که درایه در آن قرار می‌گیرد. آرگومان دوم تعیین کننده محل قرار گرفتن درایه است، $pos \in \{l, c, r\}$. آرگومان سوم متنی است که در این ستون‌ها قرار می‌گیرد.

مثال .

```
\begin{tabular}{|c|c|c|} \hline
name & a & c \\ \hline
\multicolumn{2}{|c|}{x} & z \\ \hline
\end{tabular}
```

name	a	c
x		z

۱۴.۲ مراجعه متقابل (cross reference)

منظور از مراجعه متقابل مراجعه از قسمتی از متن به قسمت دیگر آن است. در واقع علت شماره‌گذاری شکل‌ها، معادلات و غیره مراجعه خواننده به آن‌ها است. به عنوان مثال در یک قسمت از یک متن این جمله به چشم می‌خورد. در \LaTeX برچسب‌گذاری و رجوع به بخش‌ها، شکل‌ها، جدول‌ها، معادلات و غیره به آسانی از طریق دستورات زیر صورت می‌گیرد.

- $\text{\label{key}}$: این دستور برای برچسب‌گذاری یک شکل، قضیه و یا یک قطعه از متن به کار می‌رود. هنگامی که این دستور در یک قسمت متن ظاهر می‌شود، \LaTeX شماره محیط جاری را به key اختصاص می‌دهد. key می‌تواند هر دنباله‌ای از حروف، ارقام و یا کاراکترهای نقطه‌گذاری باشد. ذکر این نکته لازم است که حروف کوچک و بزرگ متفاوت هستند.

- $\text{\ref{key}}$: این دستور برای رجوع به شکل، قضیه، معادله و یا قطعه مورد نظر از متن به کار می‌رود. این دستور شماره قسمت مورد نظر را مطابق دستور \label تولید می‌کند. به عنوان مثال اگر در متنی عبارت “see (1)” برای رجوع به معادله شماره ۱ باشد، آن را به صورت زیر تایپ می‌کنیم:

```
see~(\ref{E: firstint})
```

که در آن علامت \sim باعث می‌شود تا شماره ۱ از کلمه see جدا نشود. $E: firstint$ برچسب معادله شماره ۱ است که با دستور \label به آن اختصاص یافته است.

فصل ۳

محیط‌ها

یکی دیگر از امکانات مهم \LaTeX محیط‌ها (environments) هستند. محیط که با فرمانهای $\{$ نام محیط $\}$ و $\backslash\text{end}\{$ نام محیط $\}$ در ابتدا و انتهایش مشخص می‌شود، محدوده‌ای است که در داخل آن اعمال خاصی بسته به نوع محیط انجام می‌شوند. یکی از این محیط‌ها، محیط `enumerate` یا شمارش است. کاربرد این محیط در مواقعی است که بخواهیم مطالبی را به ترتیب شماره‌گذاری کنیم. برای این کار در ابتدا و انتهای قسمتی که می‌خواهیم شماره‌گذاری شود فرمانهای $\backslash\text{begin}\{enumerate\}$ و $\backslash\text{end}\{enumerate\}$ را قرار می‌دهیم و سپس در جلوی هر فقره، فرمان $\backslash\text{item}$ را می‌نویسیم.

مثال .

```
\begin{enumerate}
\item First item
\item Second item
\item Third item
\end{enumerate}
```

1. First item
2. Second item
3. Third item

با توجه به توضیحات ذکر شده واضح است که `document` نیز نوعی محیط است که از آن برای تعیین محدوده متن اصلی استفاده می‌شود.

اگر بخواهیم قسمتی از متن با ترتیبی غیر از روال عادی حروفچینی شود، مثلاً بخواهیم جمله‌های یک متن در وسط سطر قرار بگیرد، یا اصطلاحاً وسط چین شود، محیط `center` به کار برده می‌شود. بدین ترتیب که در ابتدا و انتهای متنی که می‌خواهیم وسط چین شود به ترتیب فرمانهای $\backslash\text{begin}\{center\}$ و $\backslash\text{end}\{center\}$ را قرار می‌دهیم.

مثال .

```
\begin{center}
In the center of line
\end{center}
```

In the center of line

در صورتی که بخواهیم بیش از یک سطر را وسط چین کنیم، در انتهای هر سطر باید فرمان `\` را قرار دهیم؛ با این کار بقیه متن در سطر بعدی و به طور وسط چین حروفچینی می‌شود.

مثال .

```
\begin{center}
This is two lines in the\
center
\end{center}
```

This is two lines in the
center

به طور مشابه برای راست چین کردن و چپ چین کردن متن به ترتیب از محیط‌های `flushright` و `flushleft` استفاده می‌کنیم.

۱.۳ محیط `verbatim`

در مواردی لازم است که قسمتی از متن به همان صورتی که در پرونده ورودی وجود دارد حروفچینی شود؛ مثلاً ممکن است بخواهیم در بین متن اصلی یک برنامه کامپیوتری گنجانده شود. در این مورد از محیط `verbatim` استفاده می‌کنیم.

مثال .

```
\begin{verbatim}
functin factorial(n: Integer): Integer;
begin
  if n = 0 then factorial :=1
  else factorial :=factorial(n-1) * n;
end;
\end{verbatim}
```

در خروجی، متن داخل این محیط با قلم `\tt` نوشته می‌شود تا از متن معمولی تمییز داده شود (مشابه مثالهای ذکر شده در این جزوه). فرمان `+` متن `\verb+` نیز مانند محیط `verbatim` عمل می‌کند با این تفاوت که در بین متن اصلی نیز می‌توان از آن استفاده کرد. ضمناً به جای نویسه `+` می‌توان از نویسه‌های غیر حرفی دیگر هم استفاده کرد.

مثال .

For generating `\LaTeX` logo use `\verb+\LaTeX+` command.

For generating `LATEX` logo use `\LaTeX` command.

۲.۳ محیط tabbing

در قسمتهای قبل دیدیم که با استفاده از محیط tabular می‌توان جدول‌های مختلف را آماده کرد. با استفاده از محیط tabbing نیز می‌توان انواع دیگری از جدول‌ها، مانند برنامه‌های کامپیوتری، متن‌هایی با تورفتگیهای متفاوت، اشعار و غیره را تهیه کرد. عملکرد این محیط تا حدودی شبیه به کلیدهای جدول بندی ماشین تحریر است. برای این منظور محل ستونها را به وسیله `=` مشخص می‌کنیم و در سطرهای بعدی با کمک `>` بقیه مطالب سطر از ابتدای ستون بعدی حروفچینی می‌شوند. فرمان `\` نیز به همان معنی قبل برای مشخص کردن انتهای سطر به کار می‌رود.

مثال .

```
\begin{tabbing}
fu\nctin factor(n: Integer): Integer;\
\> be\=gin\
\>\> if n = 0 then factor :=1\
\>\> else factor :=factor(n-1) * n;\
\> end;
\end{tabbing}
```

```
functin factor(n: Integer): Integer;
begin
  if n = 0 then factor :=1
  else factor :=factor(n-1) * n;
end;
```

فرمان‌های دیگر این محیط عبارتند از:

- + : پس از این فرمان سطرهای بعدی از یک ستون جلوتر آغاز می‌شوند.
 - : شروع سطرهای بعدی از یک ستون عقب تر خواهد بود. (این فرمان خنثی کننده فرمان `+` است.)
 - < : در ابتدای سطر می‌آید و سطر جاری از یک ستون عقب تر شروع می‌شود. (البته قبلاً باید از فرمان `+` استفاده شده باشد.)
 - ' : کلیه مطالب پیش از این فرمان درستون جاری، در سمت چپ ستون می‌آید و مطالب پس از آن از ابتدای ستون ادامه می‌یابد.
 - ^ : متن پس از این فرمان در انتهای راست صفحه چاپ می‌شود.
- توجه کنید که می‌توان ستون‌ها را در یک سطر نمونه معین کنید و در انتهای سطر فرمان `\kill` را قرار دهید تا آن سطر حروفچینی نشود.

۳.۳ محیط‌های array و eqnarray

محیط array برای تولید آرایه‌ها و اجزاء جدولی در متن ریاضی طراحی شده است. از این محیط می‌توان برای چیدن ماتریس‌ها، فرمول‌های چند سطری، توابع چند ضابطه‌ای و غیره استفاده کرد. روش به کارگیری این فرمان همانند محیط tabular است که در قسمت‌های قبل به آن اشاره شد.

مثال .

```

\left[\begin{array}{cccc}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{array}\right]_{m \times n}

```

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

در مثال بعد کاربرد این محیط را برای تولید توابع چند ضابطه‌ای نشان می‌دهیم. فاصله‌هایی که در ورودی رعایت شده است به منظور خوانایی بیشتر است و در خروجی اثری ندارد. می‌توان این متن را بدون رعایت فاصله و پشت سر هم تایپ کرد و نتیجه در خروجی تغییر نخواهد کرد.

مثال .

```

z = \left\{ \begin{array}{l}
re^{i\theta} & 0 < \theta < \pi \\
o & -\pi < \theta < 0
\end{array} \right.

```

$$z = \begin{cases} re^{i\theta} & 0 < \theta < \pi \\ o & -\pi < \theta < 0 \end{cases}$$

محیط `eqnarray` یکی دیگر از محیط‌های حالت ریاضی است. این محیط تلفیقی از محیط‌های `equation` و `array` با سه ستون است که در انتهای هر سطر آن شماره معادله قرار می‌گیرد. کاربرد این محیط زمانی است که بخواهیم چند معادله یا فرمول متوالی نسبت به هم تنظیم شوند. اگر بخواهیم یک سطر شماره نداشته باشد، از فرمان `\nonumber` در آن سطر استفاده می‌کنیم.

مثال .

```

\begin{eqnarray}
x & = & 5k \\
y & < & a + b + c + d + \nonumber \\
& & e + f + g + h + \dots
\end{eqnarray}

```

$$x = 5k \quad (۱)$$

$$y < a + b + c + d + e + f + g + h + \dots \quad (۲)$$

۴.۳ محیط picture

این محیط با اختصاص کادری فرضی امکان ترسیم خط، پیکان، دایره و دیگر اشکال را در داخل آن فراهم می‌کند. قبل از شروع این محیط می‌توان واحد طول را با کمک فرمان `\unitlength` تعریف کرد. در صورتی که این واحد تعریف نشود، واحد طول به طور پیش فرض 1pt انتخاب می‌شود. (هر پوینت در حدود ۰.۳۵ میلی متر است.)

مثال .

```
\unitlength=1cm
\begin{picture}(x,y)(a,b)
:
\end{picture}
```

در این فرمان x و y به ترتیب اندازه کادر مفروض در جهت محور x ها و محور y ها است و (a,b) مختصات گوشه سمت چپ پایین کادر است. در صورتی که این نقطه انتخاب نشود، $(0,0)$ در نظر گرفته می‌شود. بعد از تعریف این محیط، می‌توان با کمک فرمان زیر، از دیگر فرمان‌های ترسیمی در محل تعیین شده استفاده کرد:

```
\put(x,y){picture-object}
```

در این صورت شکل مورد نظر در نقطه مرجع با مختصات (x,y) ترسیم می‌شود. اگر بخواهیم شکلی را در چند نقطه واقع بر یک امتداد ترسیم کنیم، می‌توانیم از فرمان زیر استفاده کنیم:

```
\multiput(x,y)(dx,dy){n}{picture-object}
```

این فرمان معادل n بار ترسیم شکل است که شکل i ام در نقطه مرجعی با مختصات $(x+(i-1)dx, y+(i-1)dy)$ ترسیم شده است. فرمان‌های زیر برای ترسیم کادر استفاده می‌شوند:

```
\makebox(x,y)[pos]{text}
```

```
\framebox(x,y)[pos]{text}
```

```
\dashbox(x,y)[pos]{text}
```

هر یک از فرمانهای فوق می‌تواند یک کادر با طول و عرض x و y و محتوی متن `text` ایجاد کند. در فرمان `\makebox` کادر ترسیم نمی‌شود ولی در `\framebox` این کادر ترسیم می‌شود. فرمان `\dashbox` برای ترسیم کادر با خط چین است. گزینه اختیاری `[pos]` برای تعیین موقعیت `text` نسبت به کادر استفاده می‌شود. این موقعیت می‌تواند `t`، `r`، `l`، `b` و `c` به ترتیب معرف ضلع چپ، راست، بالا، پایین و وسط کادر باشد. در حالت عادی این

موقعیت وسط کادر است. بدین ترتیب موقعیت `tr` باعث می‌شود که `text` در گوشهٔ راست و بالای کادر قرار گیرد. در این سه فرمان نقطهٔ مرجع، گوشهٔ سمت چپ پایین کادر است. نوع دیگر فرمان‌های ترسیم شکل به صورت زیر هستند:

```
\line(h,v){len}
\vector(h,v){len}
```

که به ترتیب برای ترسیم خط و پیکان با شیب v/h به کار می‌روند. مقدار h و v می‌تواند با فاصله‌های یک واحدی، از -6 تا 6 برای `\line` و از -4 تا 4 برای `\vector` تغییر کند و طول خط یا پیکان در امتداد محور x به اندازهٔ `len` است، مگر اینکه h برابر با صفر باشد که در این صورت این طول در امتداد محور y فرض می‌شود. فرمان‌های `\circle{diam}` و `\circle*{diam}` به ترتیب برای ترسیم دایرهٔ توخالی و توپر به کار می‌روند که قطر آن برابر با `diam` است. این قطر برای `\circle` حداکثر برابر با 40° پوینت (حدود ۱۴ میلی متر) و برای `\circle*` حداکثر برابر با ۱۵ پوینت (حدود ۵.۲۵ میلی متر) است.

فرمان `\oval(x,y)[part]` نیز برای ترسیم کادری با گوشه‌های ربع دایره است. در این فرمان x و y به ترتیب طول و عرض این کادر را تعیین می‌کند و نقطهٔ مرجع، مرکز کادر است. گزینهٔ اختیاری `part` می‌تواند یکی یا دو تا از حروف زیر باشد، که باعث می‌شود تا یا نصف کادر کشیده شود: `l` چپ، `r` راست، `t` بالا و `b` پایین کادر را ترسیم می‌کند. حالت پیش فرض، ترسیم کامل این کادر است.

فرمان `\frame{picture-object}` برای ترسیم کادری بدون فاصله دور شکل یا متن معرفی شده به کار می‌رود.

دیگر فرمان این محیط `\shortstack[pos]{rows}` است که معادل با فرمان‌های زیر عمل می‌کند:

```
\begin{tabular}[b]{pos}
rows
\end{tabular}
```

علاوه بر این‌ها می‌توان با کمک فرمان‌های `\thinlines` و `\thicklines` خطوط، دایره‌ها و دیگر اشکال را به ترتیب نازک یا ضخیم ترسیم کرد. در این محیط در حالت پیش فرض فرمان `\thinlines` فعال است. همچنین می‌توان با فرمان `\linethickness{len}` ضخامت خطوط افقی و عمودی را به اندازهٔ `len` تعیین کرد. این فرمان روی خطوط مایل، دایره‌ها و ربع دایره‌ها در گوشه‌های `\oval` تأثیر ندارد.

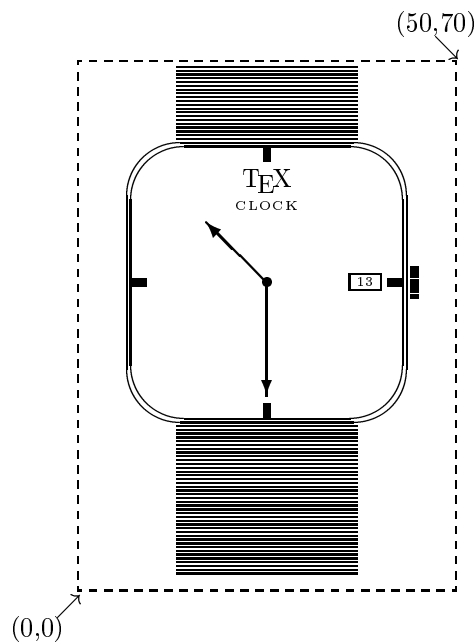
مثال .

```
\unitlength=1mm
\begin{picture}(50,70)(0,0)
\thinlines
\put(25,40){\oval(36,36)}
\put(25,40){\oval(37,37)}
\multiput(13,21)(0,-0.5){40}{\line(1,0){24}}
\multiput(13,59)(0,0.5){20}{\line(1,0){24}}
\multiput(44,38)(0,0.4){11}{\line(1,0){1}}
\put(36,39){\framebox(4,2){\tiny13}}
\thicklines
```

```

\put(25,40){\vector(-1,1){8}}
\put(25,40){\vector(0,-1){15}}
\put(25,40){\circle*{1.5}}
\linethickness{1mm}
\put(7,40){\line(1,0){2}}
\put(43,40){\line(-1,0){2}}
\put(25,22){\line(0,1){2}}
\put(25,58){\line(0,-1){2}}
\put(25,50){\makebox(0,5)[t]{\shortstack[c]{\TeX \\ \tiny CLOCK}}}
\end{picture}

```



۵.۳ محیط theorem

در صورتی که بخواهیم از قضیه، مثال، گزاره و ... در متن استفاده کنیم ابتدا باید محیط جدید را در قسمت پیش درآمد با یکی از دو دستور زیر تعریف کنیم.

```
\newtheorem{env-name}{caption}[within]
```

```
\newtheorem{env-name}[numbered-like]{caption}
```

- `env-name`: نام محیط تعریف شده، که به صورت رشته‌ای از حروف می‌باشد (این نام نباید نام محیط یا شماره از قبل تعریف شده باشد).
- `caption`: عبارتی که در ابتدای محیط درست قبل از شماره چاپ می‌شود. به عنوان مثال این متن می‌تواند به صورت "Theorem" یا "Lemma" باشد.
- `within`: نام یک شماره از پیش تعریف شده که معمولاً مربوط به یک قسمت بخش است.

روش‌هایی را برای تنظیم شمارندهٔ قضیهٔ جدید (یا مثال، گزاره و ...) در یک قسمت بخش ایجاد می‌کند.

- `numbered-like`: نام یک محیط قضیه مانند که قبلاً تعریف شده است. در صورت استفاده از این دستور، شماره‌ها از ادامه شماره‌های محیط قضیه قبلی شروع می‌شود.

بعد از تعریف این دستور در قسمت پیش درآمد کافی است در قسمتی از متن که می‌خواهیم از قضیه (مثال، گزاره و ...) استفاده کنیم، متن قضیه را در محیط زیر قرار دهیم:

```
\begin{env-name}
:
\end{env-name}
```

مثال.

با اضافه کردن دستور `[chapter]{قضیه}` `\newtheorem{thm}` در پیش درآمد، می‌توانیم در داخل متن با دستور زیر قضیه‌ها را با در نظر گرفتن شمارهٔ فصل، شماره‌گذاری نماییم.

```
\begin{thm}
:
\end{thm}
```

همچنین با تعریف دستور `[thm]{مثال}` `\newtheorem{exm}` و دستور `[thm]{گزاره}` `\newtheorem{prop}` در قسمت پیش درآمد می‌توان از مثال یا گزاره نیز در متن استفاده کرد. با این دستورها قضایا، مثال‌ها و گزاره‌ها با یک شمارنده شماره‌گذاری می‌شوند.

۶.۳ عناصر شناور

عناصر شناور همان‌طور که از نامشان پیداست، عناصری هستند که محل حروفچینی‌شان ممکن است تغییر کند و در جای دیگری به غیر از محلی که در پروندهٔ ورودی قرار گرفته‌اند حروفچینی شوند. به عنوان مثال، فرض کنید که در بین متن شکلی قرار گرفته باشد که هنگام صفحه‌بندی، در انتهای صفحه جای خالی مناسب برای آن وجود نداشته باشد و شکل در ابتدای صفحهٔ بعد بیاید، در این صورت در صفحهٔ قبلی فضایی خالی می‌ماند. عناصر شناور برای رفع این مشکلات به وجود آمده‌اند.

در صورت استفاده از این عناصر، \LaTeX بهترین محل ممکن را برای این عناصر پیدا می‌کند. محیط‌های `table` و `figure` دو نمونه از این عناصر هستند که به ترتیب برای تعیین محل شکل‌ها و جدول‌ها به کار می‌روند. در داخل این دو محیط اغلب از محیط‌های `picture` (محل ترسیم شکل) و `tabular` استفاده می‌شود. همچنین در داخل این محیط‌ها می‌توان با کمک فرمان‌های `\vspace{اندازه}` و `\vspace*{اندازه}` فضای خالی مناسب را برای چسباندن شکل‌ها و جدول‌های جداگانه در نظر گرفت.

بعد از فرمان‌های `\begin{table}` و `\begin{figure}` می‌توان در داخل یک جفت کروشه با کمک یک تا چهار نویسه، محل‌های دلخواه را به ترتیب اولویت برای این عناصر مشخص کرد. این نویسه‌ها عبارتند از: `h` برای محل جاری، `t` برای بالای صفحه، `b` برای پایین صفحه و `p` برای درج در یک صفحهٔ جداگانهٔ مخصوص شناورها. در حالتی که اولویت مشخص نشود، \LaTeX ترتیب `tbp` (از چپ به راست) را در نظر می‌گیرد. به این معنی که محل‌های مورد نظر برای شناور، به ترتیب اولویت عبارت است از: بالای صفحه، پایین صفحه و صفحهٔ جداگانهٔ شناورها.

در داخل این دو محیط می‌توان به وسیلهٔ فرمان `\caption` توضیح شکل (یا جدول) را در پایین آن قرارداد. این توضیح پس از این فرمان در داخل یک جفت آکولاد قرارداد می‌شود و در هنگام حروفچینی به همراه شمارهٔ شکل (یا جدول) در زیر آن قرار می‌گیرد.

مثال .

```
\begin{table}[h]
\begin{tabular}{c c|c}


$p$ & $q$ & $p \wedge q$ \\
\hline
T & T & T \\
T & F & F \\
F & T & F \\
F & F & F \\
\end{tabular}
\caption{conjunction}
\end{table}


```

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

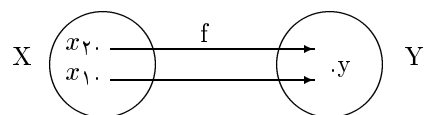
جدول ۶.۳ : conjunction

```
\begin{figure}[h]
\unitlength=1mm
\begin{picture}(70,70)(0,0)
\put(35,35){\circle{14}}
\put(65,35){\circle{14}}
\put(23,35){X}
\put(75,35){Y}
\put(36,33){\vector(1,0){27}}
\put(48,38){f}
\put(36,37){\vector(1,0){27}}
\put(30,33){$x_1$.}
\put(30,37){$x_2$.}
\put(65,34){.y}
\end{picture}
\end{figure}
```

```

\end{picture}
\caption{$x_1$ \& $x_2$ are y's preimages}
\end{figure}

```



شکل ۱.۳: x_1 & x_2 are y's preimages

۷.۳ محیط thebibliography

برای ایجاد کتاب‌نامه یا فهرست مراجع در کتاب، مقاله، گزارش و ... از محیط thebibliography استفاده می‌شود. این محیط به صورت زیر تعریف می‌شود.

```

\begin{thebibliography}{widest-label}
\bibitem[label]{cite-key}
:
\end{thebibliography}

```

• widest-label: متنی که وقتی چاپ می‌شود تقریباً پهناهای آن به اندازه پهناهای بزرگترین label ای است که توسط دستور \bibitem تولید می‌شود.

• \bibitem[label]{cite-key}: یک ورودی مشخص شده با برچسب مشخص شده در [label] را ایجاد می‌کند. اگر آرگومان label حذف شود، با استفاده از شماره‌دهنده enumi یک عدد به عنوان برچسب ایجاد می‌شود. آرگومان cite-key می‌تواند هر دنباله‌ای از حروف، اعداد و علائم نقطه‌گذاری به جز کاما باشد و از آن برای رجوع به این مرجع استفاده می‌شود. دستور \bibitem برچسب item و cite-key آن را در فایللی با پسوند .aux ذخیره می‌کند. وقتی این فایل با پسوند .aux توسط فرمان \begin{document} خوانده می‌شود ارتباط بین برچسب item و cite-key برقرار شده و از طریق دستور \cite برچسب مربوطه برای منبع مورد نظر در cite-key ایجاد می‌گردد.

• \cite[*text*]{*key-list*}: از این دستور برای رجوع به مرجع مورد نظر در متن استفاده می‌شود. به عنوان مثال دستور \cite[p.2]{knuth} باعث ایجاد عبارت '[knuth,p.2]' در متن می‌شود.

فصل ۴

سبک‌ها

۱.۴ سبک نوشتار

همان‌طور که قبلاً گفته شد در ابتدای پرونده ورودی \LaTeX باید با فرمان \documentstyle ، سبک حروفچینی را مشخص کنیم. در \LaTeX چهار سبک اصلی article (مقاله)، report (گزارش)، book (کتاب) و letter (نامه) وجود دارد.

سبک article بیشتر برای متون ساده و مقاله‌های کم صفحه استفاده می‌شود. با انتخاب این سبک شماره صفحه در پایین صفحه قرار می‌گیرد و صفحات زوج و فرد به طور یکسان حروفچینی می‌شوند. سبک‌های report و book نیز برای نوشتارهای بزرگتر مانند گزارش‌ها و کتاب‌های معمولی به کار می‌روند. البته هر کدام از این سبک‌ها با سبک‌های دیگر تفاوت‌های زیادی دارد. برای مثال در سبک book شماره صفحات در بالا و به همراه نام فصل و قسمت می‌آید و صفحات زوج و فرد دارای حاشیه‌های متفاوتی هستند. در بعضی از نوشتارها مانند مقاله‌های تحقیقاتی، مرسوم است که عنوان نوشتار به همراه نام نویسنده و چکیده‌ای از کل نوشتار در ابتدا قرار داده شود. در \LaTeX نیز فرمان‌هایی برای این منظور در نظر گرفته شده است. عنوان، نویسنده و تاریخ به ترتیب با فرمان‌های \title ، \author و \date مشخص می‌شوند. سپس با استفاده از فرمان \maketitle بخش عنوان ساخته می‌شود.

مثال .

```
\title{An Introduction to Conflict Theory!}  
\author{Tom Cat \and Jerry Mouse}  
\date{3 July 2004}  
\maketitle
```

An Introduction to Conflict Theory!

Tom Cat Jerry Mouse

3 July 2004

همان‌طور که در مثال قبل دیده می‌شود، برای جدا کردن نام نویسندگان از `\and` استفاده می‌شود. برای درج زیرنویس در بخش عنوان مقاله (یا گزارش) از فرمان `\thanks` نیز استفاده می‌شود. به جای 3 July 2004 می‌توان از `\today` استفاده کرد که در آن صورت `LATEX` به طور خودکار تاریخ امروز را قرار می‌دهد. برای حروفچینی چکیده کافی است که متن چکیده را در داخل محیط `abstract` یعنی بین `\begin{abstract}` و `\end{abstract}` قرار دهیم.

سبک `letter` برای آماده کردن نامه در نظر گرفته شده‌است. با انتخاب این سبک می‌توان از فرمان‌هایی استفاده کرد که به سهولت حروفچینی نامه کمک می‌کنند. برای مثال، با کمک دو فرمان `\address` و `\signature` آدرس فرستنده، همچنین نام و سِمَت او مشخص می‌شوند که در هنگام حروفچینی در مکان‌های مناسب قرار می‌گیرند. البته باید متن نامه را در داخل محیط `letter` منظور کرد. ضمناً آدرس گیرنده را باید پس از `\begin{letter}` در داخل آکولاد قرار داد. با استفاده از فرمان‌های `\opening` و `\closing` نیز عبارت‌های شروع و پایان نامه مشخص می‌شوند.

مثال .

```
\documentstyle{letter}
\begin{document}
\address{Your Address \\ Street, City etc. }
\signature{Your name}
\begin{letter}{Iranian Mathematical society\\
P.O. Box 13145-418\\ Tehran, Iran}
\opening{Dear sirs}
...
    Contents of your letter
...
\closing{Best regards}
\end{letter}
\end{document}
```

خروجی را می‌توانید با حروفچینی متن فوق ببینید.

۲.۴ سبک‌های فرعی

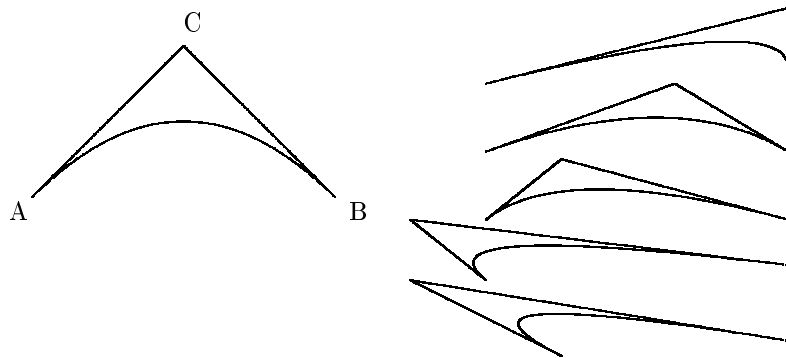
در کنار سبک‌های اصلی می‌توان از سبک‌های فرعی نیز استفاده کرد. برای انتخاب این سبک‌ها باید نام آن‌ها را در بین یک جفت کروشهٔ باز و بسته بلافاصله پس از `\documentstyle` و قبل از آکولاد حاوی نام سبک اصلی قرار داد. چند سبک فرعی را می‌توان با «» از یکدیگر جدا کرد. در مثال `\documentstyle[12pt,twocolumn]{article}` سبک اصلی `article` و سبک‌های فرعی `12pt` و `twocolumn` انتخاب شده‌اند. در ادامه ضمن معرفی این دو سبک فرعی، چند سبک فرعی مهم دیگر را آورده‌ایم.

11pt: اندازهٔ قلم عادی به جای ۱۰ پوینت، ۱۱ پوینت انتخاب می‌شود. بقیهٔ اندازهٔ قلم‌ها و بعضی از ابعاد و

فاصله‌ها نیز به همین نسبت اضافه می‌شوند (مثلاً فاصله عمودی بین خطوط).
 twocolumn: صفحات به صورت دو ستونی حروفچینی می‌شوند.
 twoside: در سبک article بین صفحات فرد و زوج در تنظیم حاشیه تفاوت ایجاد می‌کند.
 leqno: شماره معادلات در فرمول چینی در سمت چپ قرار می‌گیرد.
 fleqno: فرمول‌ها در سبک ریاضی نمایشی چپ‌چین می‌شوند.

۳.۴ سبک فرعی bezier

در صورتی که در ابتدای نوشتار سبک فرعی bezier (bezier.sty) انتخاب شده باشد، می‌توان از فرمان
 $\backslash\text{bezier}\{\text{num}\}(x_1,y_1)(x_2,y_2)(x_3,y_3)$ در محیط picture برای ترسیم منحنی‌های بی‌زی‌یر، که از
 خانواده منحنی‌های اسپلاین هستند، استفاده کرد. این منحنی‌ها از کنار هم قرار دادن نقاط به دست می‌آید. با
 انتخاب سه نقطه این منحنی را تعیین می‌کنیم که دو نقطه آن ابتدا و انتهای منحنی و نقطه سوم نقطه کنترل
 منحنی است.

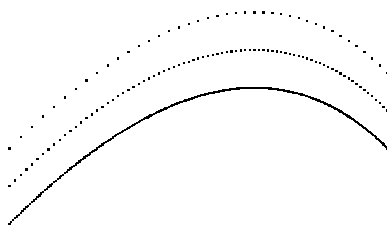


مثال .

اگر A، B و C به ترتیب سه نقطه مفروض باشند، منحنی بین دو نقطه A و B طوری رسم می‌شود که در
 نقطه A بر پاره خط AC و در نقطه B بر پاره خط BC مماس باشد. عدد num تعداد نقاط را در طول این منحنی
 تعیین می‌کند. بدیهی است که اگر تعداد این نقاط کم باشد، مسیر منحنی نقطه چین خواهد شد. لازم به ذکر
 است که در خروجی پاره خط‌های AC و BC دیده نمی‌شوند.

مثال .

```
\unitlength=1mm
\begin{picture}(50,35)
\bezier{40}(0,10)(30,40)(50,20)
\bezier{80}(0,5)(30,35)(50,15)
\bezier{240}(0,0)(30,30)(50,10)
\end{picture}
```



در پایان اشاره می‌کنیم که برای سهولت کار ترسیم می‌توان از برنامه `latexcad.exe` استفاده کرد. با کمک این برنامه می‌توان ضمن مشاهده شکل ترسیم شده، یک فایل خروجی شامل فرمان‌های محیط `picture` برای شکل مورد نظر به دست آورد.

۴.۴ استفاده از شکل یا عکس در متن

در صورتی که بخواهیم در قسمتی از متن از شکل یا عکسی که در یک فایل با پسوند `.ps` یا `.eps` وجود دارد، استفاده کنیم باید ابتدا در قسمت پیش درآمد با دستور `\input{epsf}` فایل را آماده پردازش کنیم و سپس در قسمتی از متن که قصد داریم از شکل یا عکس مورد نظر استفاده کنیم دستورات زیر را تایپ کنیم.

```
\begin{figure}
\epsfysize=5cm
\epsfbox[0 30 300 130]{filename.eps(ps)}
\end{figure}
```

در دستور `\epsfbox`، $(0,30)$ مختصات گوشه پایین سمت راست کادر و $(300,130)$ مختصات گوشه بالای سمت چپ کادر مورد نظر است.

فصل ۵

فارسی‌تک

فارسی‌تک (FarsiTeX) یک سیستم حروفچینی دوزبانه (انگلیسی - فارسی) مستقل، براساس برنامه TeX، Donald Knuth است. TeX و برنامه مشابه اش METAFONT سال‌ها ریاضی‌دانان و نویسندگان فنی سراسر دنیا را سرویس دادند و به آنها کمک کردند تا متون علمی خود را زیباتر و راحت‌تر بنویسند. برخلاف بسیاری از سیستم‌های تهیه متون فارسی، فارسی‌تک نه تنها در حروفچینی ریاضی بسیار قوی است بلکه دارای قابلیت‌هایی در رابطه با علائم و نشانه‌گذاری‌ها نیز می‌باشد. فارسی‌تک (نسخه تغییر یافته از LaTeX) یک بسته حروفچینی انگلیسی - فارسی به منظور برآورده کردن حداقل نیازهای متون فارسی شامل فرمول‌های ریاضی و چاپ فنی است.

۱.۵ مقدمه

چاپ متون علمی فارسی در دهه ۱۹۵۰ توسط انتشارات غلامحسین مصاحب (کسی که سبک قلم ایرانی، که به صورت ایتالیک از راست به چپ در متون مورد استفاده قرار می‌گرفت را ابداع نمود) و انتشارات دانشگاه تهران، که ابزار و وسائل چاپ متون را با بالاترین کیفیت قابل دسترسی آن زمان توسعه داد، شکوفا شد. در آن زمان حروفچین‌ها از بسیاری روش‌های پیشرفته بومی برای توسعه ماشین‌های حروفچینی وارد شده، استفاده می‌کردند. این روش‌ها امروزه «روش‌های چوب‌کبریتی» نامیده می‌شود زیرا در بسیاری از این روش‌ها از قطعات چوب کبریت برای ایجاد فواصل مورد نیاز در فرمول‌های ریاضی استفاده می‌کردند.

این روش‌ها در اواخر دهه ۱۹۷۰ با ظهور ماشین‌های جدید حروفچینی ساخته شده توسط Linotype که راه‌کارهای ساده‌تری برای حروفچینی متون ریاضی شامل متون فارسی فراهم می‌نمود، تغییر یافت. این ماشین‌ها به ناشران تازه وارد از جمله انتشارات دانشگاه تهران و مؤسسه انتشارات فاطمی کمک کرد تا کتاب‌های فنی را در مدت زمان کمتری منتشر نمایند، به طوری که در دهه ۱۹۸۰ و اوایل دهه ۱۹۹۰ حجم زیادی از کتب ریاضی منتشر شد. در سال ۱۹۹۲ با پیدایش دو بسته حروفچینی براساس TeX، به نام‌های TeX_e_parsi و LaTeX_e_farsi جهشی در حروفچینی ریاضیات ایجاد شد. بسته LaTeX_e_farsi به علت ناکارایی در مدت کوتاهی از بین رفت، اما بسته TeX_e_parsi توسط شرکت داده‌کاوی ایران با سرمایه‌گذاری توسط دو مرکز انتشاراتی ذکر شده در فوق، قابل استفاده باقی ماند. طراحی TeX_e_parsi به شدت تحت تأثیر روش Knuth در ایجاد TEX بود و با تحقیقات روی فن چاپ موجود در آن زمان صورت گرفت. بسته نرم‌افزاری مذکور، در یک شکل بسیار مناسب توسط مراکز انتشاراتی ذکر شده و تعدادی گروه ریاضی

استفاده می‌شد، اما متأسفانه برای نویسندگان خاص و دانشجویان قابل استفاده نبود. بنابراین این بسته نمی‌توانست به نویسندگان کمک کند تا خودشان متون خود را آماده کنند. اما جهش بزرگتر بسته نرم‌افزاری دیگری به نام زرنگار بود که در اوایل دهه ۱۹۹۰ به خاطر کیفیت بالای حروفچینی با استفاده از کامپیوترهای شخصی که جریان اصلی حروفچینی با قلم‌های مختلف و زبان نشانه‌گذاری بصری را مورد توجه قرار می‌داد به وجود آمد. بسته مذکور به خاطر کیفیت خوب خروجی و قیمت مناسب محبوبیت زیادی به دست آورد و هنوز هم به طور وسیعی استفاده می‌شود. تخمین زده می‌شد که این نرم‌افزار دومین نرم‌افزار تهیه‌شده متون بعد از نرم‌افزار word در ایران باشد اما متأسفانه کیفیت حروفچینی زرنگار در ریاضیات بسیار ضعیف بود به طوری که باعث به وجود آمدن کتاب‌های فنی با حروفچینی بد شد.

۲.۵ فارسی‌تک FarsiTeX

فارسی‌تک به عنوان یک پروژه دانشگاهی توسط دکتر محمد قدسی در گروه مهندسی کامپیوتر دانشگاه صنعتی شریف آغاز شد. پروژه در سال اول با عنوان FaTeX شناخته شد و در سال ۱۹۹۱ به عنوان سه پروژه کارشناسی با نظارت دکتر قدسی برای ارائه مفاهیم پایه آن ادامه یافت. سرانجام بعد از آزمایشات زیاد، FarsiTeX روی موتور TEX_XET و محیط MS-DOS قرار گرفت.

به مدت دو سال نویسندگان استفاده محدودی از این سیستم داشتند، تا اینکه در اوایل سال ۱۹۹۶ دکتر قدسی تیم جدیدی برای توجه بیشتر به نسخه عمومی نرم‌افزار تحت نظارت GNU (مؤسسه نرم‌افزاری مستقل با مسئولیت محدود، ۱۹۹۱) ایجاد کرد.

اولین نسخه همگانی فارسی‌تک در اکتبر ۱۹۹۶، به عنوان ضمیمه‌ای برای بخش emTeX که در آن زمان عمومیت زیادی داشت به وجود آمد و صراحتاً به عنوان نرم‌افزاری با کیفیت β شناخته شد. همچنین فارسی‌تک اولین نرم‌افزار ایرانی بود که تحت نظارت GPL عرضه شد. علی‌رغم تصور مؤلفین فارسی‌تک، مبنی بر محدود بودن مخاطبین آن به دلیل مشکل درجه بندی و معایب شناخته شده مختلف، به سرعت در میان دانشجویان، اساتید ریاضیات، مهندسی کامپیوتر و فیزیک در سراسر کشور گسترش پیدا کرد، چرا که تنها گزینه موجود و مناسب برای کارهای تایپی آن‌ها بود. فارسی‌تک همچنین توسط اساتید جدید که بعد از تحصیلاتشان در یک دانشگاه آمریکایی یا اروپایی به ایران برگشته بودند، ترویج یافت. تیم پروژه فارسی‌تک در سال ۱۹۹۶ تشکیل شد و در مرکز کامپیوتر دانشگاه صنعتی شریف به کار خود ادامه می‌دهد.

۳.۵ سبک کلی فایل‌های فارسی‌تک

یک متن در فارسی‌تک در قالب فایل با پسوند .ftx ذخیره می‌شود. این فایل نیز از دو قسمت اصلی پیش در آمد و متن اصلی تشکیل شده است.

در قسمت پیش در آمد دستورات لازم مانند \LaTeX است با این تفاوت که دستورات می‌توانند از سمت راست تایپ شوند. هم‌چنین باید در دستور \documentstyle سبک فرعی farsi بارگذاری شود:

```
{article} [farsi] documetstyle\
:
{document} begin\
:
{document} end\
```

در ویرایشگر فارسی‌تک دو قلم فارسی و انگلیسی فعال وجود دارد که بسته به نیاز می‌توان آن‌ها را انتخاب کرد. برای نوشتن از چپ به راست می‌توان از کلید `shift+>` استفاده کرد. برای تغییر محیط از فارسی به انگلیسی و از انگلیسی به فارسی دستورهای `english\` و `farsi\` به کار می‌روند. توجه کنید که می‌توانید برای چپ چین کردن یک متن در محیط فارسی، متن مورد نظر را بین دو دستور `{leftline} begin\` و `{leftline} end\` قرار دهید.

فصل ۶

جدول‌ها

در ۱۲ جدول این فصل، نمادهای مختلفی را که \LaTeX در اختیار ما می‌گذارد آورده‌ایم. نمادهای جدول‌های ۱ و ۲ را در هر نوع متنی می‌توان به کاربرد و نمادهای ده جدول دیگر تنها در متن ریاضی قابل تعریف هستند.

Type	Typset	Type	Typset	Type	Typset
$\backslash'{}o\}$	ò	$\backslash"{}o\}$	ö	$\backslash\sim{}o\}$	õ
$\backslash'{}{o}\}$	ó	$\backslash={o}\}$	ō	$\backslashH{o}\}$	ö
$\backslash\sim{}{o}\}$	ô	$\backslash.{}o\}$	ó	$\backslasht{}o\}$	ô
$\backslashu{}o\}$	ö	$\backslashv{}o\}$	ö	$\backslashc{}o\}$	ø
$\backslashd{}o\}$	ø	$\backslashb{}o\}$	ø		

جدول ۱: Accents

Type	Typset	Type	Typset	Type	Typset
\backslashdag	†	\backslashS	§	\backslashcopyright	©
\backslashddag	‡	\backslashP	¶	\backslashpounds	£
\backslashoe	œ	\backslashO	Ø	\backslashl	ł
\backslashOE	Œ	\backslashAA	Å	\backslashae	æ
\backslasho	ø	\backslashss	ß	\backslashAE	Æ

جدول ۲: Foreign Symbols

Lowercase

Type	Typset	Type	Typset	Type	Typset
<code>\alpha</code>	α	<code>\theta</code>	θ	<code>o</code>	o
<code>\beta</code>	β	<code>\vartheta</code>	ϑ	<code>\pi</code>	π
<code>\gamma</code>	γ	<code>\iota</code>	ι	<code>\varpi</code>	ϖ
<code>\delta</code>	δ	<code>\kappa</code>	κ	<code>\rho</code>	ρ
<code>\epsilon</code>	ϵ	<code>\lambda</code>	λ	<code>\varrho</code>	ϱ
<code>\varepsilon</code>	ε	<code>\mu</code>	μ	<code>\sigma</code>	σ
<code>\zeta</code>	ζ	<code>\nu</code>	ν	<code>\varsigma</code>	ς
<code>\eta</code>	η	<code>\xi</code>	ξ	<code>\tau</code>	τ
<code>\upsilon</code>	υ	<code>\phi</code>	ϕ	<code>\varphi</code>	φ
<code>\chi</code>	χ	<code>\psi</code>	ψ	<code>\omega</code>	ω

Uppercase

Type	Typset	Type	Typset	Type	Typset
<code>\Gamma</code>	Γ	<code>\Lambda</code>	Λ	<code>\Sigma</code>	Σ
<code>\Delta</code>	Δ	<code>\Xi</code>	Ξ	<code>\Upsilon</code>	Υ
<code>\Theta</code>	Θ	<code>\Pi</code>	Π	<code>\Phi</code>	Φ
<code>\Psi</code>	Ψ	<code>\Omega</code>	Ω		

جدول ۳: Greek Letters

Type	Typset	Type	Typset	Type	Typset
<code>\pm</code>	\pm	<code>\cap</code>	\cap	<code>\diamond</code>	\diamond
<code>\mp</code>	\mp	<code>\cup</code>	\cup	<code>\bigtriangleup</code>	\triangle
<code>\times</code>	\times	<code>\uplus</code>	\uplus	<code>\bigtriangledown</code>	∇
<code>\div</code>	\div	<code>\sqcap</code>	\sqcap	<code>\triangleleft</code>	\triangleleft
<code>\ast</code>	$*$	<code>\sqcup</code>	\sqcup	<code>\triangleright</code>	\triangleright
<code>\star</code>	\star	<code>\vee</code>	\vee	<code>\lhd</code>	\triangleleft
<code>\circ</code>	\circ	<code>\wedge</code>	\wedge	<code>\rhd</code>	\triangleright
<code>\bullet</code>	\bullet	<code>\setminus</code>	\setminus	<code>\unlhd</code>	\triangleleft
<code>\cdot</code>	\cdot	<code>\wr</code>	\wr	<code>\unrhd</code>	\triangleright
<code>\oplus</code>	\oplus	<code>\ominus</code>	\ominus	<code>\otimes</code>	
<code>\oslash</code>	\oslash	<code>\odot</code>	\odot	<code>\bigcirc</code>	\bigcirc
<code>\dagger</code>	\dagger	<code>\ddagger</code>	\ddagger	<code>\amalg</code>	\amalg

جدول ۴ : Binary operation symbols

Type	Typset	Type	Typset	Type	Typset
<code>\leq</code>	\leq	<code>\geq</code>	\geq	<code>\equiv</code>	\equiv
<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>\ll</code>	\ll	<code>\gg</code>	\gg	<code>\asymp</code>	\asymp
<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</code>	\sqsubset	<code>\sqsupset</code>	\sqsupset	<code>\neq</code>	\neq
<code>\sqsubseteq</code>	\sqsubseteq	<code>\sqsupseteq</code>	\sqsupseteq	<code>\doteq</code>	\doteq
<code>\in</code>	\in	<code>\ni</code>	\ni	<code>\propto</code>	\propto
<code>\vdash</code>	\vdash	<code>\dashv</code>	\dashv	<code>\models</code>	\models
<code>\perp</code>	\perp	<code>\mid</code>	\mid	<code>\parallel</code>	\parallel
<code>\bowtie</code>	\bowtie	<code>\Join</code>	\Join	<code>\smile</code>	\smile
<code>\frown</code>	\frown				

جدول ۵ : Relation Symbols

Type	Typset	Type	Typset
<code>\leftarrow</code>	\leftarrow	<code>\longleftarrow</code>	\longleftarrow
<code>\Leftarrow</code>	\Leftarrow	<code>\Longleftarrow</code>	\Longleftarrow
<code>\rightarrow</code>	\rightarrow	<code>\longrightarrow</code>	\longrightarrow
<code>\Rightarrow</code>	\Rightarrow	<code>\Longrightarrow</code>	\Longrightarrow
<code>\leftrightarrow</code>	\leftrightarrow	<code>\longleftrightarrow</code>	\longleftrightarrow
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\Longleftrightarrow</code>	\Longleftrightarrow
<code>\mapsto</code>	\mapsto	<code>\longmapsto</code>	\longmapsto
<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookleftarrow</code>	\hookleftarrow
<code>\leftharpoonup</code>	\leftharpoonup	<code>\rightharpoonup</code>	\rightharpoonup
<code>\leftharpoondown</code>	\leftharpoondown	<code>\rightharpoondown</code>	\rightharpoondown
<code>\rightleftharpoons</code>		<code>\leadsto</code>	\leadsto
<code>\uparrow</code>	\uparrow	<code>\Uparrow</code>	\Uparrow
<code>\downarrow</code>	\downarrow	<code>\Downarrow</code>	\Downarrow
<code>\updownarrow</code>	\updownarrow	<code>\Updownarrow</code>	\Updownarrow
<code>\nearrow</code>	\nearrow	<code>\searrow</code>	\searrow
<code>\swarrow</code>	\swarrow	<code>\nwarrow</code>	\nwarrow

جدول ۶: Arrow Symbols

Type	Typset	Type	Typset	Type	Typset
<code>\aleph</code>	\aleph	<code>\prime</code>	$'$	<code>\forall</code>	\forall
<code>\hbar</code>	\hbar	<code>\emptyset</code>	\emptyset	<code>\exists</code>	\exists
<code>\imath</code>	\imath	<code>\nabla</code>	∇	<code>\neg</code>	\neg
<code>\jmath</code>	\jmath	<code>\surd</code>	\surd	<code>\flat</code>	\flat
<code>\ell</code>	ℓ	<code>\top</code>	\top	<code>\natural</code>	\natural
<code>\wp</code>	\wp	<code>\bot</code>	\bot	<code>\sharp</code>	\sharp
<code>\Re</code>	\Re	<code>\ </code>	$\ $	<code>\backslash</code>	\backslash
<code>\Im</code>	\Im	<code>\angle</code>	\angle	<code>\partial</code>	∂
<code>\mho</code>	\mho	<code>\infty</code>	∞	<code>\Box</code>	\Box
<code>\Diamond</code>	\Diamond	<code>\triangle</code>	\triangle	<code>\clubsuit</code>	\clubsuit
<code>\diamondsuit</code>	\diamondsuit	<code>\heartsuit</code>	\heartsuit	<code>\spadesuit</code>	\spadesuit

جدول ۷: Miscellaneous Symbols

Source	Name	Type	Typset	
L ^A T _E X	left parenthesis	((
	right parenthesis))	
	left bracket	[or \lbrack	[
	right bracket] or \rbrack]	
	left brace	\{ or \lbrace	{	
	right brace	\} or \rbrace	}	
	backslash	\backslash	\	
	forward slash	/	/	
	left angle bracket	\langle	<	
	right angle bracket	\rangle	>	
	vertical line	or \vert		
	double vertical line	\ or \Vert		
	left floor	\lfloor	⌊	
	right floor	\rfloor	⌋	
	left ceiling	\lceil	⌈	
	right ceiling	\rceil	⌉	
	amsmath	upper-left corner	\ulcorner	⌞
		upper-right corner	\urcorner	⌟
		lower-left corner	\llcorner	⌝
		lower-right corner	\lrcorner	⌞

Standard delimiters : ۸ جدول

Type	Typset	Type	Typset	Type	Typset	Type	Typset
\arccos	arccos	\cot	cot	\hom	hom	\sin	sin
\arcsin	arcsin	\coth	coth	\ker	ker	\sinh	sinh
\arctan	arctan	\csc	csc	\lg	lg	\tan	tan
\arg	arg	\deg	deg	\ln	ln	\tanh	tanh
\cos	cos	\dim	dim	\log	log		
\cosh	cosh	\exp	exp	\sec	sec		

Operators without limits : ۹ جدول

Source	Type	Typset	Type	Typset
L ^A T _E X	<code>\det</code>	det	<code>\limsup</code>	lim sup
	<code>\gcd</code>	gcd	<code>\max</code>	max
	<code>\inf</code>	inf	<code>\min</code>	min
	<code>\lim</code>	lim	<code>\Pr</code>	Pr
	<code>\liminf</code>	lim inf	<code>\sup</code>	sup
amsmath	<code>\injlim</code>	inj lim	<code>\projlim</code>	proj lim
	<code>\varliminf</code>	\varliminf	<code>\varlimsup</code>	\varlimsup
	<code>\varinjlim</code>	\varinjlim	<code>\varprojlim</code>	\varprojlim

جدول ۱۰: Operators with limits

Type	Inline	Displayed
<code>\int_{a}^{b}</code>	\int_a^b	\int_a^b
<code>\oint_{a}^{b}</code>	\oint_a^b	\oint_a^b
<code>\prod_{i=1}^n</code>	$\prod_{i=1}^n$	$\prod_{i=1}^n$
<code>\coprod_{i=1}^n</code>	$\coprod_{i=1}^n$	$\coprod_{i=1}^n$
<code>\bigcap_{i=1}^n</code>	$\bigcap_{i=1}^n$	$\bigcap_{i=1}^n$
<code>\bigcup_{i=1}^n</code>	$\bigcup_{i=1}^n$	$\bigcup_{i=1}^n$
<code>\bigwedge_{i=1}^n</code>	$\bigwedge_{i=1}^n$	$\bigwedge_{i=1}^n$
<code>\bigvee_{i=1}^n</code>	$\bigvee_{i=1}^n$	$\bigvee_{i=1}^n$
<code>\bigsqcup_{i=1}^n</code>	$\bigsqcup_{i=1}^n$	$\bigsqcup_{i=1}^n$
<code>\biguplus_{i=1}^n</code>	$\biguplus_{i=1}^n$	$\biguplus_{i=1}^n$
<code>\bigotimes_{i=1}^n</code>	$\bigotimes_{i=1}^n$	$\bigotimes_{i=1}^n$
<code>\bigoplus_{i=1}^n</code>	$\bigoplus_{i=1}^n$	$\bigoplus_{i=1}^n$
<code>\bigodot_{i=1}^n</code>	$\bigodot_{i=1}^n$	$\bigodot_{i=1}^n$
<code>\sum_{i=1}^n</code>	$\sum_{i=1}^n$	$\sum_{i=1}^n$

جدول ۱۱ : Large operators

Type	Typset
<code>\a \equiv v \pmod{\theta}</code>	$a \equiv v \pmod{\theta}$
<code>\a \bmod b</code>	$a \bmod b$
<code>\a \equiv v \pmod{\theta}</code>	$a \equiv v \pmod{\theta}$
<code>\a \equiv v \pmod{\theta}</code>	$a \equiv v(\theta)$

جدول ۱۲ : Congruences

- [1] G. Grätzer. *Math into L^AT_EX*. Birkhäuser Boston, 2000.
- [2] D.E. Knuth. *The TeXbook*. Amer. Math. Soc., 1984.
- [3] L. Lamport. *L^AT_EX: A Document preparation System*. Addison-Wesley, 1986.
- [4] L. Lamport, et. al. *L^AT_EX₂ ϵ* . 1994.