بسم الله الرحمن الرحیم

راهنمای فرمول‌نویسی تک

یکی از استانداردهای فراگیر برای نوشتن عبارات ریاضی استاندارد مورد استفاده در خانوادهٔ تک است. با گسترش وب و به وجود آمدن ابزارهای قدرتمندی همچون $MathJax$، بسیاری از سایت‌های ریاضی نیز برای نمایش عبارات ریاضی از این استاندارد استفاده می‌کنند. از این رو یادگیری این استاندارد می‌تواند مفید باشد.

راه‌های یادگیری فرمول‌نویسی لاتک:

۱. ادامه به مطالعهٔ همین نوشته!

۲. مطالعهٔ مطالب آموزشی لاتک در سطح اینترنت به ویژه
راهنمای سایت $StackOverflow$ برای فرمول‌نویسی $MathJax$ و
راهنمای ویکی‌پدیا برای فرمول‌نویسی.

۳. استفاده از سایت‌های تولید عبارت لاتک توسط واسط گرافیکی مانند
ویرایشگر سایت $codecogs$ و
ویرایشگر سایت $hostmath$.

۴. استفاده از افزونه $MathType$ در محیط $Word$ و دریافت عبارت لاتک با دکمهٔ $Toggle\ Tex$ در نوار ابزار برنامه.

۵. نوشتن در سایت‌هایی که از $MathJax$ استفاده می‌کنند و یا دست به کار شدن و کار با محیط واقعی لاتک.

پس همین الان به سایت http://www.mathjax.org/demos/ رفته و در جعبهٔ موجود عبارت `$a \times b = c$` را وارد کنید تا $a \times b = c$ نمایش داده شود. الآن می‌توانیم حدس بزنیم که `\times` علامت ضرب را نشان می‌دهد! در ادامهٔ این نوشته سعی کنید همهٔ عبارات معرفی شده را به همین صورت آزمایش کنید.

نوشتن عبارات در جای مناسب

پیش از هر چیز باید بدانیم که برای نوشتن یک عبارت ریاضی تک در میان متن، باید آن را داخل `$ ... $` قرار دهیم. برای مثال جملهٔ «برای حل معادلهٔ `$ax^2+bx+c=0$` روش زیر را به کار می‌گیریم.» پس از پردازش به صورت «برای حل معادلهٔ $ax^۲ + bx + c = ۰$ روش زیر را به کار می‌گیریم.» نمایش داده خواهد شد.

اگر بخواهیم عبارت ما در خطی جداگانه و با فضای بیشتر نمایش داده شود، آن را در میان `$$ ... $$` قرار می‌دهیم. برای مثال جملهٔ «و در نهایت پاسخ به شکل `$$\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$` خواهد بود.» (نگران این

عبارت نباشید، سخت‌تر از این را هم یاد خواهیم گرفت) پس از پردازش به شکل «و در نهایت پاسخ به شکل

$$\frac{-b \pm \sqrt{b^۲ - ۴ac}}{۲a}$$

خواهد بود.» نمایش داده می‌شود.

گاهی با قرار دادن عبارت میان `$ ... $` و `$$ ... $$` دو نتیجهٔ متفاوت به دست می‌آوریم. برای مثال عبارت `$\sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6}$` در فرم اول به شکل $\sum_{i=۰}^{n} i^۲ = \frac{(n^۲+n)(۲n+۱)}{۶}$ و در فرم دوم به شکل

$$\sum_{i=۰}^{n} i^۲ = \frac{(n^۲+n)(۲n+۱)}{۶}$$

نمایش داده خواهد شد. در فرم اول پردازنده به هر حال باید عبارت را در «یک خط» جای دهد و به همین خاطر عبارت کمی جمع و جورتر می‌شود ولی در فرم دوم چنین نیست.

دو شیوهٔ دیگر (`\[ ... \]` و `\( ... \)`) برای نوشتن عبارات ریاضی وجود دارد که با کمی جستجو و یا سعی و خطا به سادگی می‌توانید رفتار آنها را کشف کنید.

عبارات سادهٔ ریاضی

عبارات سادهٔ ریاضی واقعا ساده هستند! برای نوشتن $+$، $-$، $()$، $[]$ و $=$ کافی است همین حروف را در عبارات خود به کار ببرید. برای مثال با نوشتن `$(2+2-4)=[4-2]-2$` ، $(۲ + ۲ - ۴) = [۴ - ۲] - ۲$ را به دست می‌آورید.

خوب تا اینجای کار جمع و تفریق را بلدیم، برای ضرب کافی است دستور `\times` را به کار ببریم. این دستور دقیقا یک $\times$ در عبارت شما قرار می‌دهد. برای مثال `$2 \times 2 = 4$` به شکل $۲ \times ۲ = ۴$ نمایش داده می‌شود.

دستور تقسیم هم به همین صورت است. کافی است `\div` را در عبارت خود به کار ببرید تا $\div$ ظاهر شود.

از آنجایی که نمایش عبارات ریاضی توسط $\div$ کار مشکلی به نظر می‌رسد، بهتر است خیلی زود نمایش کسرها را نیز یاد بگیریم، برای نمایش کسر به صورت $\frac{abc}{xyz}$ کافی است دستور `$\frac{abc}{xyz}$` را وارد کنید.
در واقع دستور `\frac{}{}` دو ورودی دریافت کرده و اولی را در صورت و دومی را در مخرج قرار می‌دهد. کسرهای را می‌توان به صورت تودرتو هم به کار برد برای مثال `$$1+\frac{1}{2+\frac{2}{3+\frac{3}{4}}}$$` عبارت زیر را تولید می‌کند.

$$۱ + \frac{۱}{۲ + \frac{۲}{۳ + \frac{۳}{۴}}}$$

چنانچه کسرهای شما پیچیده باشند، ممکن است ترجیه دهید از دستور `\over` استفاده کنید. مثلاً برای نمایش کسر $\frac{a+1}{b+1}$ کافی است دستور `${a+1 \over b+1}$` را به کار ببرید.

توجه کنید که در دستور `\frac{}{}` از `{}` استفاده کردیم. پس طبیعی است که برای نوشتن `{` دچار دردسر شویم. برای نوشتن «آکولاد» لازم است پیش از حرف، `\` استفاده کنیم. برای مثال `$\{2,3,5,7\}$` موجب تولید عبارت $\{۲,۳,۵,۷\}$ می‌شود.

و اما «توان» و «اندیس»، برای نوشتن عبارتی مانند a^b از `$a^b$` و برای نوشتن عبارتی به شکل a_i از `$a_i$` استفاده می‌کنیم. توان و اندیس می‌توانند همزمان نیز به کار برده شوند. برای مثال `$a_i^2$` عبارت $a_i^۲$ را تولید می‌کند.

نکتهٔ مهم: در صورتی که طول توان و اندیس از یک حرف بیشتر باشد، لازم است آن را به وسیلهٔ `{...}` در یک «گروه» قرار دهید. برای مثال `$13^93$` عبارت ۱۳^93 را تولید می‌کند که مطلوب نیست. برای به دست آوردن عبارت $۱۳^{۹۳}$ لازم است `$13^{93}$` نوشته شود. نکتهٔ جالب توجه اینجاست که شما در همه جای عبارات خود می‌توانید عناصر را گروه کنید. برای مثال برای به دست آوردن $۱۴۳۵+۱۳۹۳^{۲۰۱۴}$ می‌توانید عبارت `${13}{93}^{20{14}}{+{{14}35}}$` را بنویسید.

توجه کنید که گروه‌ها نیز می‌توانند تودرتو باشند. برای مثال عبارت $۱۳۹۳^{۲۰۱۴^{۱۴۳۵}}$ توسط `${1393}^{2014^{1435}}$` تولید می‌شود. به عنوان کنجکاوی خودتان عبارت `${1393^2014}^{1435}$` را امتحان کرده و نتیجه را با نتیجهٔ عبارت قبل مقایسه کنید.

برای نوشتن عبارات رادیکالی دستور `\sqrt` به کار می‌آید. برای نوشتن عبارات مانند $\sqrt{x}$ از `$\sqrt{x}$` و برای نوشتن عبارت $\sqrt[۳]{۲}$ از `$\sqrt[3]{2}$` استفاده می‌کنیم.

طبیعی است در عبارات پیچیده به جای `$\sqrt[a]{bc}$` که موجب $\sqrt[a]{bc}$ می‌شود از `$(bc)^{1 \over a}$` که موجب $(bc)^{\frac{۱}{a}}$ می‌شود استفاده کنیم.

نکتهٔ اضافی: می‌توان حدس زد که چون پس از `\sqrt` از `[]` استفاده کردیم، عبارت `$\sqrt[1+]+1]{2}$` موجب بروز خطای منطقی شود (یعنی خروجی $\sqrt[۱+]{+۱]۲}$ که مورد انتظار ما نیست تولید شود). برای تولید عبارت (هر چند بی‌معنی) $\sqrt[۱+]{۲}$ می‌توان از `$\sqrt[1+\rbrack+1]{2}$` استفاده کرد. در واقع `\lbrack` و `\rbrack` همواره (و به ویژه در موارد خاص) موجب تولید [و] می‌شوند.

حروف یونانی

برای نوشتن حروف یونایی کافی است پیش از نام آنها `\` قرار دهید. برای مثال `\gamma` گامای کوچک(γ) و `\Gamma` گامای بزرگ(Γ) را نشان می‌دهد. در «پیوست ۱» لیست کاملی از حروف یونانی قرار دارد.

تکمیل حروف و دستورات ریاضی

در قسمت‌های قبل حرف‌ها و دستورات سادهٔ ریاضی را بررسی کردیم. در ادامه این کار را تکمیل می‌کنیم.

برای نوشتن «روابط و نابرابری‌ها» از دستورات زیر استفاده می‌کنیم.

`\prec` ≺ ، `\equiv` ≡ ، `\cong` ≅ ، `\sim` ∼ ، `\approx` ≈ ، `\neq` ≠ ، `\neq` ≠ ، `\ge` ≥ ، `\le` ≤ ، `>` > ، `<` <

با تقریب خوبی قرار دادن `\not` پشت هر عبارت روی آن یک خط می‌کشد. برای مثال `$\not\le$` عبارت $\not\le$ و `$\not\sim$` عبارت $\not\sim$ را تولید می‌کند.

نکته: برای ∤ (عاد نکردن) بهتر است به جای استفاده از `\not` به همراه `|` که خروجی آن ∤ بوده و زیبا نیست، از دستور زیبای `$\nmid$` استفاده کرد. خوب است اشاره کنیم `$\mid$` در واقع همان | (عاد کردن) است.

برای ایجاد علامت‌های مثبت-منفی، عبارات `\pm` و `\mp` به ترتیب خروجی‌های ± و ∓ را تولید می‌کنند.

دستورات ریاضی تک شامل انواع متنوع «فلش‌ها» هستند. این دستورات می‌توانند برای نتیجه گیری‌های عبارات و پرهیز از توضیح بی‌مورد استفاده شوند. برای مثال «از $a>b>0$ می‌توان $a^2>b^2$ را نتیجه گرفت.» را می‌توان به شکل کوتاه‌تر و البته خواناتر «$a>b>0 \Rightarrow a^2>b^2$» خلاصه نمود. برای به دست آوردن عبارت اخیر کافی است `$a>b>0 \Rightarrow a^2>b^2$` را بنویسید. در زیر می‌توانید تعدادی از فلش‌های معمول را مشاهده کنید.
دستورات لاتک به کوچک و بزرگ بودن حروف حساس هستند. (به → و ⇒ دقت کنید.)

`\to` → ، `\rightarrow` → ، `\leftarrow` ← ، `\leftrightarrow` ↔ ، `\longleftrightarrow` ⟷ ،
`\mapsto` ↦ ، `\Rightarrow` ⇒ ، `\Leftarrow` ⇐ ، `\Leftrightarrow` ⇔ ، `\Longleftrightarrow` ⟺

برای نوشتن مجموع‌ها، انتگرال‌ها و ... ابتدا توجه می‌کنیم که حروف مشخص کنندهٔ هر یک ، `\prod` = ∏ ، `\sum` = ∑
`\iiint` = ∭ ، `\iint` = ∬ ، `\int` = ∫ ، `\bigcap` = ⋂ ، `\bigcup` = ⋃ هستند. برای مشخص کردن «حدود» این دستورات دقیقا مانند مشخص کردن «توان» و «اندیس» عمل می‌کنیم. برای مثال برای نوشتن عبارت $\sum_{i=0}^{n}\frac{1}{i}$ می‌نویسیم `$\sum_{i=0}^{n}{1 \over i}$`. اگر همین دستور را میان `$$...$$` قرار دهیم، شکل زیر نتیجه می‌شود.

$$\sum_{i=0}^{n}\frac{1}{i}$$

برای استفادهٔ زیباتر عبارات $\min, \max, \ln, \sin, \cot, \log, \lim, \cdots$ ، (برای مثال $\lim$ به جای lim `\lim`) می‌توانیم از دستورات `\min,\max,\ln,\sin,\cot,\log,\lim,...` استفاده کنیم. به طور خاص برای تعیین حدود دستور lim می‌توانیم از `$\lim_{x\to 0}$` $\lim_{x\to 0}$ استفاده کنیم. (این دستور را بین `$$...$$` نیز امتحان کنید.)

برای نوشتن عبارات منطقی و جبر مجموعه‌ای کافی‌است دستورات زیر را بلد باشید:

$\cup$ `\cup` ، $\cap$ `\cap` ، $\setminus$ `\setminus` ، $\subset$ `\subset` ، $\subseteq$ `\subseteq` ، $\subsetneq$ `\subsetneq` ، $\supset$ `\supset`
$\in$ `\in` ، $\notin$ `\notin` ، $\emptyset$ `\emptyset` ، $\forall$ `\forall` ، $\exists$ `\exists` ، $\nexists$ `\nexists` ، $\therefore$ `\therefore`
$\because$ `\because` ، $\land$ `\land` ، $\lor$ `\lor` ، $\lnot$ `\lnot` ، $\top$ `\top` ، $\bot$ `\bot` ، $\vdash$ `\vdash` ، $\vDash$ `\vDash`

گاهی برای نوشتن عبارات هندسی لازم است تا روی زاویه، بردار و ... علامت‌های خاصی را قرار دهیم. مثال‌های زیر برای این امر به قدر کفایت گویا هستند.

$\hat{ABC}$ `\hat{ABC}` ، $\widehat{ABC}$ `\widehat{ABC}` ، $\vec{ABC}$ `\vec{ABC}` ، $\overrightarrow{ABC}$ `\overrightarrow{ABC}`
$\bar{ABC}$ `\bar{ABC}` ، $\overline{ABC}$ `\overline{ABC}` ، $\overleftarrow{ABC}$ `\overleftarrow{ABC}`

دستورات زیر نیز در عبارات هندسی بسیار پرکاربرد هستند.

$\parallel$ `\parallel` ، $\nparallel$ `\nparallel` ، $\perp$ `\perp` ، 45° `45^\circ` ، $\triangleq$ `\triangleq` ، A' `A^\prime`
$\angle$ `\angle` ، $\sphericalangle$ `\sphericalangle` ، $\measuredangle$ `\measuredangle` ، $\Box$ `\Box` ، $\bigcirc$ `\bigcirc` ، $\triangle$ `\triangle`

بعضی از کتاب‌ها برای نوشتن «انتخاب m از n» از نماد $\binom{n}{m}$ استفاده می‌کنند. برای نوشتن این عبارت کافی است از `${n \choose m}$` و یا `$\binom{n}{m}$` استفاده کنید.

بهتر است به جای سه بار استفاده از نقطه، از دستور `\ldots` ... برای سه نقطه در سطح پایین و از دستور `\cdots` $\cdots$ برای سه نقطه در سطح میانی استفاده کنید. برای مثال `$a_1+a_2+\cdots+a_n$` که نتیجهٔ $a_1+a_2+\cdots+a_n$ است بسیار چشم نواز تر از $a_1+a_2+...+a_n$ جلوه می‌کند.

برای نمایش تحدید توابع به مقادیر خاص از `|_{}` استفاده می‌کنیم. به عنوان مثال برای به دست آوردن عبارت $f(n)=n^k+n|_{n=p}$ از `$f(n)=n^k+n|_{n=p}$` استفاده می‌کنیم.

در مورد بی‌نهایت‌ها به معرفی دو نماد ∞ `\infty` و $\aleph_0$ `\aleph_0` بسنده می‌کنیم. برای نمایش قسمت حقیقی و موهومی اعداد مختلط از $\Re$ `\Re` و $\Im$ `\Im` استفاده می‌کنیم. همچنین برای مشتق‌های پاره‌ای و گرادیان نیز می‌توان از دستورات ∂ `\partial` و ∇ `\nabla` استفاده نمود.

و در نهایت برای نمایش هم‌نهشتی اعداد سه شیوهٔ زیر را پیشنهاد می‌کنیم.

$a \equiv b \pmod{p}$ `$a \equiv b \pmod{p}$`
$a \overset{p}{\equiv} b$ `$a \overset{p}{\equiv} b$`
$a \overset{\underset{\mathrm{p}}{}}{\equiv} b$ `$a \overset{\underset{\mathrm{p}}{}}{\equiv} b$`

قلم‌های معنی دار

مرسوم است برخی حروف و عبارات ریاضی را به شکل خاصی بنویسند. برای مثال عموماً مجموعه اعداد طبیعی و حقیقی را با نماد $\mathbb{N}$ و $\mathbb{R}$ نشان می‌دهند. به همین خاطر چنین قلم‌هایی در استاندارد لاتک پیش‌بینی شده‌است.

آنچه بیشتر از هر چیز مورد استفاده است قلم‌های blackboard bold هستند که در نتیجهٔ آنها $\mathbb{N}$ و $\mathbb{R}$ تولید می‌شوند. برای استفاده از این حروف کافی است حرف (یا حروف) مورد نظر خود را در دستور `\mathbb{}` و یا `\Bbb{}` به کار ببرید. نتیجهٔ این دستور بر روی حروف لاتین به شکل زیر خواهد بود. این دستور تنها بر حروف بزرگ عمل می‌کند.

$\Bbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ `$\Bbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$`

قلم‌های خاص دیگری نیز در لاتک موجود هستند که صرفاً به ارائهٔ مثال‌هایی از بعضی از آنها بسنده می‌کنیم.

$\mathbf{IRaN}$ `$\mathbf{IRaN}$`، $\mathtt{IRaN}$ `$\mathtt{IRaN}$`، IRaN `$\mathrm{IRaN}$`، $\mathcal{IR}\mathcal{N}$ `$\mathcal{IR}\mathcal{N}$`
IRaN `$\mathit{IRaN}$`، IRaN `$\mathsf{IRaN}$`، $\mathfrak{IRaN1393}$ `$\mathfrak{IRaN 1393}$`

پرانتزگذاری‌ها

پس از مدتی کار با دستورات لاتک به این نتیجه خواهید رسید که بعضی عبارات به اندازهٔ کافی زیبا نیستند. برای مثال براکت بیرونی عبارت $[\frac{1+\binom{n}{2}}{1+y^2}]^3$ حاصل از `$[1+{\binom{n}{2} \over 1+y^2}]^3$` خیلی کوچک به نظر می‌رسد. برای رفع مشکل اندازهٔ براکت‌ها راهکاری وجود دارد. دستورهای `\left[ , \right]` جایگزین علامت‌های `[ , ]` هستند به نحوی که علاوه بر همان معنا، خود را با عبارت داخلشان تنظیم می‌کنند. برای مثال چنانچه همان عبارت قبل را به `$\left[1+{\binom{n}{2} \over 1+y^2}\right]^3$` تغییر دهیم، نتیجه $\left[1+\frac{\binom{n}{2}}{1+y^2}\right]^3$ می‌شود.

همین روش برای انواع پرانتزها به ویژه `( )` $(\frac{x^2}{3})$، `\{ \}` $\{\frac{x^2}{3}\}$، `||` $|\frac{x^2}{3}|$، `\langle \rangle` $\langle\frac{x^2}{3}\rangle$، `\lceil \rceil` $\lceil\frac{x^2}{3}\rceil$، `\lfloor \rfloor` $\lfloor\frac{x^2}{3}\rfloor$ قابل پیاده سازی است. برای مثال $\left\{\frac{x^2}{3}\right\}$ و $\left\lceil\frac{x^2}{3}\right\rceil$ را می‌توان توسط `$\left\{{x^2 \over 3}\right\}$` و `$\left\lceil{x^2 \over 3}\right\rceil$` به دست آورد.

یک نکتهٔ بامزه این است که پرانتز سمت چپ می‌تواند از یک نوع و پرانتز سمت راست از نوع دیگر باشد. برای مثال می‌توان $\left\{\frac{x^2}{3}\right\rangle$ را به وسیلهٔ عبارت `$\left\langle{x^2 \over 3}\right\}$` به دست آورد. نکتهٔ بامزهٔ دیگر این است که به وسیلهٔ `.` (نقطه) بعد از `\left` یا `\right` می‌توان پرانتز نامرئی تولید کرد! برای مثال خروجی عبارت `$\left\{{x^2 \over 3}\right.$` برابر $\left\{{x^2 \over 3}\right.$ خواهد بود.

علاوه بر تنظیم اندازه توسط پردازندهٔ متن، می‌توان اندازهٔ پرانتزها را به صورت دستی نیز تغییر داد. برای این کار کافی‌است عبارات `\big \Big \bigg \Bigg` را پیش از انواع پرانتزها به کار برد. برای مثال `\{` را در نظر گرفته و توجه کنید که خروجی عبارت `$\{ \big\{ \Big\{ \bigg\{ \Bigg\{$` برابر $\{ \big\{ \Big\{ \bigg\{ \Bigg\{$ است.

در «پیوست ۲» لیست مناسبی از پرانتزها موجود است.

حالت‌بندی‌ها

فرض کنید می‌خواهیم تابعی با حالات مختلف معرفی کنیم. مثلا تابع قدرمطلق را اینگونه تعریف می‌کنیم:

$$abs(x) = \begin{cases} x & x \ge ۰ \\ -x & x < ۰ \end{cases}$$

این فرمول نتیجهٔ عبارت زیر است.

```
$abs(x) =
\begin{cases}
 x & x \ge 0 \\
-x & x  <  0
\end{cases}$
```

برای نوشتن یک عبارت دارای حالت بندی ابتدا `\begin{cases} ... \end{cases}` را نوشته و حالت‌ها را در میان آن می‌گنجانیم. هر یک از سطرهای حالت بندی به شکل `value & condition \\` پیاده سازی می‌شود که `value` نقش مقدار تابع (عبارت سمت چپ) و `condition` نقش شرط حالت (عبارت سمت راست) را دارد. برای مثال در توصیف تابع قدر مطلق `x & x \ge 0 \\` به همین فرم است. برای سطر آخر حالت بندی می‌توان `\\` را ننوشت ولی در صورت نوشتن هم اتفاق بدی نخواهد افتاد. توجه کنید این عبارات را می‌توان در یک خط نیز نوشت. مثلاً عبارت مثال را می‌توان به شکل `$abs(x) = \begin{cases} x & x \ge 0 \\ -x & x  <  0 \end{cases}$` نیز پیاده‌سازی کرد.

فاصله‌ها

اگر دقت کرده‌باشید متوجه می‌شوید که هر مقدار استفاده از کاراکتر space تفاوتی در خروجی فرمول تک ندارد. برای مثال خروجی هر دو عبارت `$xy$` و `$x  y$` برابر xy است. پس سوال این است که چگونه فاصله ایجاد کنیم؟

یک راه دم دستی و البته مناسب این است که از تعدادی `\,` استفاده کنیم تا فاصلهٔ مورد نظر ما ساخته‌شود. برای مثال خروجی `$x \, y$` و `$x \,\,\, y$` برابر $x\,y$ و $x\,\,\,y$ خواهد بود.
برای اطلاع دقیق‌تر از فواصل به «پیوست ۳» مراجعه فرمایید.

نکات جانبی

گاهی وقت‌ها لازم است در میان فرمول ریاضی از عبارات معمول استفاده کنیم. در چنین مواردی می‌توان عبارات معمول را داخل `\text{...}` قرار دارد. برای مثال عبارت `\{x\in s\mid \text{ x is extra large }\}` به $\{x\in s\mid \text{ x is extra large }\}$ تبدیل می‌شود!

نکتهٔ دیگری که می‌توان به آن اشاره کرد این است که برای به دست آوردن علامت‌های $\backslash, \{, \$, _$ که در روش فرمول‌نویسی استفاده می‌شوند، می‌توان از عبارات `\backslash , \{ , \$ , \_` استفاده کرد.

ماتریس‌ها

برای نمایش یک ماتریس رویکردی کاملا مشابه با «حالت بندی‌ها» داریم.
ابتدا عبارت `\begin{matrix}...\end{matrix}` را می‌نویسیم. سپس سطرها را به شکلی که توضیح می‌دهیم وارد می‌کنیم. برای هر سطر ماتریس، عبارت `a & b & c & ... & z \\` را می‌نویسیم که منظور از `a, b, c, ..., z` درایه‌های سطر مورد نظر هستند. با انجام این عمل برای سطرهای اول، دوم تا آخر، ماتریس ما کامل می‌گردد. مانند قسمت «حالت بندی» می‌توان از \\ سطر آخر صرف نظر کرد. به عنوان مثال برای روش مذکور، مشاهده کنید ماتریس

$$\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}$$

توسط عبارت زیر قابل تولید است.

```
$\begin{matrix}
    a & b & c \\
    d & e & f \\
    g & h & i
\end{matrix}$
```

برای تعیین بدنهٔ ماتریس کافی‌است عبارت داخل `\begin,\end` را تغییر دهیم. برای مثال ماتریس $\begin{bmatrix} I & R \\ A & N \end{bmatrix}$ توسط عبارت زیر به دست می‌آید.

```
$\begin{bmatrix}
    I & R \\
    A & N
\end{bmatrix}$
```

انواع دیگر بدنه‌های ماتریس توسط عبارات `patrix` ، `Bmatrix` ، `vmatrix` و `Vmatrix` تعیین می‌شوند. در زیر فهرستی از انواع بدنه‌های مارتیس‌ها موجود است.

$\begin{matrix} I & R \\ A & N \end{matrix}$ `$\begin{matrix}I&R\\A&N\end{matrix}$` ،
$\begin{pmatrix} I & R \\ A & N \end{pmatrix}$ `$\begin{pmatrix}I&R\\A&N\end{pmatrix}$`
$\begin{Bmatrix} I & R \\ A & N \end{Bmatrix}$ `$\begin{Bmatrix}I&R\\A&N\end{Bmatrix}$` ،
$\begin{vmatrix} I & R \\ A & N \end{vmatrix}$ `$\begin{vmatrix}I&R\\A&N\end{vmatrix}$`
$\begin{Vmatrix} I & R \\ A & N \end{Vmatrix}$ `$\begin{Vmatrix}I&R\\A&N\end{Vmatrix}$` ،
$\begin{bmatrix} I & R \\ A & N \end{bmatrix}$ `$\begin{bmatrix}I&R\\A&N\end{bmatrix}$`

پیوست ۱. حروف یونانی

α= \alpha ، β= \beta ، γ= \gamma ، δ= \delta ، ϵ= \epsilon ، ε= \varepsilon ، ζ= \zeta ، η= \eta ،
θ= \theta ، ϑ= \vartheta ، ι= \iota ، κ= \kappa ، μ= \mu ، ν= \nu ، ξ= \xi ، π= \pi ، ϖ= \varpi ،
ρ= \rho ، ϱ= \varrho ، τ= \tau ، υ= \upsilon ، ϕ= \phi ، φ= \varphi ، ω= \omega
Γ= \Gamma ، Δ= \Delta ، Θ= \Theta ، Λ= \Lambda ، Ξ= \Xi ، Π= \Pi ، Σ= \Sigma ، Υ= \Upsilon ،
Φ= \Phi ، Ψ= \Psi ، Ω= \Omega

پیوست ۲. انواع پرانتزها

$($ (، $)$) ، $[$ [، $]$] ، $\{$ \{ ، $\}$ \} ، $|$ \textbar ، $\langle$ \langle ، $\rangle$ \rangle ، $\vert$ \vert ، $\Vert$ \Vert
$\lfloor$ \lfloor ، $\rfloor$ \rfloor ، $\lceil$ \lceil ، $\rceil$ \rceil ، $/$ / ، $\backslash$ \backslash ، $\uparrow$ \uparrow ، $\downarrow$ \downarrow
$\Uparrow$ \Uparrow ، $\Downarrow$ \Downarrow ، $\updownarrow$ \updownarrow ، $\Updownarrow$ \Updownarrow ، $/ / / / / \dots \| \| \| \| \|$
/ \big/ \Big/ \bigg/ \Bigg/ \dots \Bigg\Vert \bigg\Vert \Big\Vert \big\Vert \Vert

پیوست ۳. انواع فواصل

برای ایجاد فاصله در عبارات لاتک، دستورات بسیاری وجود دارد که به آنها می‌پردازیم. دستور \, فاصله‌ای به اندازهٔ حدود 3mu ایجاد می‌کند. برای مثال خروجی a\,b ، $a\,b$ خواهد بود. دستورهای \: و \> فاصله‌ای حدود 4mu ایجاد کرده و دستور \: فاصله‌ای حدود 5mu ایجاد می‌کند. دستور \! فاصله‌ای حدود 3mu- ایجاد می‌کند! (در واقع کم می‌کند.) دستور \enspace فاصله‌ای حدود 9mu=.5em ایجاد می‌کند. دستورهای \quad و \qquad به ترتیب فاصله‌هایی برابر 1em و 2em ایجاد می‌کنند. دستورات \kern <len> و \hskip <len> و \hspace{<len>} فاصله‌هایی به اندازهٔ <len> ایجاد می‌کنند. برای مثال خروجی $a \kern 1em b$ ، $a \kern 1em b$ خواهد بود. دستور \hphantom{<stuff>} طول <stuff> را محاسبه کرده و فاصله‌ای معادل با آن طول ایجاد می‌کند. دستور \ که در آن \ توسط space همراهی شده‌است، فاصله‌ای مصطلح به control space و دستور ~ فاصله‌ای مصطلح به non-breaking space (فاصلهٔ نشکن) را ایجاد می‌کند. می‌توانید فاصله‌های معرفی شده را در عمل در زیر مشاهده کنید.

ab ab ، $a\,b$ a\,b ، $a\ b$ a\ b ، $a\, b$ a\, b ، $a\; b$ a\; b ، $a\: b$ a\: b ، $a\> b$ a\> b ، $a\! b$ a\! b ، $a\enspace b$ a\enspace b
$a\quad b$ a\quad b ، $a\qquad b$ a\qquad b ، $a\kern 1.3em b$ a\kern 1.3em b ، $a\hskip .9em b$ a\hskip .9em b
$a\hspace{1.1em} b$ a\hspace{1.1em} b ، $a b$ a b ، $a\ b$ a\ b ، $a\~b$ a~b