

OpenOffice.org Math

完全教程

by

周桂地@青岛

email: zhouguidi@gmail.com

blog: <http://zhouguidi.blog.51cto.com/>

Jul, 2010

目录

1.简介.....	1
2.基本知识.....	2
2.1.公式的组成.....	2
2.2.复合元素.....	2
2.3.文本、空格、换行和注释.....	3
2.4.更改默认字体.....	3
3.运算符.....	3
3.1.基本运算.....	3
3.2.关系运算.....	4
3.3.集合运算.....	4
3.4.其他运算符.....	5
3.5.大型运算符.....	5
3.6.独立修饰符.....	5
3.7.依附修饰符.....	6
3.8.上下标.....	6
3.9.向量和矩阵.....	6
4.函数.....	6
4.1.内置函数.....	6
4.2.自定义函数.....	7
5.括号.....	7
6.内置常量及其他符号.....	8
7.自定义运算符.....	9
8.格式控制.....	9
9.示例.....	10

说明:

本教程为原创文章，请保持文章内容的完整性。允许自由分发，但请注明作者出处，否则将追究法律责任。

1. 简介

经常写公式的人，都用过各种公式编辑器，比如 Microsoft 的公式编辑器 MS 公式 3.0（Word 附带）、MathType 等。相比于 MS 公式 3.0，MathType 功能强大的多，也是很多人使用的公式编辑器。然而，MathType（以及 MS 公式 3.0）都是采用的都是“所见即所得”的操作方式，所有的输入都要依赖鼠标在公式编辑器中点击选择各种公式元素来完成。如果只输入几个简单的公式，这没什么问题。然而，如果你面临的是一篇充满公式的科技论文，甚至是一本数学书，那么我想不管是谁也会发抖的。虽然可以使用内置的许多快捷键来加速操作，但要快速准确的按下这么多快捷键也不是一件容易的事。而且，还有一个重要的问题——这些软件都是商业的，你需要向软件开发者支付一笔价格不菲的费用。当然，除非你觉得版权问题无所谓，从而心安理得的使用破解软件。

除了微软的 Office 软件之外，Oracle（以前是 Sun）的 OpenOffice.org 应该是最流行的一种办公软件了。它功能上的强大并不亚于 MS Office，而且是开源软件，你可以自由获取使用它，甚至如果可能，你可以修改并重新发布它。OpenOffice.org 于 MS Office 同样属于所见即所得的编辑模式，然而它的组件之一——OpenOffice.org Math 公式编辑器却采用“所想即所得”的方式。使用它，你不需要频繁的切换于键盘和鼠标之间，可以不受任何打扰地将存在于你头脑中的公式像流水一样的输入到文档中。这是通过 Math 使用的公式解释语言来完成的。只要你掌握这种简单高效的语言，你就能将你的公式像说一段自然语言一样“说”出来。Math 提供的各种公式元素和公式控制符号能够让你轻松输入极其复杂的公式。事实上，OpenOffice.org Math 也提供了类似 MS 公式 3.0 和 MathType 的依靠鼠标点选来输入的操作方式，以提供一个直观的输入感受，使习惯 MS 公式 3.0 或 MathType 的用户能够继续使用其熟悉的输入方式并逐渐过渡。

另外，MS Office 和 OpenOffice.org 之外，LaTeX 也是使用十分广泛的排版工具。它完全采用所想即所得的理念，一篇文档的任何输入、编辑、格式控制都是通过其命令语言来完成的，当然也包括公式。与 MS Office 和 LaTeX 相比，OpenOffice.org 可以说是一种折衷：其文档编辑（OpenOffice.org Writer）是所见即所得的，而公式编辑是所想即所得的。这使得熟悉 MS Office 的人可以快速转移到 OpenOffice.org，而不需要去学习大量的命令；同时又使公式的编辑变得十分高效。同时，有一个 OOoLatex 插件可以让你在 OOo 中输入 LaTeX 风格的公式。

Math 既是 OpenOffice.org Writer 的一部分，也是一个单独的程序。你可以单独运行 Math，编辑公式，然后存为 odf 格式，然后插入到 Writer 中；也可以在 Writer 里通过“插入”-“对象”-“公式”菜单来插入一个新公式并编辑它。

这篇文章就来详细介绍一下 OpenOffice.org Math 的使用，让我们一起来享受输入公式的乐趣吧～因为我也是刚学，所以里面出现什么问题也是很正常的，发现什么问题可以给我发邮件，或在博客上留言。文章写的比较系统，所以有些地方显得有点细琐，不过我相信耐心看完的话（毕竟才这么几页，呵呵）一定会对 Math 达到比较熟练的程度的。主要参考资料是 OpenOffice.org Math 帮助文档和陈珂写的《OpenOffice.org 排版指南》，
<http://forum.ubuntu.org.cn/viewtopic.php?f=35&t=136728>，表示感谢。

2. 基本知识

2.1. 公式的组成

Math 采用一种公式解释语言来描述公式。这种语言由各种运算符、保留字、函数、变量、常量和文本组成。运算符、函数是预定义的，用来表示各种运算；保留字是用于控制公式显示方式的，它们自身是不可见的；变量是出现在公式中的自变量、因变量；常量一般是指数字；文本是区别于前面各种公式组成成分的，它用于在公式中显示一些纯文本。另外，Math 允许用户自定义运算符和函数。各种公式成分往往具有不同的外观，如函数、常量和文本一般显示为正体，变量一般显示为斜体。这也是可以自定义的。

在这篇文章里，我们将用 $\langle ? \rangle$ 表示任意一个变量、常量或文本，并将它们称为公式元素。

Math 将任何连续的字母（英文字母和希腊字母）、数字视为一个公式元素。公式元素不能包含空格和运算符。比如

$a, abc, a1, 123, 12a$

都是合法的公式元素。而

$a\ b, a+b, a(b$

等将被视为两个公式元素。

Math 中希腊字母用 %+字母名称表示，比如 %alpha= α ，%rho= ρ ，%DELTA= Δ ，%THETA= Θ 。

现在我们可以输入简单的公式了，比如

$z=x+y, z=2x+y, z=2(x+y)$

等，都是很简单的。

2.2. 复合元素

有些时候需要将多个公式元素一起当作一个来使用，比如下面的公式

$x_{i=1}$ ，

“i=1”是一起作为 x 的下标出现的。如果输入

$x_i=1$

（_是下标运算符）将会出现这样的结果：

$x_i=1$

因为 i=1 是两个元素，而下标运算符_只能将其后的第一个元素作为下标。此时需要复合运算符。

复合运算符是用花括号 {} 来定义的。任何放在花括号里的公式成分都将作为一个单独的公式元素来使用，而不管其内部有多复杂。因此上面的公式应该表示为

$x_{\{i=1\}}$ 。

公式元素也可以嵌套，比如

$x^{\{e_1^2\}}$

代表公式

$x^{e_1^2}$ ，

其中^是上标（或乘幂）运算符。

下面我们提到公式元素时，都是既包括单一元素，也包括复合元素的。

2.3. 文本、空格、换行和注释

默认情况下，除了常量之外，任何公式元素都是作为变量的。如果想将一个元素设为文本（比如为了使之变成正体，而不是斜体），则可以用一对双引号将其包围。比如微分算子

$$\frac{dy}{dx} == \{d y\} \text{ over } \{d x\}$$

（over 是分式运算符），一般情况下微分中的“d”都是正体的，而 Math 把“d”也作为变量了，变量的默认字体是斜体的。只要改成

$$\{“d” y\} \text{ over } \{“d” x\}$$

就能使“d”成为文本，从而使用文本的字体（一般是正体），即

$$\frac{dy}{dx}。$$

这里有个小问题，就是分子和分母都没有居中对齐，而是左对齐的，这样有些不好看。解决的办法我们下面会提到。

前面提过，空格是用来分隔元素的，除了这个作用之外，空格别无其他作用。在其他空格或运算符已经将元素分开了的情况下，多余的空格都会被忽略。那么如果要在公式中加入空格怎么办呢？可以用“`”（键盘上 Tab 上面那个键）或“~”（shift+`）。` 是小空格，~ 是大空格。它们都属于公式元素的一种，可以理解为特殊的内置变量。另外还可以用空元素 {} 来插入空格，它相当于`，也是小空格。

另外，Math 的公式默认都是在一行显示的，即便你输入了回车符也是一样。因此一行公式可以分几行输入，这样可以增加代码的可读性。如果输入分几行显示的公式，需要用 newline 保留字，它把其后的公式放在下一行显示。

为了增加代码的可读性，也可以在代码中加入注释。注释使用 %% 定义，并一直作用到一行代码的结束，因此如果使用注释，必须分行写代码了。如果要在公式中显示 % 和 %%，可以用双引号将之变为文本。

2.4. 更改默认字体

默认的字体在那里改呢？点击“格式”菜单下的“字体”子菜单，将弹出“字体”对话框，里面列出了各种公式元素使用的字体。点击右边的“更改”按钮可以改变默认字体。

3. 运算符

好了，现在我们做好了准备，来看一下 Math 如何输入基本运算符号吧。下面给出了各种运算符的用法，每个运算符都给有写法、作用、和一个例子。以第二个运算符为例，<?> + <?> 是它的写法，它的作用是用作加号，a 代表第一个 <?> 表示的公式元素，b 代表第二个。

$a+b$ 是公式的显示结果。较复杂的公式后面还给出了代码。

3.1. 基本运算

+<?> 正号 $+a$

<?> + <?> 加号 $a+b$

-<?> 负号 $-a$

<?> - <?> 减号 $a-b$

~+<?> 可负可正 $\mp a$

+~<?> 可正可负 $\pm a$

neg <?> 逻辑非 $\neg a$

$\langle ? \rangle *$ $\langle ? \rangle$ 乘号 (星) $a * b$
 $\langle ? \rangle \cdot$ $\langle ? \rangle$ 乘号 (点) $a \cdot b$
 $\langle ? \rangle \times$ $\langle ? \rangle$ 乘号 (叉) $a \times b$
 $\langle ? \rangle \text{ and 或 } \langle ? \rangle$ 逻辑与 $a \wedge b$
 $\langle ? \rangle \div$ $\langle ? \rangle$ 除法 $a \div b$
 $\langle ? \rangle /$ $\langle ? \rangle$ 除法 (/) a / b
 $\langle ? \rangle \text{ over } \langle ? \rangle$ 除法 (分式) $\frac{a}{b}$
 $\langle ? \rangle \text{ or 或 } \langle ? \rangle$ 逻辑或 $a \vee b$
 $\langle ? \rangle \circ$ $\langle ? \rangle$ 连结 (圆圈) $a \circ b$

3.2. 关系运算

$\langle ? \rangle =$ $\langle ? \rangle$ 等于 $a = b$
 $\langle ? \rangle \neq$ $\langle ? \rangle$ 或 $\text{neq } \langle ? \rangle$ 不等于 $a \neq b$
 $\langle ? \rangle \approx$ $\langle ? \rangle$ 约等于 $a \approx b$
 $\langle ? \rangle \text{ divides } \langle ? \rangle$ 可被除于 (竖线) $a | b$
 $\langle ? \rangle \text{ ndivides } \langle ? \rangle$ 不可被除于 $a \nmid b$
 $\langle ? \rangle <$ $\langle ? \rangle$ 或 $\text{lt } \langle ? \rangle$ 小于 $a < b$
 $\langle ? \rangle >$ $\langle ? \rangle$ 或 $\text{gt } \langle ? \rangle$ 大于 $a > b$
 $\langle ? \rangle \leq$ $\langle ? \rangle$ 或 $\text{leslant } \langle ? \rangle$ 小于等于 $a \leq b$
 $\langle ? \rangle \geq$ $\langle ? \rangle$ 或 $\text{geslant } \langle ? \rangle$ 大于等于 $a \geq b$
 $\langle ? \rangle \leq$ $\langle ? \rangle$ 或 $\text{le } \langle ? \rangle$ 小于等于 (等于线是平的) $a \leq b$
 $\langle ? \rangle \geq$ $\langle ? \rangle$ 或 $\text{ge } \langle ? \rangle$ 大于等于 (等于线是平的) $a \geq b$
 $\langle ? \rangle \ll$ $\langle ? \rangle$ 或 $\text{ll } \langle ? \rangle$ 远小于 $a \ll b$
 $\langle ? \rangle \gg$ $\langle ? \rangle$ 或 $\text{gg } \langle ? \rangle$ 远大于 $a \gg b$
 $\langle ? \rangle \stackrel{\text{def}}{=}$ $\langle ? \rangle$ 定义为 $a \stackrel{\text{def}}{=} b$
 $\langle ? \rangle \simeq$ $\langle ? \rangle$ 相似等于 (波浪线下一条直线) $a \simeq b$
 $\langle ? \rangle \sim$ $\langle ? \rangle$ 相似于 (波浪线) $a \sim b$
 $\langle ? \rangle \propto$ $\langle ? \rangle$ 正比于 $a \propto b$
 $\langle ? \rangle \parallel$ $\langle ? \rangle$ 平行于 (双竖线) $a \parallel b$
 $\langle ? \rangle \perp$ $\langle ? \rangle$ 垂直于 $a \perp b$
 $\langle ? \rangle \equiv$ $\langle ? \rangle$ 恒等于 $a \equiv b$
 $\langle ? \rangle \rightarrow$ $\langle ? \rangle$ 趋近于 (箭头) $a \rightarrow b$
 $\langle ? \rangle \Leftarrow$ $\langle ? \rangle$ 左向双箭头 $a \Leftarrow b$
 $\langle ? \rangle \Leftrightarrow$ $\langle ? \rangle$ 双向双箭头 $a \Leftrightarrow b$
 $\langle ? \rangle \Rightarrow$ $\langle ? \rangle$ 右向双箭头 $a \Rightarrow b$
 $\langle ? \rangle \leftrightarrow$ $\langle ? \rangle$ 对应符号图像 (实心点和空心点相连) $a \leftrightarrow b$
 $\langle ? \rangle \rightharpoonup$ $\langle ? \rangle$ 原始对应符号 (空心点和实心点相连) $a \rightharpoonup b$

3.3. 集合运算

$\langle ? \rangle \in$ $\langle ? \rangle$ 属于 $a \in b$
 $\langle ? \rangle \notin$ $\langle ? \rangle$ 不属于 $a \notin b$
 $\langle ? \rangle \ni$ $\langle ? \rangle$ 含有 (反向属于) $a \ni b$
 $\langle ? \rangle \cap$ $\langle ? \rangle$ 交集 $a \cap b$

$\langle ? \rangle$ union $\langle ? \rangle$ 并集 $a \cup b$
 $\langle ? \rangle$ setminus 或 bslash $\langle ? \rangle$ 差集 $a \setminus b$
 $\langle ? \rangle$ slash $\langle ? \rangle$ 商数集 a/b
 $\langle ? \rangle$ subset $\langle ? \rangle$ 包含于 (真子集) $a \subset b$
 $\langle ? \rangle$ subseteq $\langle ? \rangle$ 包含于 (子集) $a \subseteq b$
 $\langle ? \rangle$ supset $\langle ? \rangle$ 包含 (真父集) $a \supset b$
 $\langle ? \rangle$ supseteq $\langle ? \rangle$ 包含 (父集) $a \supseteq b$
 $\langle ? \rangle$ nsubset $\langle ? \rangle$ 不包含于 (非子集) $a \not\subset b$
 $\langle ? \rangle$ nsubseteq $\langle ? \rangle$ 不包含于 (非子集, 或等集) $a \not\subseteq b$
 $\langle ? \rangle$ nsupset $\langle ? \rangle$ 不包含 (非父集) $a \not\supset b$
 $\langle ? \rangle$ nsupseteq $\langle ? \rangle$ 不包含 (非父集, 或等集) $a \not\supseteq b$

3.4. 其他运算符

$\langle ? \rangle$ oplus $\langle ? \rangle$ 圆圈加号 $a \oplus b$
 $\langle ? \rangle$ ominus $\langle ? \rangle$ 圆圈减号 $a \ominus b$
 $\langle ? \rangle$ odot $\langle ? \rangle$ 圆圈点乘号 $a \odot b$
 $\langle ? \rangle$ otimes $\langle ? \rangle$ 圆圈叉乘号 $a \otimes b$
 $\langle ? \rangle$ odivide $\langle ? \rangle$ 圆圈除号 $a \oslash b$
 $\langle ? \rangle$ wdiv $\langle ? \rangle$ 大斜杠除号 a/b
 $\langle ? \rangle$ wdiv $\langle ? \rangle$ 大反斜杠除号 $a \backslash b$

3.5. 大型运算符

$\lim \langle ? \rangle$ 极限 $\lim a$
 $\sum \langle ? \rangle$ 求和 $\sum a$
 $\prod \langle ? \rangle$ 连乘 $\prod a$
 $\coprod \langle ? \rangle$ 合数 $\coprod a$
 $\int \langle ? \rangle$ 积分 $\int a$
 $\iint \langle ? \rangle$ 二重积分 $\iint a$
 $\iiint \langle ? \rangle$ 三重积分 $\iiint a$
 $\oint \langle ? \rangle$ 曲线积分 $\oint a$
 $\oint \langle ? \rangle$ 二重曲线积分 $\oint a$
 $\oint \langle ? \rangle$ 三重曲线积分 $\oint a$

3.6. 独立修饰符

这些修饰符是可以单独使用的, 用于对公式元素进行修饰。

$\bar{}$ $\langle ? \rangle$ 上短线 (一声) \bar{a}
 $\acute{}$ $\langle ? \rangle$ 右上重音 (二声) \acute{a}
 $\check{}$ $\langle ? \rangle$ 反向抑扬符号 (三声) \check{a}
 $\grave{}$ $\langle ? \rangle$ 右下重音 (四声) \grave{a}
 $\mathring{}$ $\langle ? \rangle$ 上加圈 \mathring{a}
 $\tilde{}$ $\langle ? \rangle$ 波浪线 \tilde{a}

hat <?> 抑扬符号 (尖号) \hat{a}
 dot <?> 上加点 (导数) \dot{a}
 ddot <?> 上加双点 \ddot{a}
 dddot <?> 上加三点 \dddot{a}
 breve <?> 短音符号 \breve{a}
 vec <?> 矢量箭头 \vec{a}
 widevec <?> 宽矢量箭头 \overrightarrow{abc}
 widetilde <?> 宽波浪线 \widetilde{abc}
 widehat <?> 宽尖号 \widehat{abc}
 overline <?> 上划线 \overline{abc}
 underline <?> 下划线 \underline{abc}
 overstrike <?> 删除线 \overline{abc}

3.7. 依附修饰符

这些修饰符不能单独使用，必须依附于其他运算符，用来指明其他运算符的上限下限。

from <?> 下限 \int_a^x (int from a x)

to <?> 上限 \int^a_x (int to a x)

from <?> to <?> 上下限 $\int_0^1 x$ (int from 0 to 1 x)

3.8. 上下标

<?> ^或 sup <?> 上标 a^b

<?> _或 sub <?> 下标 a_b

<?> lsup <?> 左上标 ba

<?> lsub <?> 左下标 $_ba$

<?> csup <?> 顶标 $\overset{b}{a}$

<?> csub <?> 底标 $\underset{b}{a}$

3.9. 向量和矩阵

binom <?> <?> 垂直编排两个元素 $\begin{matrix} a \\ b \end{matrix}$

stack {<?> # <?> # <?>} 垂直编排三个元素 $\begin{matrix} a \\ b \\ c \end{matrix}$

matrix {<?> # <?> ## <?> # <?>} 矩阵 $\begin{matrix} a & b \\ c & d \end{matrix}$, ##用于换行, #用于分隔同行的元素

4. 函数

4.1. 内置函数

Math 有一些内置函数，当输入它们时默认是正体显示的。下面列出了这些函数。

func $e^{<?>}$ 指数函数（上标形式） e^a
 ln(<?>) 自然对数 $\ln(a)$
 exp(<?>) 指数 $\exp(a)$
 log(<?>) 对数 $\log(a)$
 sin(<?>) 正弦 $\sin(a)$
 cos(<?>) 余弦 $\cos(a)$
 tan(<?>) 正切 $\tan(a)$
 cot(<?>) 余切 $\cot(a)$
 arcsin(<?>) 反正弦 $\arcsin(a)$
 arccos(<?>) 反余弦 $\arccos(a)$
 arctan(<?>) 反正切 $\arctan(a)$
 arccot(<?>) 反余切 $\operatorname{arccot}(a)$
 sinh(<?>) 双曲正弦 $\sinh(a)$
 cosh(<?>) 双曲余弦 $\cosh(a)$
 tanh(<?>) 双曲正切 $\tanh(a)$
 coth(<?>) 双曲余切 $\coth(a)$
 arsinh(<?>) 反双曲正弦 $\operatorname{arsinh}(a)$
 arcosh(<?>) 反双曲余弦 $\operatorname{arcosh}(a)$
 artanh(<?>) 反双曲正切 $\operatorname{artanh}(a)$
 arcoth(<?>) 反双曲余切 $\operatorname{arcoth}(a)$
 <?> ^或 sup <?> 乘幂 a^b
 sqrt <?> 开方 \sqrt{a}
 nroot <?> <?> n 次方根 $\sqrt[n]{b}$
 abs <?> 绝对值 $|a|$
 fact <?> 阶乘 $a!$

上面函数中的括号不是必须的。例如 $\sin(\%alpha)$ 将显示为 $\sin(\alpha)$ ，而 $\sin \%alpha$ 显示为 $\sin \alpha$ ，他们都是函数。

4.2. 自定义函数

上面列出的第一个内置函数写法为 $\text{func } e^{<?>}$ ，其实这就是一个自定义函数。自定义函数以保留字 `func` 开头，它将后续的公式元素设为函数。可以用这种方式定义任何函数。

5. 括号

Math 预定义的各种括号：

(<?>) 圆括号 (a)
 [<?>] 方括号 $[a]$
 ldbracket <?> rdbracket 双重方括号 $\llbracket a \rrbracket$
 lline <?> rline 左右单线（模） $|a|$
 ldline <?> rdline 左右双线（范数） $\|a\|$
 lbrace <?> rbrace 花括号 $\{a\}$
 langle <?> rangle 角括号 $\langle a \rangle$
 langle <?> mline <?> rangle 运算符括号 $\langle a|b \rangle$

lfloor <?> rfloor 带下檐的左右线 $\lfloor a \rfloor$

lceil <?> rceil 带上檐的左右线 $\lceil a \rceil$

这些括号都是固定高度的，也就是说，不论他们括起来的内容有多高，它们总显示为一个字符高度。有时候这并不美观，我们想让括号高度随内容变化怎么办呢？别急，Math 提供了 left ... right ... 保留字将各种括号变为可变大小的。只要在左括号前加上 left 保留字，在右括号前加上 right 关键字，就能实现这一点。比如公式

$$\left[\sum_{n=1}^N x_n \right]$$

代码为

left lbracket sum from n=1 to N x_n right rbracket.

如果只想使用一半括号，例如下面的方程组

$$\begin{cases} y=2x+1 \\ y=x+3 \end{cases}$$

该怎么做呢？如果写成 left lbrace binom {y=2x+1} {alignl y=x+3}，是不能正常显示的，因为 left 和 right 保留字必须配对。此时 none 保留字就很有用了。用 none 来代替 right 后面本来应该出现的 rbrace，就可以使右半括号不显示。即 left lbrace binom {y=2x+1} {alignl y=x+3} right none。

有时候想将各种括号变成单独的字符来使用，这可以简单地通过在括号前加一个\来完成。也就是说，\ (、\ lbrace 等代表一个普通的字符，它们可以像其他任何字符一样来使用，也具有与其他字符一样的高度。但是只有不加 left 和 right 保留字的时候才能用这种方式。

此外还有两个垂直方向的括号：

<?> overbrace <?> 上花括号 \overbrace{a}^b

<?> underbrace <?> 下花括号 \underbrace{a}_b ，

一般用于表示推导过程。

6. 内置常量及其他符号

内置常量包括：

emptyset 空集 \emptyset

aleph aleph \aleph

setN 自然数集 \mathbb{N}

setZ 整数集 \mathbb{Z}

setQ 有理数集 \mathbb{Q}

setR 实数集 \mathbb{R}

setC 复数集 \mathbb{C}

infinity 或 infity 无穷大 ∞

其他符号：

partial <?> 偏微分 ∂a

nabla 哈密尔顿算子（梯度） ∇

exists 存在 \exists

forall 对任意 \forall

hbar 带横线的 h \hbar

lambdabar 带横线的 lambda $\bar{\lambda}$

re 复数的实部 \Re
im 复数的虚部 \Im
wp p 函数 (Weierstrass p) \wp
backepsilon 反向 epsilon \ni
leftarrow 左箭头 \leftarrow
rightarrow 右箭头 \rightarrow
uparrow 上箭头 \uparrow
downarrow 下箭头 \downarrow
dotslow 下三点 \dots
dotsaxis 中三点 \cdots
dotsvert 竖三点 \vdots
dotsup 左下-右上三点 \cdotp
dotsdown 右上-左下三点 \cdotp
newline 换行
` 小空格
~ 大空格

7. 自定义运算符

Math 提供了用户自定义运算符的功能，主要通过下面三个保留字实现：

oper %... 用户定义运算符

uoper %... <?> 用户定义一元运算符

<?> boper %... <?> 用户定义二元运算符

其中%...部分就是用户自定义的运算符，可以当作一元和二元运算符来使用。在公式中使用特殊字符时此功能将十分有用，可以用来插入不存在于默认 OpenOffice.org 字符集中的字符。

例如下面的公式：

$$\sum_{i=1}^n \text{☎} x_i$$

采用了一个特殊字符（电话）。它是这样实现的：点击“工具”目录下的“图标”，在弹出的对话框中选择“特殊字符”作为符号集，然后单击“编辑”按钮，在下一个对话框中再次选择“特殊字符”作为符号集，选择一个符号并点击，然后在“图标”文本框中输入一个名称（例如 telephone）。单击“新增”，然后单击“确定”、“关闭”。此时就可以在代码中通过 oper %telephone 来使用这个符号。

8. 格式控制

下面这些关键字用于控制公式元素的格式：

color <color> <?> 指定字符颜色，<color>处为颜色名称*

font <?> 设置字体，处为字体名称**

bold <?> 加粗 **a**

nbold <?> 不加粗

ital <?> 斜体 *a*

nitalic <?> 不倾斜 a

size n 或+n 或-n 或*n 或/n <?> 大小设为 n 磅，或增大/缩小 n 磅，或变大/变小 n 倍

phantom <?> 透明字符，只占位置而不显示

alignl <?> 左对齐***

alignr <?> 右对齐***

alignc <?> 居中对齐***

* <color>可为“black”、“white”、“cyan”、“magenta”、“red”、“blue”、“green”或“yellow”之一。

** 关于字体，font <?>中的可以是“serif”、“sans”或“fixed”（即衬线字体、无衬线字体和固定字体）。这三类字体分别与系统中安装的一种字体对应，并且是可以自行设定的。单击“格式”菜单下的“字体”，在弹出的对话框下方有这三种字体目前对应于哪种字体。可以单击“更改”来改变这种对应关系，从而允许在公式中使用系统中的任意一种字体。

*** 使用 alignc、alignl 和 alignr时要注意，它们只能出现在一个表达式的开头。也就是说，表达式是作为调整对齐方式的最小单位的。例如 a+alignr b 是错误的，而 a+b alignr c 是允许的。而且多个 align 命令之间是相互影响的。例如 {alignl{alignr a}}over{b+c} 会导致 a 向右对齐。同时，如果一行或一个表达式以文本开头，那么默认是左对齐的。可以利用这个特性，在公式或表达式开头加上一个空文本（空的双引号）来使之变成左对齐。

还记得前面我们说过的一个小问题吗？就是微分算子的对齐问题。当时我们采用的是文本的方式使微分算子中的“d”变为正体。但是我们刚刚提到了，这样一来相应的表达式就是以文本开头了，那么 Math 将默认把它显示为左对齐。不过不要紧，现在运用这一章的新内容就可以轻松改变这一现象了。我想到了两种方法，一种是用 nitalic 保留字加在“d”前面，另一种是用 alignc 保留字直接指定对齐方式。试试看吧：

$$\frac{dy}{dx} \quad (\{nitalic d y\} \text{ over } \{nitalic d x\}), \quad \frac{dy}{dx} \quad (\{alignc "d"y\} \text{ over } \{alignc "d"x\})$$

果然成功了吧～～

9. 示例

好了，到现在为止，你已经能轻松写出漂亮的公式了。随着使用 Math 的时间越来越长，你就会越来越喜欢并享受这种流畅的公式输入体验！

下面来看些例子吧，作为一种回顾和一种练习吧，同时也是这篇教程的结束～

（这些例子来自 OpenOffice.org Math 帮助文档）

$$D_{mn}^{(2)} \quad (D_mn^{\wedge} \text{ size } /2 \text{ left(} 3 \text{ over } 2 \text{ right))}$$

$${}^3\Sigma_g^+ \quad (\%SIGMA_g^{\wedge}\{\{\}+\{\}\}\text{lsup } 3)$$

$$\Phi_{k_1 k_2 \dots k_n}^{i_1 i_2 \dots i_n} \quad (\%PHI^{\wedge}\{i_1 i_2 \text{ dotsaxis } i_n\}_{k_1 k_2 \text{ dotsaxis } k_n})$$

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & & & A_{2n} \\ \vdots & & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

$$(\text{ font sans bold size } *2 \text{ A } = \text{left[matrix}\{A_11\#A_12\#\text{dotsaxis}\#A_1n\}\#\#A_21\#\{\}\#\#A_2n\}\#\#\text{dotsvert}\#\{\}\#\#\text{dotsvert}\#\#A_n1\}\#A_n2\}\#\text{dotsaxis}\#A_nn\}\text{right}])$$

$$G_{x_m x_n}^{(\alpha, \beta)} = \begin{bmatrix} \arctan(\alpha) & \arctan(\beta) \\ x_m + x_n & x_m - x_n \end{bmatrix}$$

$$(\text{func } G^{\wedge}\{(\%alpha" , " \%beta)\}_{x_m x_n} = \text{left[matrix } \{ \arctan(\%alpha) \# \arctan(\%beta)$$

x_m + x_n # x_m - x_n }right])

$$f(\boldsymbol{x}, \boldsymbol{y}) = \left[\begin{array}{c} \boldsymbol{x} + \frac{\boldsymbol{y}}{z} + \left\{ \begin{array}{ccc} 2 & 3 & 4 \\ 4 & 5 & 6 \\ 6 & 7 & 8 \end{array} \right\} \\ \frac{\boldsymbol{y} + \sin(\boldsymbol{x})}{\alpha} \\ z + \frac{\boldsymbol{y}}{\boldsymbol{g}} \end{array} \right]$$

(bold { f(x", "y) = left [stack { x + y over z + left lbrace matrix { 2 # 3 # 4 ## 4 # 5 # 6 ## 6 # 7 # 8 } right rbrace # {y + sin (x)} over %alpha # z + y over g } right]})

$$f(x,y)=\frac{x\sin x\,\tan y}{\cos x}\quad (\text{func } f(x", "y)=\{x\sin x\sim\tan y\}\text{ over }\{\cos x\})$$

$$\Lambda_{deg,t}=1+\alpha_{deg}\sqrt{\frac{M_t}{M_{(t=0)}}-1}\quad.$$

(%LAMBDA_{deg","t}=1+%alpha_deg Sqrt {M_t over M_{(t=0)}}-1}\sim".)

$$f(t)=\int\limits_0^1\left[g(t')+\sum_{i=1}^Nh_i(t')\right]$$

(f(t)=int from size*1.5 0 to 1 left[g(t')+sum from i=1 to N h_i(t')right])

$$\rho(\boldsymbol{q},\omega)=\int e^{i\omega t}\rho(\boldsymbol{q},t)dt$$

(%rho(font sans bold q", "%omega) = int func e^{i %omega t}%rho(font sans bold q", "t)"d"t)

(这些例子来自陈珂写的《 OpenOffice.org 排版指北 》 , <http://forum.ubuntu.org.cn/viewtopic.php?f=35&t=136728>)

$$\begin{array}{l} \iint\limits_b\sqrt[3]{f(x,y)}=\sum\limits_b\sqrt[3]{f(x,y)} \\ \iint\limits_a\sqrt[3]{f(x,y)}=\sum\limits_a\sqrt[3]{f(x,y)} \end{array}$$

(iint nroot{3} {f(x,y)} = sum {nroot{3} {f(x,y)}}\} newline iint from a to b nroot{3}

{f(x,y)} =sum from a to b {nroot{3} {f(x,y)}}\})

$$\frac{\partial^2 f(x,y)}{\partial^2 x}+\frac{\partial^2 f(x,y)}{\partial x\partial y}+\frac{\partial^2 f(x,y)}{\partial^2 y}=0$$

{\partial^2 f(x,y)} over {\partial^2 x} + {\partial^2 f(x,y)} over {\partial x \partial y} + {\partial^2 f(x,y)} over {\partial^2 y} = 0)

$$a=12$$

$$b+c+d=b^2-2$$

$$a+b-d=c^3-5$$

(stack{ alignr a={}} # alignr b+c+d={} # alignr a+b-d={}} stack{ alignl 12 # alignl b^2-2 # alignl c^3-5})

$$\left[\begin{array}{cc} (\lambda+2\mu)\frac{\partial^2}{\partial^2 x}+\mu\frac{\partial^2}{\partial^2 z} & (\lambda+\mu)\frac{\partial^2}{\partial x\partial z} \\ (\lambda+\mu)\frac{\partial^2}{\partial x\partial z} & \mu\frac{\partial^2}{\partial^2 x}+(\lambda+2\mu)\frac{\partial^2}{\partial^2 z} \end{array} \right] \left(\begin{array}{c} u_x \\ u_y \end{array} \right) = \rho \left(\begin{array}{c} \ddot{u}_x \\ \ddot{u}_z \end{array} \right)$$

$$\left(\left[\text{stack} \left\{ \left\{ (\lambda + 2\mu) \frac{\partial^2}{\partial x^2} + \mu \frac{\partial^2}{\partial z^2} \right\} \# \left\{ (\lambda + \mu) \frac{\partial^2}{\partial x \partial z} \right\} \right\} \sim \text{stack} \left\{ \left\{ (\lambda + \mu) \frac{\partial^2}{\partial x \partial z} \right\} \# \left\{ \mu \frac{\partial^2}{\partial x^2} + (\lambda + 2\mu) \frac{\partial^2}{\partial z^2} \right\} \right\} \right] \text{left lbrace stack} \{ u_x \# u_y \} \text{right rbrace} = \rho$$

$$\text{left lbrace stack} \{ \ddot{u}_x \# \ddot{u}_z \} \text{right rbrace} \right)$$

$$\begin{bmatrix} a & b \\ a+b & c+d \end{bmatrix} \begin{bmatrix} \Gamma \\ X \end{bmatrix} = \begin{bmatrix} x_2^2 \\ y_2^2 \end{bmatrix}$$

$$\left(\left[\text{matrix} \{ a \# b \# a+b \# c+d \} \right] \text{left lbrace matrix} \{ \Gamma \# \chi \} \right) \text{right rbrace} = \text{left lbrace matrix} \{ x_2^2 \# y_2^2 \} \text{right rbrace}$$