LaTeX Workshop

An Introduction to LaTeX

Rakesh Jana
j.rakesh@iitg.ernet.in

Date: 2017/09/09

Research Scholar
Department of Mathematics
IIT Guwhati

Overview

1. Mathematical Expression

Mathematical Expression

Introduction

- Basic equations in LaTeX can be easily "programmed".

Introduction

- Basic equations in LaTeX can be easily "programmed".

We know Pythagorean theorem $x^2 + y^2 = z^2$ was proved to be invalid for other exponents, that is, following equation has no integer solutions:

$$x^n + y^n = z^n, \quad \forall n \geq 3$$

Introduction

- Basic equations in LaTeX can be easily "programmed".

We know Pythagorean theorem `$x^2 + y^2 = z^2$` was proved to be invalid for other exponents, that is, following equation has no integer solutions:

```
\[ x^n + y^n = z^n, \quad \forall n\geq 3 \]
```

We know Pythagorean theorem $x^2 + y^2 = z^2$ was proved to be invalid for other exponents, that is, following equation has no integer solutions:

$$x^n + y^n = z^n, \quad \forall n \geq 3$$

Introduction

- Basic equations in LaTeX can be easily "programmed".

The mass-energy equivalence is described by the famous equation

$$E = mc^2$$

discovered in 1905 by Albert Einstein. In natural units ($c = 1$), the formula expresses the identity

$$E = m \tag{1}$$

Introduction

- Basic equations in LaTeX can be easily "programmed".

We know Pythagorean theorem `$x^2 + y^2 = z^2$` was proved to be invalid for other exponents, that is, following equation has no integer solutions:

```
\[ x^n + y^n = z^n, \quad \forall n\geq 3 \]
```

The mass-energy equivalence is described by the famous equation

$$E = mc^2$$

discovered in 1905 by Albert Einstein. In natural units ($c = 1$), the formula expresses the identity

$$E = m \tag{1}$$

Allignment

Put equations on a separate line with the `equation*` environment.

When you will put `equation*` no equation number will appear

$$e^{\pi i} + 1 = 0$$

```
\begin{equation*}
e^{\pi i} + 1 = 0
\end{equation*}
```

Allignment

Put equations on a separate line with the `equation*` environment.
When you will put `equation*` no equation number will appear

$$e^{\pi i} + 1 = 0$$

```
\begin{equation*}
e^{\pi i} + 1 = 0
\end{equation*}
```

You can break into multiple lines.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

```
\begin{multline*}
  \sin (x)=x-\frac{x^3}{3!} \\
      +\frac{x^5}{5!}-\cdots
\end{multline*}
```

Allignment

Align using the `eqnarray*` environment

$$
\begin{aligned}
z_0 &= d = 0 \\
z_{n+1} &= z_n^2 + c
\end{aligned}
$$

```
\begin{eqnarray*}
  z_0 &=& d = 0 \\
  z_{n+1} &=& z_n^2+c
\end{eqnarray*}
```

(you can have an empty left or right side of the alignment).

Allignment

Align using the `eqnarray*` environment

$$\begin{aligned} z_0 &= d = 0 \\ z_{n+1} &= z_n^2 + c \end{aligned}$$

```
\begin{eqnarray*}
  z_0 &=& d = 0 \\
  z_{n+1} &=& z_n^2+c
\end{eqnarray*}
```

(you can have an empty left or right side of the alignment).

- `&` operator use to specify alignment.
- `&` operator can not be used in `$ $` mode and `\[ \]`
- Atmost two alignment option can be used in `eqnarray` in each line.

Allignment

Align using the `align*` environment (`&` operator use to specify alignment and you can use as many as you want alignment option in each line)

$$
\begin{aligned}
x &= y & w &= z & a &= b + c \\
2x &= -y & 3w &= \frac{1}{2}z & a &= b \\
-4 + 5x &= 2 + y & w + 2 &= -1 + w & ab &= cb
\end{aligned}
$$

Code:

```
\begin{align*}
x&=y          &  w &=z                  &  a&=b+c\\
2x&=-y        &  3w&=\frac{1}{2}z       &  a&=b\\
-4 + 5x&=2+y  &  w+2&=-1+w              &  ab&=cb
\end{align*}
```

Operator

- _ ^ Subscripts and superscripts.

Operator

- _ ^ Subscripts and superscripts.

$$\int_0^1 x^2 + y^2 \, dx$$

$$a_1^2 + a_2^2 = a_3^2$$

$$x^{2\alpha} - 1 = y_{ij} + y_{ij}$$

$$(a^n)^{r+s} = a^{nr+ns}$$

Operator

- _ ^ Subscripts and superscripts.

$$\int_0^1 x^2 + y^2 \ dx$$

$$a_1^2 + a_2^2 = a_3^2$$

$$x^{2\alpha} - 1 = y_{ij} + y_{ij}$$

$$(a^n)^{r+s} = a^{nr+ns}$$

```
\begin{align*}
\int\limits_0^1 x^2 + y^2 \ dx\\
a_1^2 + a_2^2 = a_3^2  \\
x^{2 \alpha} - 1 = y_{ij} + y_{ij}\\
(a^n)^{r+s} = a^{nr+ns}
\end{align*}
```

Operator

- _ ^ Subscripts and superscripts.
- A Complicated Expression

Operator

- _ ^ Subscripts and superscripts.
- A Complicated Expression

$$\sum_{i=1}^{\infty} \frac{1}{n^s} = \prod_{p} \frac{1}{1 - p^{-s}}$$

Operator

- _ ^ Subscripts and superscripts.
- A Complicated Expression

$$\sum_{i=1}^{\infty} \frac{1}{n^s} = \prod_{p} \frac{1}{1 - p^{-s}}$$

```
\[ \sum_{i=1}^{\infty} \frac{1}{n^s}
= \prod_p \frac{1}{1 - p^{-s}} \]
```

Operator

- `_` `^` Subscripts and superscripts.
- A Complicated Expression
- Use of `\lim` inside math mode.

Operator

- `_` `^` Subscripts and superscripts.
- A Complicated Expression
- Use of `\lim` inside math mode.

Sum $\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text

Improved sum $\displaystyle\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text

Operator

- `_` `^` Subscripts and superscripts.
- A Complicated Expression
- Use of `\lim` inside math mode.

Sum $\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text

Improved sum $\displaystyle\sum_{n=1}^{\infty} 2^{-n} = 1$ inside text

```
Sum $\sum_{n=1}^{\infty} 2^{-n} = 1$
 inside text\\

Improved sum
$\sum\limits_{n=1}^{\infty}
 2^{-n} = 1$ inside text
```

Operator

- `_` `^` Subscripts and superscripts.
- A Complicated Expression
- Use of `\lim` inside math mode.
- adding `\displaystyle` beforehand will make the symbol large and easier to read.

Operator

- `_` `^` Subscripts and superscripts.
- A Complicated Expression
- Use of `\lim` inside math mode.
- adding `\displaystyle` beforehand will make the symbol large and easier to read.

Try the following code

```
$\displaystyle\int\frac{1}{2}dx -
\int\frac{1}{2}dx$\\
$\int\frac{1}{2}dx -
\mathlarger{ \int\frac{1}{2}dx}$
```

Brackets and Parentheses

- `{ }` these parentheses is reserved for latex compiler. To get those parenthesis you have to use `\{ \}` .

Brackets and Parentheses

- { } these parentheses is reserved for latex compiler. To get those parenthesis you have to use \{ \} .

$$\left\{ \begin{array}{ccc} 1 & 5 & 8 \\ 0 & 2 & 4 \\ 3 & 3 & -8 \end{array} \right\}$$

Brackets and Parentheses

- { } these parentheses is reserved for latex compiler. To get those parenthesis you have to use \{ \} .

$$\left\{ \begin{tabular}{ccc} 1 & 5 & 8 \\ 0 & 2 & 4 \\ 3 & 3 & -8 \end{tabular} \right\}$$

```
\[
\left \{
  \begin{tabular}{ccc}
  1 & 5 & 8 \\
  0 & 2 & 4 \\
  3 & 3 & -8
  \end{tabular}
\right \}
\]
```

Brackets and Parentheses

- `{ }` these parentheses is reserved for latex compiler. To get those parenthesis you have to use `\{ \}` .
- Manually sized brackets

Brackets and Parentheses

- { } these parentheses is reserved for latex compiler. To get those parenthesis you have to use \{ \} .
- Manually sized brackets

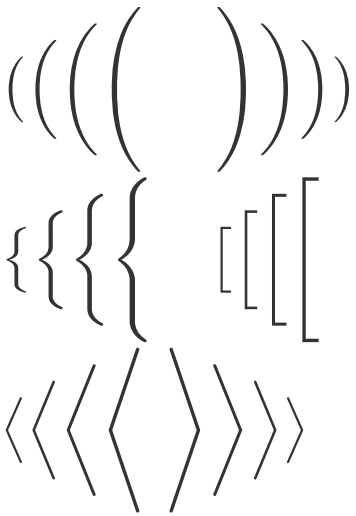
$$\left(\left(\left(\left(\right)\right)\right)\right)$$

$$\left\{\left\{\left\{\left\{\right.\right.\right.\right.\left.\left.\left.\left[\left[\left[\left[\right]\right]\right]\right]\right.\right.\right.\right.$$

$$\left\langle\left\langle\left\langle\left\langle\right\rangle\right\rangle\right\rangle\right\rangle$$

Brackets and Parentheses

- `{ }` these parentheses is reserved for latex compiler. To get those parenthesis you have to use `\{ \}` .
- Manually sized brackets



```
$\big( \Big( \bigg( \Bigg($
$\Bigg)\bigg)\Big)\big)$\\
$\big\{ \Big\{ \bigg\{ \Bigg\{$
$\big[ \Big[ \bigg[ \Bigg[$\\
$\big \langle \Big \langle \bigg \langle \Bigg \langle
$\Bigg \rangle\bigg \rangle\Big \rangle \big \rangle
$
```

New Environment

- New environments can be defined in the preamble in following way

New Environment

- New environments can be defined in the preamble in following way

New Environment

- New environments can be defined in the preamble in following way

```
\newtheorem{theorem}{Theorem}[section]
\newtheorem{corollary}{Corollary}[theorem]
\newtheorem{lemma}[theorem]{Lemma}
```

The first command wile create new environment `theorem` and it has the additional parameter [section] that restarts the theorem counter at every new section.

New Environment

- New environments can be defined in the preamble in following way

The second command wile create new environment `corollary` and the counter of this new environment will be reset every time a new theorem environment is used.

New Environment

- New environments can be defined in the preamble in following way

In last case a new environment called lemma is created, it will use the same counter as the theorem environment.

New Environment

- New environments can be defined in the preamble in following way
- `\usepackage{amsmath}` provide many predefined environment.

THANK
YOU