

✕ (/)

让 Hexo 搭建的博客支持 LaTeX

📅 2019-03-16 📌 Hexo (/tags/Hexo/)

数学是个充斥着糟糕符号的领域，这些符号在语义上的随意性给知识的交流和传承带来了巨大的困扰，人们在参与数学活动的时候，超过一半的时间是用来尝试理解这些数学符号，而不是处理数学问题本身。尽管如此，使用这些符号来表达数学思想确实是现阶段不可避免的。



LaTeX (<https://www.latex-project.org/>) 可以让我们在博客里使用这些数学符号，它有着特定的语法。下面的教程展示了如何让通过 Hexo 技术搭建的博客 (/2016/10/08/how-to-blog-with-hexo/) 支持显示使用 LaTeX 语法所描述的数学表达式。

安装插件

安装 hexo-math (<https://github.com/hexojs/hexo-math>) 插件，该插件 (plugin) 可支持使用 MathJax (<https://www.mathjax.org>) 或 KaTeX (<https://katex.org>) 来实现 LaTeX 排版系统，进而在网页上渲染出数学表达式（本文以 MathJax 为例）。

```
## 打开终端，进入 hexo 博客所在文件夹
```

```
$ cd ~/blog
```

```
## 安装 hexo ; --save 参数会让 npm 在安装 hexo-math 之后自动将它写入 packag
```

```
$ npm install hexo-math --save
```

将 Hexo 默认的 markdown 渲染引擎 `hexo-renderer-marked` (<https://github.com/hexojs/hexo-renderer-marked>) 更换为 `hexo-renderer-kramed` (<https://github.com/sun11/hexo-renderer-kramed>)，引擎是在默认的渲染引擎的基础上修改了一些 bug 而已。此处不更换也没问题，本文以更换为例。

```
## 卸载默认 markdown 渲染引擎 hexo-renderer-marked; 若不卸载，会和新的引擎发生冲突
$ npm uninstall hexo-renderer-marked --save

## 安装新引擎 hexo-renderer-kramed
$ npm install hexo-renderer-kramed --save
```

修改 kramed 配置，解决语义冲突



由于 LaTeX 与 Markdown 语法存在冲突（例如在 markdown 中，斜体可以用 `*` 或者 `_` 表示，而 LaTeX 也会用到 `_`），所以我们要对 kramed 默认的语法规则进行修改，否则之后会出现很多奇怪的排版样式。

打开 `~/blog/node_modules\kramed\lib\rules\inline.js` 文件（Hexo 博客所在文件夹的根目录下的 `node_modules` 文件夹），把第 11 行的 `escape` 变量的值修改为：

```
escape: /^\([`*\[\]()#$+~-.!_>])/,
```

同时把第 20 行的 `em` 变量修改为：

```
em: /^\((?:\*|_|[\s\S])+?)\*(?!\\*)/,
```

改好后的代码块显示如下：

```
// ~/blog/node_modules\kramed\lib\rules\inline.js

var inline = {
  //escape: /\^\\([\^\\`*{}\\[\]()#$+\\-\\.!_>])/,           // 注释掉的默认规则
  escape: /\^\\([\^\\`*\\[\]()#$+\\-\\.!_>])/,               // 新增的规则
  autolink: /\^<([\^ >]+(@|:\/)[\^ >]+)>/,
  url: noop,
  html: /\^<!--[\s\S]*?-->|\^<(\w+(?:\:\/\/|[\^\\w\s@]*@)\b)*?(?:\"[^\"]*"|'[^']*'/,
  link: /\^!?\\[(inside)\\]\(href\\)/,
  reflink: /\^!?\\[(inside)\\]\s*\\([\\^\\])*\\)/,
  nolink: /\^!?\\[(?:\\([\\^\\])*\\)|\\([\\^\\])*\\)/,
  reffn: /\^!?\\[\\^\\(inside)\\]/,
  strong: /\^__([\\s\\S]+?)__(!_)|\\^\\*\\*([\\s\\S]+?)\\*\\*(?!\\*)/,
  //em: /\^\\b_((?:__|\\[\\s\\S])+?)_\\b|\\^\\*\\*([\\s\\S]+?)\\*\\*(?!\\*)/,
  em: /\^\\*\\*([\\s\\S]+?)\\*\\*(?!\\*)/,
  code: /\^(`+)\s*([\\s\\S]*?[\\^`])\s*\\1(?:!`)/,
  br: /\^ {2,}\\n(?:!\\s*$)/,
  del: noop,
  text: /\^\\[\\s\\S]+?(?=[\\^<|\\[\\_`$]| {2,}\\n|$)/,
  math: /\^\\$\\$\\s*([\\s\\S]*?[\\^\\$])\\s*\\$\\$\\s*(?!\\$)/,
};

//...
```

设置 _config.yml 开启 MathJax 渲染引擎（重要）

在 ~/blog/_config.yml 文件（注意，是 Hexo 博客文件夹根目录中的 /_config.yml 而不是主题目录下的 /themes/next/_config.yml）中增加 MathJax 的支持，并手动设置下面的 src（这一步很重要，使用默认的 src 会导致数学表达式渲染显示失败。这里的关键是 src 中的 ?config=TeX-MML-AM_CHTML 这个字段）

```

✕. (/)
...

# MathJax
math:
  engine: 'mathjax'
  mathjax:
    src: https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.4/MathJax.

...

```

使用 LaTeX 语法显示数学表达式



经过以上所有设置后，重启 Hexo，让各种设置生效

```

$ hexo clean    ## 清除缓存 - 让新的 markdown 渲染引擎 hexo-renderer-kramed
$ hexo s        ## 启动服务器 - 让 _config.yml 文件中的配置生效

```

LaTeX 的常用语法如下（更多语法请参考基本语法）：

- 使用 `\\(` 和 `\\)`，或者 ``$`` 和 `$`` 来包裹一个内联（inline）的数学表达式，建议优先用后者，前者常出现语法冲突。后者有问题再换用前者。（参见 hexo-renderer-kramed 的官方说明 (<https://github.com/sun11/hexo-renderer-kramed>) 了解更多 **Tips**)

```

`$\pi$`
`$\int_a^b f(x) dx$`
\\(lim_{x\rightarrow \infty}\frac{1}{\sin x}\\)
\\(lim_{n\rightarrow \infty}(1+2^n+3^n)^{\frac{1}{x+\sin n}}\\)

```

✕ 上述代码分别显示为： π ， $\int_a^b f(x)dx$ ， $\lim_{x \rightarrow \infty} \frac{1}{\sin x}$ 和 $\lim_{n \rightarrow \infty} (1 + 2^n + 3^n)^{\frac{1}{x + \sin n}}$

- 使用 `\\[` 和 `\\]` 或者一对 `$$` 来包裹一个块状 (block) 的数学表达式 (这类表达式会在页面上居中显示)

```

$$
A = \begin{bmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
a_{31} & a_{22} & \dots & a_{3n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{bmatrix}, b = \begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
\vdots \\
b_n
\end{bmatrix}
$$

```

以上代码会显示一个矩阵 (居中显示)：

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{22} & \dots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

下面是概率界的贝叶斯公式：


✘ (/\$\$

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{\sum_{j=1}^n P(A_j)P(B | A_j)}$$
 \$\$

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{\sum_{j=1}^n P(A_j)P(B | A_j)}$$

sign 函数：

```
\begin{equation}
  sign(x)=\begin{cases}
    -1 & \text{if } x<0, \\
    0 & \text{if } x=0, \\
    1 & \text{if } x>0.
  \end{cases}
\end{equation}
```



$$sign(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

其他看起来复杂的数学表达式：

```
$$
\begin{split}
& \frac{\partial \mathcal{E}}{\partial x_l} = \\
& \frac{\partial \mathcal{E}}{\partial x_L} \frac{\partial x_L}{\partial x_l} = \frac{\partial \mathcal{E}}{\partial x_L} \Big(1 + \frac{\partial \mathcal{F}(x_i, \mathcal{W}_i)}{\partial x_L} \Big)
\end{split}
$$
```

✕ (/)

$$\frac{\partial \mathcal{E}}{\partial x_l} = \frac{\partial \mathcal{E}}{\partial x_L} \frac{\partial x_L}{\partial x_l}$$

$$= \frac{\partial \mathcal{E}}{\partial x_L} \left(1 + \frac{\partial}{\partial x_l} \sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i) \right)$$

引用与延伸

有关 Hexo 设置参考资料：

- MathJax - Use Math in Hexo, Just Like Tex! (Including Common Issue Solutions) (<https://linkinpark213.com/2018/04/24/mathjax/>)
- hexo 下 LaTeX 无法显示的解决方案 (<https://www.jianshu.com/p/d95a4795f3a8>)
- 在 hexo 使用 mathjax (<http://www.sail.name/2018/05/31/use-mathjax-in-hexo/>)

更多 LaTeX 语法请参考：

- A Primer on Using LaTeX in Jupyter Notebooks (<http://data-blog.udacity.com/posts/2016/10/latex-primer/>) (写得挺好的，所以尽管原文页面已无法访问，还是写出来做个纪念。网上能搜到该文，我的 Dropbox 里也有保存)
- Online LaTeX reference (<https://docs.latexbase.com/>) (比上面更详细更多细节)

打赏

✖ (/)

