



HARBIN INSTITUTE OF TECHNOLOGY

LILAC TECHREPROT №.000

# papyrus 模板介绍及使用说明

Ver 0.3

作者 1 作者 2 著

作者 1 作者 2 译

指导老师

指导老师 1 指导老师 2

2016 年 10 月 25 日

# papyrus 模板介绍及使用说明

## Ver 0.3

7HxzZ<sup>1</sup>, 作者<sup>2</sup>

作者<sup>3</sup>

<sup>1</sup> 哈尔滨工业大学 Lilac 战队

<sup>2</sup> 银河工业大学

<sup>3</sup> 宇宙工业大学

**摘 要** 摘要是对文章的内容不加注释和评论的简短陈述，要求扼要地说明研究工作的目的、研究方法和最终结论等。本模板目前是用一种非常 ad-hoc 方法实现排版摘要，用户只需要在相关地方输入相关文字即可。

**关键词** 关键词, 要用, 英文逗号, 隔开

## 1 Papyrus 模板简述

本文档是哈尔滨工业大学 7HxzZ 战队 WriteUp L<sup>A</sup>T<sub>E</sub>X 写作模板 papyrus 的简单简介。虽然使用 Markdown 写作比较方便，但对于一些高级格式，Markdown 无法正确排版。另外，如果想要将文档整理成适合印刷的出版物，Markdown 可能就有出力不从心了。故此，整理了一份 L<sup>A</sup>T<sub>E</sub>X 格式的 WriteUp 写作模板。本模板不但可用于排版单个 WriteUp，还可用来排版相关技术报告或科技论文等，具有极高的实用价值。

papyrus 是“羊皮卷”的意思，本模板的特色有：

1. 可以单独按照文章编译（ctexart 类），也可以将所有 WriteUp 文章整理成册，按照书籍编译（ctexbook 类）。两者无需改动文章源码。
2. 提供了 \code 命令，可用于编排单行代码（不完美）。
3. 封装了 minted 环境，可以简单地排版出含有代码高亮的代码。
4. 提供了 quizdesc 环境，可用于编排问答模式 CTF 比赛的题目描述。

本模板托管在 <https://github.com/DeathKing/papyrus>，用户可以关注该代码仓库，获得本模板的最新更新。当然，本模板还有许多问题，需要大家多提提意见，最好能发一些 Pull Request。

## 2 学习 L<sup>A</sup>T<sub>E</sub>X 排版的相关资源

跟 Markdown 相比，L<sup>A</sup>T<sub>E</sub>X 可能稍显复杂。但是，如果稍微花费一点时间学习 L<sup>A</sup>T<sub>E</sub>X，你会惊奇地发现，L<sup>A</sup>T<sub>E</sub>X 是如此地强大！碍于版面的限制，这里简单地罗列一些 L<sup>A</sup>T<sub>E</sub>X 学习资源，读者可以根据自身情况选择合适的资料进行学习。

1. 一份不太简短的 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 介绍 (或 93 分钟学会 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>): <http://www.mohu.org/info/lshort-cn.pdf>。这份约莫 100 页的中文文档, 囊括了 L<sup>A</sup>T<sub>E</sub>X 最常用的功能, 之中包含了大量的范例代码, 简洁易懂。
2. L<sup>A</sup>T<sub>E</sub>X Tutorials A PRIMER: <http://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf>。这份广受赞誉的 L<sup>A</sup>T<sub>E</sub>X 教程是由印度 T<sub>E</sub>X 用户组撰写的, 全书用英文写就。
3. 邹思宇的 L<sup>A</sup>T<sub>E</sub>X 排版手记: <https://github.com/Zousiyu/Study-LaTeX>。这是网友自己整理的 L<sup>A</sup>T<sub>E</sub>X 学习笔记, 对初学者来说很有用处。
4. 一份其实很短的 L<sup>A</sup>T<sub>E</sub>X 入门文档: <http://liam0205.me/2014/09/08/latex-introduction/>。这是由 C<sub>T</sub><sub>E</sub>X 开发者之一撰写的文档, 有很高的人气。
5. More Math Into L<sup>A</sup>T<sub>E</sub>X: [http://www.latexstudio.net/wp-content/uploads/2016/09/More\\_Math\\_Into\\_LaTeX-Springer2016.pdf](http://www.latexstudio.net/wp-content/uploads/2016/09/More_Math_Into_LaTeX-Springer2016.pdf)。这是一本关于 L<sup>A</sup>T<sub>E</sub>X 数学公式排版的详尽参考书, 如果你想进一步成为 L<sup>A</sup>T<sub>E</sub>X 排版专家或者使用 L<sup>A</sup>T<sub>E</sub>X 进行严肃的学术写作, 那么请仔细阅读本书!

## 3 关于 papyrus 宏包的使用说明

### 3.1 引入 papyrus 宏包

本模板的核心是 papyrus 宏包, 该宏包使用 minted 宏包编排代码, 而该宏包与 TeXpad 软件有部分冲突, 本样式文件已修复。请用户在使用本模板时, 注意第 25 行左右的代码:

---

```

25 %=====
26 % TeXpad 请使用词条命令
27 \usepackage[texpad]{papyrus}
28 %-----
29 % 其它软件用户请使用下述命令
30 % \usepackage{papyrus}
31 %=====

```

---

请根据自己使用的编辑器选择合适的命令引入 papyrus 样式包!

### 3.2 quizdesc 环境的使用

第一节通常用来简要描述题目的要求, 可以用 `\begin{quizdesc}...\end{quizdesc}` 环境来描述:

Crypto100 Simple

Become admin!

`http://52.69.244.164:51913`

`simple-01018f60e497b8180d6c92237e2b3a67.rb`

对应的代码是:

---

---

```

1 \begin{quizdesc}[label=Crypto100 Simple]
2 Become admin!
3
4 http://52.69.244.164:51913
5 simple-01018f60e497b8180d6c92237e2b3a67.rb
6 \end{quizdesc}

```

---

### 3.3 代码编排

撰写 WriteUp 时，经常需要插入代码。行内代码请使用 `\code/inline code/` 来编排，例如，代码 `while (true) { printf("Hello, World!\n"); }` 的编排代码如下：

---

```

1 \code/while (true) { printf("Hello, World!\n"); } /

```

---

当然，也可以使用 `\verb|the-inline-code|` 来编排行内代码，这个功能由 Verbtain 包提供。

由于编排行内代码存在对特殊符号支持不是很好，所以不推荐用单行代码来编排 Flag。多行代码请使用 `minted` 环境来编排：

---

```

1 #!/usr/bin/env ruby
2
3 class SampleCode
4   def initialize
5   end
6 end

```

---

对应的代码是：

```

\begin{minted}{ruby}
#!/usr/bin/env ruby

class SampleCode
  def initialize
  end
end
\end{minted}

```

`minted` 环境对于 Tab 缩进支持不是很好，推荐使用空格进行缩进。对于大部分代码（特别是 Python 代码），由于其前面自带的 Tab 缩进，推荐为 `minted` 环境添加 `[autogobble]` 选项来去除不必要的缩进，或者使用 Sublime Text Editor 等具有列编辑功能的编辑器快速删除掉缩进。

关于 `minted` 环境的使用方法，请参考：<ftp://ftp.dante.de/tex-archive/macros/latex/contrib/minted/minted.pdf>。

### 3.4 关于代码高亮 - Pygments

`minted` 环境是靠一个外部工具 Pygments 实现代码高亮的，因此需要先安装并设置 Pygments。这

里提供几个有用的连接:

1. 如何安装(特别是 Pygments): <http://tex.stackexchange.com/questions/108661/how-to-use-minted-under>
2. 支持的语言: <http://pygments.org/languages/>
3. 支持的词法器(这个网页中的 Short names 就应该是你的语言名称): <http://pygments.org/docs/lexers/>

### 3.5 图片的编排

插图是科技论文和 WriteUp 写作的硬需求,很多同学反映用  $\text{\LaTeX}$  插图非常麻烦。其实不然,利用 `\includegraphics{}` 命令,就可以很轻松地插入一张图片。通常来说,我们把这个图片放入一个 `figure` 环境下,并使用一个 `\caption{}` 命令为图标标注名称。

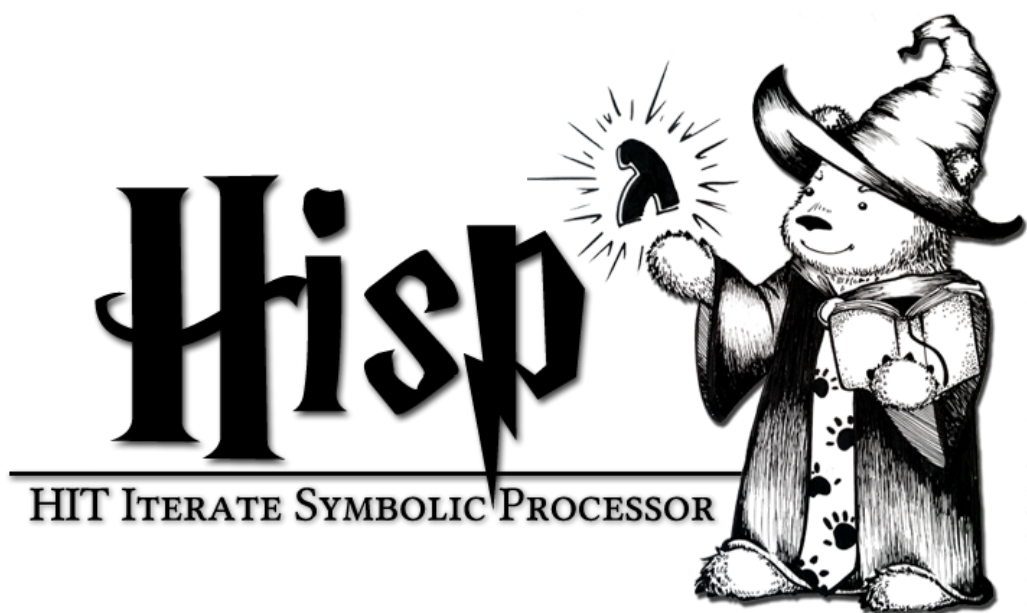


图 3.1: Hisp 语言

图 3.1 是 Hisp 语言的 Logo。它对应的代码如下:

---

```

1 \begin{figure}[hbt!]
2   \includegraphics[width=0.80\textwidth,center]{hisp2}
3   \caption{Hisp 语言}\label{hisp}
4 \end{figure}

```

---

这段代码的解释如下:

1. 首先,我们启用了 `figure` 环境,图形 (figure) 环境有一个可选参数项允许用户来指示图形有可能被放置的位置。关于参数的具体意义,请参考 <http://www.ctex.org/documents/latex/graphics/node64.html>。
2. 然后, `\includegraphics{}` 命令用于插入一组图片,不用书写图片的扩展名,并且要将图片与 `.tex` 文件存放在同一文件夹。(绝对路径、设定了 `\includegraphics{}` 命令搜索的文件夹的情况下除外。)

`\includegraphics{}` 命令还可以接收一组参数, 本例中 `width=0.80\textwidth` 就告知  $\text{\LaTeX}$ , 将图片按宽度缩放为文本行宽的 0.8 倍大小。而 `center` 是由 `adjustbox` 宏包提供的选项参数, 用于将图片居中。

3. 特别地, 可以用 `\label{}` 命令来标记图表或公示, 利用 `\ref{}` 命令来引用它们对应的编号。

3.6 bytefield 包简单使用说明

考虑到渗透和逆向题目的 WriteUp 中经常需要编排内存布局, 本模板的 Ver0.2 版本引入了 `bytefield` 包, 可用于描绘内存布局。图 3.2 就是一个使用 `bytefield` 包绘制的内存布局图, 其代码如代码片段 3.1 所示。

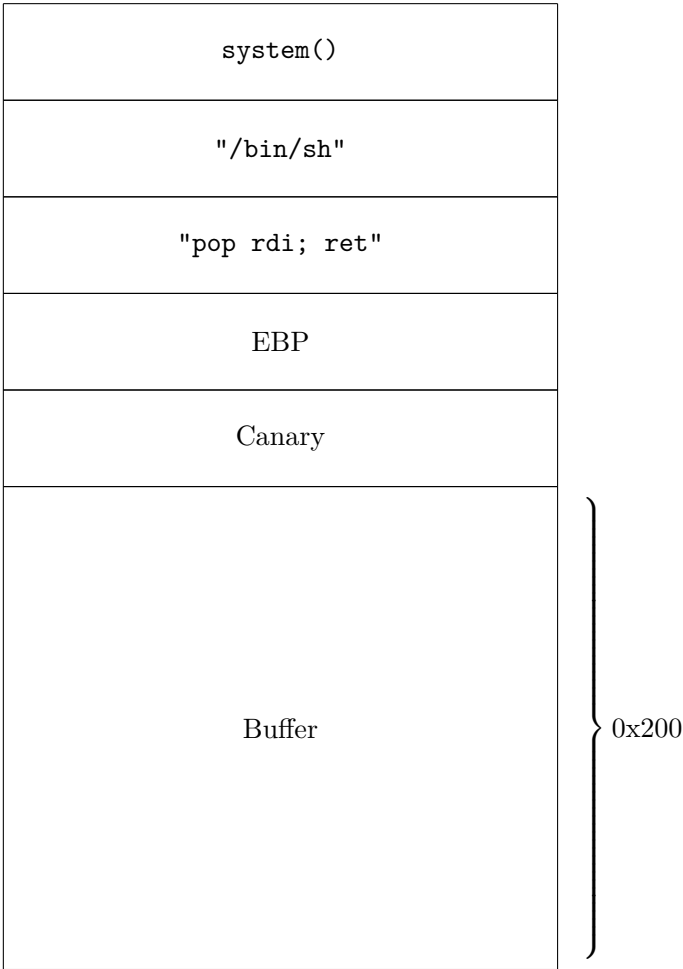


图 3.2: 攻击后的栈帧布局

3.7 ulem 宏包使用方法

`ulem` 宏包主要用于编排下划线、删除线, 代码及对应的效果如表 3.1 所示:

```
1 \begin{figure}[!htb]
2 \begin{center}
3 \begin{bytefield}{24}
4     \wordbox{2}{\code/system()/}\
5     \wordbox{2}{\Verb|"/bin/sh"|}\
6     \wordbox{2}{\code/"pop rdi; ret"/}\
7     \wordbox{2}{EBP}\
8     \wordbox{2}{Canary}\
9     \begin{rightwordgroup}{0x200}
10        \wordbox{10}{Buffer}
11    \end{rightwordgroup}
12 \end{bytefield}
13 \end{center}
14 \caption{攻击后的栈帧布局}\label{fig:hacked}
15 \end{figure}
```

代码片段 3.1: bytefiled 包使用代码示意

表 3.1: ulem 包编排效果

<code>\uline{important}</code>	underlined text like <u>important</u>
<code>\uuline{urgent}</code>	double-underlined text like <u><u>urgent</u></u>
<code>\uwave{boat}</code>	wavy underline like <u>boat</u>
<code>\sout{wrong}</code>	line struck through word like <del>wrong</del>
<code>\xout{removed}</code>	marked over like <del>removed</del>
<code>\dashuline{dashing}</code>	dashed underline like <u>dashing</u>
<code>\dotuline{dotty}</code>	dotted underline like <u>dotty</u>

4 其它元素的编排

4.1 参考文献的编排

papyrus 可以快速方便地编排参考文献。例如，本模板中的参考文献编排代码如代码片段4.1所示，如果想要引用文献，可以使用`\cite{the-label-name}[1]` 来实现。

```
1 \begin{thebibliography}{99}
2
3 \bibitem{wiki_aes} Advanced Encryption Standard,
4   \url{https://en.wikipedia.org/wiki/Advanced_Encryption_Standard}
5 \bibitem{wiki_bcmo} Block cipher mode of operation,
6   \url{https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation}
7 \bibitem{sch07} Schneier B. Applied cryptography: protocols, algorithms,
8   and source code in C[M]. john wiley \& sons, 2007.
9 \end{thebibliography}
```

代码片段 4.1: 参考文献的编排

## 4.2 附录的编排

对于一些行数较多的代码或代码文件，推荐将内容编排到附录中，并在正文中引用该章节。附录的编排请参考 `papyrus.tex` 源码。

## 参考文献

- [1] Advanced Encryption Standard, [https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)
- [2] Block cipher mode of operation, [https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation)
- [3] Schneier B. Applied cryptography: protocols, algorithms, and source code in C[M]. john wiley & sons, 2007.

# 附录

## A 附录 1

这里可以放置附录。

## B 附录 2

这里可以放置附录。