

zhlipsum: 中文乱数假文 (Lorem ipsum)

曾祥东

2018/01/06 v0.5*

如彭奇和瓦特曼的公共事业所证实的那样有一个胡子雪雪白的上帝超越时间超越空间确实存在他在神圣的冷漠神圣的疯狂神圣的哑哑的高处深深地爱着我们除了少数的例外不知什么原因但时间将会揭示他像神圣的密兰达一样和人们一起忍受着痛苦这班人不知什么原因但时间将会揭示生活在痛苦中生活在烈火中这烈火这火焰如果继续燃烧毫无疑问将使穹苍着火也就是说将地狱炸上天去天是那么蓝那么澄澈那么平静这种平静尽管时断时续总比没有好得多但是别这么快还要进一步考虑到泰斯丢和丘那德的人体测定学院的未完成的研究结果早已断定毫无疑问换句话说除了依附于人类的疑问之外别无其他疑问根据泰斯丢和丘那德的未完成的劳动的结果早已作出如下的论断但是别这么快不知什么原因根据彭奇和瓦特曼的公共事业的结果已毫无疑问地断定鉴于波波夫和贝尔契不知什么原因未完成的劳动以及泰斯丢和丘那德的未完成的劳动已经就业已被许多人所否认的论点作出论断认为泰斯丢和丘那德所假设的人认为实际存在的人认为人类总而言之统而言之尽管有进步的营养学和通大便药却在衰弱萎缩衰弱萎缩而且与此同时尤其是不知什么原因尽管体育运动在各方面都有很大进展如网球足球田径车赛游泳飞行划船骑马滑翔溜冰各式各样的网球各种各样致人死命的飞行运动各式各样的秋天夏天冬天冬天网球各种各样的曲棍球盘尼西林和代用品总之我接下去讲与此同时不知什么原因要萎缩要减少尽管有网球我接下去讲飞行滑翔九穴和十八穴的高尔夫球各种各样的网球总之不知什么原因在番克汉贝汉福尔汉克莱普汉换句话说与此同时尤其是不知什么原因但时间将会揭示要减少减少我接下去讲福尔汉克莱普汉总之自从塞缪尔·约翰逊去世以后到现在每个人的全部损失共计每人一吋四两只是大概约略粗粗计算到小数点分量很足保持整数赤裸裸的光穿着袜子在康纳马拉拉总之不知什么原因不管怎样无论如何事实俱在尤其是考虑到更加远为严肃的看来更加严肃的鉴于斯丹威格和彼特曼的徒劳看来更加严肃的鉴于鉴于鉴于斯丹威格和彼特曼徒劳在平原在山地海洋在烈火沸腾的河里天空是一样的随后是大地换句话说天空随后是大地在一片寒冷一片漆黑中天空大地石头的住所一片寒冷中哎哟哟在我们的主诞生六百年左右天空大地海洋大地石头的住所汪洋中一片寒冷中在海上在陆地空中我接下去讲不知什么原因尽管有网球事实俱在但时间将会揭示我接下去讲哎哟哟总之一句话石头的住所谁能怀疑我接下去讲但是别这么快我接下去讲头颅要萎缩衰弱减少与此同时尤其是不知什么原因尽管有网球胡子火焰球队石头那么蓝那么平静哎哟哟头颅头颅头颅头颅在康纳马拉拉尽管有网球未完成的徒然的劳动更加严肃的石头的住所总之我接下去讲哎哟哟徒劳的未完成的头颅头颅在康纳马拉拉尽管有网球头颅哎哟石头丘那德（混战，最后的狂喊）网球……石头……那么平静……丘那德……未完成的……

——萨缪尔·贝克特《等待戈多》

*<https://github.com/Stone-Zeng/zhlipsum>.

第 1 节 简介

zhlipsum 宏包用于输入中文乱数假文（拉丁语为 *Lorem ipsum*）。乱数假文是大段无意义的文字，常用来测试排版效果。支持其他语言乱数假文的宏包还有 lipsum、kantlipsum、blindtext 等。

zhlipsum 宏包支持 UTF-8、GBK 和 Big5 编码，依赖 L^AT_EX3 项目中的 expl3、xparse 和 l3keys2e 宏包。为正确输入中文，zhlipsum 需要与 CJK 宏包或 C_TE_X 宏集等配套使用。

第 2 节 使用说明

encoding
New: 2017-09-16
Updated: 2018-01-05

encoding = <utf8|gbk|big5>

用于指定编码的宏包选项，可在调用宏包的时候设定。默认为 utf8。对于 X_YL^AT_EX、LuaL^AT_EX 和 upL^AT_EX 等 Unicode 引擎，gbk 和 big5 编码无效，宏包将强制使用 utf8 编码。

在 C_TE_X 宏集中，相应的宏包选项为 UTF8 和 GBK。注意本宏包中所有选项均为小写。

\zhlipsum
Updated: 2018-01-05

\zhlipsum[<段落>][<选项>]
\zhlipsum*[<段落>][<选项>]

插入假文文本。默认会在每段文本之间分段（添加 \par）。如需禁止分段，请使用带星号的命令。在两种情况下，假文文本的前后都会强制分段。

第一个可选参数 <段落> 可通过如下方式指定：

- 只给定一个参数时，将输出对应序号的段落，如 \zhlipsum[17] 将生成第 17 段；
- 给定两个参数并用 “-” 连接时，则会输出该范围内的段落。如 \zhlipsum[5-34] 将生成第 5 至 34 段假文；
- 没有指定段落时，默认输出第 1 至 3 段假文。

目前本宏包只提供了 50 段假文文本，超出范围的数字一律按 50 处理。如 \zhlipsum[51] 将给出第 50 段，而 \zhlipsum[48-55] 将只会输出第 48、49 和第 50 段。

第二个可选参数 <选项> 通过英文逗号分隔的键值列表形式给出。如果需要设置选项，则必须显式地指明段落参数。支持的选项如下。

script
New: 2018-01-05

script = <simp|trad>

选择插入简体中文（simp）或繁体中文（trad）。注意在 Big5 编码下，只可使用繁体中文。其他编码下，默认使用简体中文。

第 3 节 已知问题

GBK 和 Big5 编码在第二字节并没有避开 ASCII 码的范围，因此部分汉字编码的第二字节恰好是 ASCII 编码中的一些特殊字符（如 {、}、\ 等），将导致编译失败。本宏包将这两种编码下的 .def 文件转换为了较安全的形式，请勿手工修改这些文件。

如无特殊需要，始终建议您采用 UTF-8 编码，并使用 Xe_{La}TeX、Lua_{La}TeX 等 Unicode 引擎编译。

第 4 节 实现细节

```

1 <{*package>
2 <{@@=zhlipsum>

    检查 LATEX3 编程环境。
3 \RequirePackage { xparse, l3keys2e }
4 \msg_new:nnn { zhlipsum } { l3-too-old }
5 {
6     Package~ "#1"~ is~ too~ old. \\\
7     Please~ update~ an~ up-to-date~ version~ of~ the~ bundles \
8     "l3kernel"~ and~ "l3packages"~ using~ your~ TeX~ package \
9     manager~ or~ from~ CTAN.
10 }
11 \clist_map_inline:nn { expl3, xparse, l3keys2e }
12 {
13     \@ifpackagelater {#1} { 2017/12/16 }
14     { } { \msg_error:nnn { zhlipsum } { l3-too-old } {#1} }
15 }
```

4.1 内部变量和函数

`\l__zhlipsum_tmpa_str` 临时变量。

```
16 \str_new:N \l__zhlipsum_tmpa_str
```

`\g__zhlipsum_encoding_tl` 存放编码信息。

```
17 \tl_new:N \g__zhlipsum_encoding_tl
```

`\c__zhlipsum_max_par_number_int` 假文的最大段落数。目前设为 50。

```
18 \int_const:Nn \c__zhlipsum_max_par_number_int { 50 }
```

`\tl_if_eq:VnTF` L_AT_EX3 函数变体。

```
19 \prg_generate_conditional_variant:Nnn \tl_if_eq:nn { Vn } { T, TF }
```

`__zhlipsum_if_unicode_engine:TF` 判断是否为 Unicode 引擎。来自于 `zhnumber` 宏包。

```

20 \bool_lazy_any:nTF
21 {
22     { \sys_if_engine_xetex_p: }
23     { \sys_if_engine luatex_p: }
24     { \sys_if_engine_uptex_p: }
25 }
26 { \cs_new_eq:NN \__zhlipsum_if_unicode_engine:TF \use_i:nn }
27 { \cs_new_eq:NN \__zhlipsum_if_unicode_engine:TF \use_ii:nn }
```

`__zhlipsum_input_def_file:n` 读入定义文件。

```

\__zhlipsum_input_def_file:V
28 \cs_new:Npn \__zhlipsum_input_def_file:n #1
29 { \file_input:n { zhlipsum- #1 .def } }
30 \cs_generate_variant:Nn \__zhlipsum_input_def_file:n { V }
```

`__zhlipsum_par:` 分段命令。

```
31 \cs_new_eq:NN \__zhlipsum_par: \tex_par:D
```

4.2 宏包选项

encoding 设置编码。

```

32 \keys_define:nn { zhlipsum / option }
33 {
34   encoding .choice:,
35   encoding / utf8 .code:n =
36     { \tl_set:Nn \g__zhlipsum_encoding_tl { utf8 } },
37   encoding / gbk .code:n =
38     {
39       \__zhlipsum_if_unicode_engine:TF
40       {
41         \tl_set:Nn \g__zhlipsum_encoding_tl { utf8 }
42         \msg_warning:nnn { zhlipsum } { unicode-engine } { gbk }
43       }
44       { \tl_set:Nn \g__zhlipsum_encoding_tl { gbk } }
45     },
46   encoding / big5 .code:n =
47     {
48       \__zhlipsum_if_unicode_engine:TF
49       {
50         \tl_set:Nn \g__zhlipsum_encoding_tl { utf8 }
51         \msg_warning:nnn { zhlipsum } { unicode-engine } { big5 }
52       }
53       { \tl_set:Nn \g__zhlipsum_encoding_tl { big5 } }
54     },
55   encoding / unknown .code:n =
56     { \msg_error:nnn { zhlipsum } { encoding-invalid } { #1 } },
57   encoding .value_required:n = true,
58   encoding .initial:n = { utf8 },

```

处理未知选项。

```

59   unknown .code:n =
60     { \msg_error:nn { zhlipsum } { unknown-option } }
61 }

```

提示信息。

```

62 \msg_new:nnn { zhlipsum } { unicode-engine }
63 {
64   You~ are~ now~ using~ Unicode~ engine~ \c_sys_engine_str. \\
65   Encoding~ "#1"~ is~ invalid.~ Changed~ into~ "utf8".
66 }
67 \msg_new:nnn { zhlipsum } { encoding-invalid }
68 {
69   Encoding~ "#1"~ is~ invalid. \\
70   Available~ encodings~ are~ "utf8",~ "gbk"~ and~ "big5".
71 }
72 \msg_new:nnn { zhlipsum } { unknown-option }
73 { Package~ option~ "\l_keys_key_tl"~ is~ unknown. }

```

将宏包选项传给 zhlipsum/option。

```

74 \ProcessKeysOptions { zhlipsum / option }

```

根据编码读入假文文本定义文件。

```

75 \__zhlipsum_input_def_file:V \g__zhlipsum_encoding_tl

```

4.3 函数选项

\l__zhlipsum_script_str 存放 simp / trad。

```

76 \str_new:N \l__zhlipsum_script_str

```

script 选择输入简体中文或繁体中文。Big5 编码不支持简体中文。

```

77 \keys_define:nn { zhlipsum }
78 {
79   script .choice:,
80   script / simp .code:n =
81   {
82     \str_set:Nn \l__zhlipsum_script_str { simp }
83     \tl_if_eq:VnT \g__zhlipsum_encoding_tl { big5 }
84     {
85       \str_set:Nn \l__zhlipsum_script_str { trad }
86       \msg_warning:nn { zhlipsum } { big5-require-trad }
87     }
88   },
89   script / trad .code:n =
90   { \str_set:Nn \l__zhlipsum_script_str { trad } },
91   script / unknown .code:n =
92   { \msg_error:nnn { zhlipsum } { script-invalid } {#1} },
93   script .value_required:n = true
94 }

```

提示信息。

```

95 \msg_new:nnn { zhlipsum } { big5-require-trad }
96 {
97   "script=simp"~ is~ not~ available~ in~ "BIG5"~ encoding.\\
98   Changed~ into~ "script=trad".
99 }
100 \msg_new:nnn { zhlipsum } { script-invalid }
101 {
102   Script~ "#1"~ is~ unknown. \\
103   Available~ scripts~ are~ "simp"~ and~ "trad".
104 }

```

初始选项设置。

```

105 \tl_if_eq:VnTF \g__zhlipsum_encoding_tl { big5 }
106 { \keys_set:nn { zhlipsum } { script = trad } }
107 { \keys_set:nn { zhlipsum } { script = simp } }

```

4.4 核心函数

\zhlipsum 输出假文，第一个可选参数表示段落数，默认为 1-3；第二个可选参数为选项列表。命令前后均插入了 `__zhlipsum_par:`，以便强制分段。带星号的命令在假文主体间不分段。

```

108 \NewDocumentCommand \zhlipsum
109 { s >{ \SplitArgument { 1 } { - } } 0{ 1 - 3 } o }
110 {
111   \group_begin:
112   \__zhlipsum_process_par_number:nn #2
113   \IfValueT {#3} { \keys_set:nn { zhlipsum } {#3} }
114   \__zhlipsum_par:
115   \IfBooleanTF {#1}
116   {
117     \zhlipsum_print_multi_no_par:VVV
118     \l__zhlipsum_begin_int
119     \l__zhlipsum_step_int
120     \l__zhlipsum_end_int
121     \__zhlipsum_par:
122   }
123   {
124     \zhlipsum_print_multi:VVV
125     \l__zhlipsum_begin_int

```

126	\l__zhlipsum_step_int	
127	\l__zhlipsum_end_int	
128	}	
129	\group_end:	
130	}	
\l__zhlipsum_begin_int	\l__zhlipsum_end_int	\l__zhlipsum_step_int
\l__zhlipsum_end_int		
\l__zhlipsum_step_int		
__zhlipsum_process_par_number:nn		
131	\int_new:N \l__zhlipsum_begin_int	
132	\int_new:N \l__zhlipsum_end_int	
133	\int_new:N \l__zhlipsum_step_int	
134	\cs_new:Npn __zhlipsum_process_par_number:nn #1#2	
135	{	
136	__zhlipsum_process_par_number_aux:Nn \l__zhlipsum_begin_int {#1}	
137	\IfNoValueTF {#2}	
138	{ \int_set_eq:NN \l__zhlipsum_end_int \l__zhlipsum_begin_int }	
139	{ __zhlipsum_process_par_number_aux:Nn \l__zhlipsum_end_int {#2} }	
140	\int_compare:nTF	
141	{ \l__zhlipsum_begin_int <= \l__zhlipsum_end_int }	
142	{ \int_set_eq:NN \l__zhlipsum_step_int \c_one }	
143	{ \int_set:Nn \l__zhlipsum_step_int { -1 } }	
144	}	
__zhlipsum_process_par_number_aux:Nn		
145	\cs_new:Npn __zhlipsum_process_par_number_aux:Nn #1#2	
146	{	
147	\int_compare:nTF { #2 <= \c__zhlipsum_max_par_number_int }	
148	{ \int_set:Nn #1 {#2} }	
149	{ \int_set_eq:NN #1 \c__zhlipsum_max_par_number_int }	
150	}	
\zhlipsum_print_multi:nnn		
\zhlipsum_print_multi:VVV		
151	\cs_new:Npn \zhlipsum_print_multi:nnn #1#2#3	
152	{	
153	\int_step_inline:nnnn {#1} {#2} {#3}	
154	{ __zhlipsum_print_single:n {##1} __zhlipsum_par: }	
155	}	
156	\cs_generate_variant:Nn \zhlipsum_print_multi:nnn { VVV }	
\zhlipsum_print_multi_no_par:nnn		
\zhlipsum_print_multi_no_par:VVV		
157	\cs_new:Npn \zhlipsum_print_multi_no_par:nnn #1#2#3	
158	{	
159	\int_step_inline:nnnn {#1} {#2} {#3}	
160	{ __zhlipsum_print_single:n {##1} }	
161	}	
162	\cs_generate_variant:Nn \zhlipsum_print_multi_no_par:nnn { VVV }	
__zhlipsum_print_single:n		
163	\cs_new:Npn __zhlipsum_print_single:n #1	
164	{	
165	\str_set:Nx \l__zhlipsum_tmpa_str { \int_to_roman:n {#1} }	
166	\str_put_left:Nx \l__zhlipsum_tmpa_str	
167	{ c__zhlipsum_text_ \l__zhlipsum_script_str _ }	
168	\str_put_right:Nn \l__zhlipsum_tmpa_str { _tl }	
169	\use:c { \str_use:N \l__zhlipsum_tmpa_str }	
170	}	

循环的开始、结束和步长。

处理段落数参数。

获得循环的开始位置和结束位置。

开始位置在结束位置之前时，步长为 1，否则为 -1。

与最大段落数进行比较，防止下标越界。

输出带段落的假文。

输出不带段落的假文。

输出第 n 段假文。方法是将段落数 n 先转换成罗马数字，再组成完整命令。

171 `</package>`

版本历史

v0.1	(2017/04/08)	v0.4	(2017/09/16)
General: 开始编写宏包。	1	General: 将安装、测试文件集成进源文件。	1
v0.2	(2017/04/14)	新增 encoding 选项。	4
General: 仿照 kantlipsum 宏包, 实现任意的段落		优化宏包实现。	1
选取。	1	v0.5	(2017/12/22 – 2018/01/06)
使用名字空间。	1	General: 添加英文版用户文档。	1
v0.3	(2017/07/28)	新增选项 script, 同时支持简体中文和繁体中	
_zhlipsum_print_single:n: 改为内部函数。 . .	6	文。	4
_zhlipsum_process_par_number:nn: 改为内部		支持 Big5 编码。	4
函数。	6	\zhlipsum: 支持选项设置。	5

代码索引

意大利体的数字表示描述对应索引项的页码; 带下划线的数字表示定义对应索引项的代码行号; 罗马字体的数字表示使用对应索引项的代码行号。

B		\int_set_eq:NN	138, 142, 149
bool commands:		\int_step_inline:nnnn	153, 159
\bool_lazy_any:nTF	20	\int_to_roman:n	165
C		\c_one	142
clist commands:		K	
\clist_map_inline:nn	11	keys commands:	
cs commands:		\keys_define:nn	32, 77
\cs_generate_variant:Nn	30, 156, 162	\l_keys_key_tl	73
\cs_new:Npn	28, 134, 145, 151, 157, 163	\keys_set:nn	106, 107, 113
\cs_new_eq:NN	26, 27, 31	M	
E		msg commands:	
encoding	2, 32	\msg_error:nn	60
F		\msg_error:nnn	14, 56, 92
file commands:		\msg_new:nnn	4, 62, 67, 72, 95, 100
\file_input:n	29	\msg_warning:nn	86
G		\msg_warning:nnn	42, 51
group commands:		N	
\group_begin:	111	\NewDocumentCommand	108
\group_end:	129	P	
I		prg commands:	
\IfBooleanTF	115	\prg_generate_conditional_variant:Nnn . .	19
\IfNoValueTF	137	\ProcessKeysOptions	74
\IfValueTF	113	R	
int commands:		\RequirePackage	3
\int_compare:nTF	140, 147	S	
\int_const:Nn	18	script	2, 77
\int_new:N	131, 132, 133	\SplitArgument	109
\int_set:Nn	143, 148		

str commands:	<code>\use_i:nn</code> 26
<code>\str_new:N</code> 16, 76	<code>\use_ii:nn</code> 27
<code>\str_put_left:Nn</code> 166	
<code>\str_put_right:Nn</code> 168	Z
<code>\str_set:Nn</code> 82, 85, 90, 165	<code>\zhlipsum</code> 2, 108
<code>\str_use:N</code> 169	zhlipsum commands:
sys commands:	<code>\zhlipsum_print_multi:nnn</code> 124, 151
<code>\c_sys_engine_str</code> 64	<code>\zhlipsum_print_multi_no_par:nnn</code> .. 117, 157
<code>\sys_if_engine luatex_p:</code> 23	zhlipsum internal commands:
<code>\sys_if_engine uptex_p:</code> 24	<code>\l_zhlipsum_begin_int</code>
<code>\sys_if_engine xetex_p:</code> 22 118, 125, 131, 136, 138, 141
T	<code>\g_zhlipsum_encoding_tl</code>
T_EX and L^AT_EX₂_ε commands: 17, 36, 41, 44, 50, 53, 75, 83, 105
<code>\@ifpackagelater</code> 13	<code>\l_zhlipsum_end_int</code> 120, 127, 131, 138, 139, 141
<code>\par</code> 2	<code>__zhlipsum_if_unicode_engine:TF</code> . 20, 39, 48
tex commands:	<code>__zhlipsum_input_def_file:n</code> 28, 75
<code>\tex_par:D</code> 31	<code>\c_zhlipsum_max_par_number_int</code> . 18, 147, 149
tl commands:	<code>__zhlipsum_par:</code> 5, 31, 114, 121, 154
<code>\tl_if_eq:nn</code> 19	<code>__zhlipsum_print_single:n</code> 154, 160, 163
<code>\tl_if_eq:nnTF</code> 19, 83, 105	<code>__zhlipsum_process_par_number:nn</code> . 112, 134
<code>\tl_new:N</code> 17	<code>__zhlipsum_process_par_number_aux:Nn</code> ..
<code>\tl_set:Nn</code> 36, 41, 44, 50, 53 136, 139, 145
U	<code>\l_zhlipsum_script_str</code> ... 76, 82, 85, 90, 167
use commands:	<code>\l_zhlipsum_step_int</code> .. 119, 126, 131, 142, 143
<code>\use:N</code> 169	<code>\l_zhlipsum_tmpa_str</code> ... 16, 165, 166, 168, 169
	<code>\zhlipsum*</code> 2