

zhnumber 宏包

李清

sobenlee@gmail.com

2012/11/23 v1.6

1 简介

zhnumber 宏包用于将阿拉伯数字按照中文格式输出。相比于 CJKnumb, 它提供的三个格式转换命令 `\zhnumber`, `\zhdigits` 和 `\zhnum` 都是可以适当展开的, 可以正常使用于 PDF 书签和交叉引用。

zhnumber 支持 GBK, Big5 和 UTF8 编码, 依赖 L^AT_EX 3 项目的 `expl3`, `xparse` 和 `l3keys2e` 宏包。

2 使用方法

encoding

Updated: 2012-5-25

encoding = `<GBK|Big5|UTF8>`

用于指定编码的宏包选项, 可以在调用宏包的时候设定, 也可以用 `\zhnumsetup` 在导言区内设定。对于 X_YL^AT_EX 和 LuaL^AT_EX, 不用指定编码, 宏包将自动使用 UTF8 编码。只有 L^AT_EX 和 pdfL^AT_EX 需要指定编码, 如果没有指定, 默认将使用 GBK, 并且此时 zhnumber 宏包应该在 CJK 或 CJKutf8 宏包之后载入。

`\zhnumber`

`\zhnumber <{number}>`

以中文格式输出数字。这里的数字可以是整数、小数和分数。例如

二十亿零一千二百零二万零一百二十
二十亿零一千二百零二万零一百二十
二十亿零一千二百零二万零一百二十
二千零一十二点零二零一二零
二千零一十二点零
零点二零一二
二万零一百二十分之二万零一百二十
二千零一十二分之零
零分之二千零一十二
二百零一又一百二十分之二千零二十

```
1 \zhnumber{2012020120}\\  
2 \zhnumber{2 012 020 120}\\  
3 \zhnumber{2,012,020,120}\\  
4 \zhnumber{2012.020120}\\  
5 \zhnumber{2012.}\\  
6 \zhnumber{.2012}\\  
7 \zhnumber{20120/20120}\\  
8 \zhnumber{/2012}\\  
9 \zhnumber{2012/}\\  
10 \zhnumber{201;2020/120}
```

`\zhdigits`

`\zhdigits <{number}>`

将阿拉伯数字转换为中文数字串。缺省状态下, `\zhdigits` 将 0 映射为 〇, 如果需要将其映射为零, 可以使用 `\zhdigits*`。例如

二〇一二〇二〇一二〇
二零一二零二零一二零

```
1 \zhdigits{2012020120}\\  
2 \zhdigits*{2012020120}
```

`\zhnum`

`\zhnum <{counter}>`

与 `\roman` 等类似, 用于将 L^AT_EX 计数器的值转换为中文数字。例如

二

```
1 \zhnum{section}
```

<code>\zhweekday</code>	<code>\zhweekday {<yyyy/mm/dd>}</code>	
New: 2012-5-25	输出日期当天的星期。例如	
	星期日	1 <code>\zhweekday{2012/5/20}</code>
<code>\zhdate</code>	<code>\zhdate {<yyyy/mm/dd>}</code>	
	<code>\zhdate* {<yyyy/mm/dd>}</code>	
New: 2012-5-25	以中文格式输出日期, 其中带 <code>*</code> 的命令还输出星期。例如	
	2012 年 5 月 21 日	1 <code>\zhdate{2012/5/21}\\</code>
	2012 年 5 月 21 日星期一	2 <code>\zhdate*{2012/5/21}</code>
<code>\zhtoday</code>	与 <code>\today</code> 类似, 以中文输出当天的日期。例如	
New: 2012-5-25	2012 年 11 月 23 日	1 <code>\zhtoday</code>
<code>\zhtime</code>	<code>\zhtime {<hh:mm>}</code>	
New: 2012-5-25	以中文格式输出时间。例如	
	23 时 56 分	1 <code>\zhtime{23:56}</code>
<code>\zhcurrtime</code>	输出当前的时间。例如	
New: 2012-5-25	18 时 52 分	1 <code>\zhcurrtime</code>
<code>\zhnumExtendScaleMap</code>	<code>\zhnumExtendScaleMap [<character>] {<character 1>, <character 2>, ..., <character n>}</code>	
New: 2012-5-25	缺省状态下 <code>\zhnumber</code> 能正确中文格式化的最大整数是 $10^{48} - 1$, <code>\zhditits</code> 不受这个大小的限制。可以通过 <code>\zhnumExtendScaleMap</code> 来扩展 <code>\zhnumber</code> 。<character> <i>i</i> 设置 $10^{4(i+11)}$ 。若给出可选项 <character>, 则当数字大于 $10^{4(n+12)} - 1$ 时, 统一用 <character> 设置输出数字的进位。	
<code>\zhnumsetup</code>	<code>\zhnumsetup {<key1>=<var1>, <key2>=<var2>, ...}</code>	
	用于在导言区或文档中, 设置中文数字的输出格式。目前可以设置的 <key> 如下介绍。	
<code>time</code>	<code>time = {<Arabic> <Chinese>}</code>	
New: 2012-5-25	设置日期和时间的数字格式, <Arabic> 为阿拉伯数字, 而 <Chinese> 为中文数字。默认使用阿拉伯数字。例如	
	二〇一二年十一月二十三日十八时五十二分	1 <code>\zhnumsetup{time=Chinese}</code> 2 <code>\zhtoday\zhcurrtime</code>
<code>style</code>	<code>style = {<Simplified> <Traditional> <Normal> <Financial> <Ancient>}</code>	
Updated: 2012-5-25	意义分别为	
	Simplified 以简体中文输出数字(对 Big5 编码无效);	
	Traditional 以繁体中文输出数字(对 Big5 编码无效);	
	Normal 以小写形式输出中文数字;	
	Financial 以大写形式输出中文数字;	
	Ancient 以廿输出 20, 以卅输出 30, 以卅输出 40, 以佰输出 200。	
	可以设置 <code>style</code> 为其中一个, 也可以是前三个与后两个的适当组合, 默认是简体小写。例如	
	陸萬貳仟零壹拾貳點叁 廿一	1 <code>\zhnumsetup{style={Traditional,Financial}}</code> 2 <code>\zhnumber{62012.3}\\</code> 3 <code>\zhnumsetup{style=Ancient}</code> 4 <code>\zhnumber{21}</code>

`null` `null = <true|false>`

缺省状态下,除了 `\zhdigits` 外,其它的格式转换命令,将 0 映射成零,如果需要将 0 映射成○,可以使用这个选项。

`zhnumber` 提供下列选项来控制阿拉伯数字的中文映射。

```
- -0 0 1 2 3 4 5 6 7 8 9 10 20 30 40 200
E2 E3 E4 E8 E12 E16 E20 E24 E28 E32 E36 E40 E44
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 FE2 FE3
dot and parts
year month day hour minute weekday mon tue wed thu fri sat sun
```

其中 `-` 设置负, `-0` 设置○, `dot` 设置小数的点, `and` 和 `parts` 分别设置分数的“又”和“分之”, `En` 设置 10^n , 而 `Fn` 设置数字 n 的大写。其它的选项同字面意思,不再赘述。例如

```
\zhnumsetup{2={两}}
```

可以将 2 映射成两。需要说明的是, `zhnumber` 将优先使用这里的设置,所以可能会影响到 `style` 选项。如果要恢复 `style` 的功能,可以使用 `reset` 选项。

`reset`

Updated: 2012-5-25

`reset`

用于恢复 `zhnumber` 对阿拉伯数字的初始化映射。`zhnumber` 的中文数字初始化设置见源代码(第 4 节)。

`\zhnumber`
`\zhdigits`
`\zhnum`

```
\zhnumber [<options>] {<number>}
\zhdigits [<options>] {<number>}
\zhnum [<options>] {<counter>}
```

如果只改变当前数字的中文输出格式,可以使用带选项的格式转换命令,其中 `<options>` 与 `\zhnumsetup` 的参数相同,如上所介绍。这些带了选项的命令是不可展开的,在某些场合使用时要小心。

3 zhnumber 宏包代码实现

```
1 <*package>
2 \msg_new:nnn { zhnumber } { l3-too-old }
3 {
4   Support~package~'expl3'~too~old. \\\
5   Please~update~an~up~to~date~version~of~the~bundles\\\
6   'l3kernel'~and~'l3packages'\\\
7   using~your~TeX~package~manager~or~from~CTAN.
8 }
9 \@ifpackagelater { expl3 } { 2012/07/16 } { }
10 { \msg_error:nn { zhnumber } { l3-too-old } }
11 \RequirePackage{xparse}
12 \RequirePackage{l3keys2e}
```

`\zhnumber` 用于将输入的数字按照中文格式输出。

```
13 \DeclareExpandableDocumentCommand \zhnumber { o m }
14 { \zhnum_number_aux:nNn {#1} \zhnum_number:n {#2} }
15 \cs_new_nopar:Nn \zhnum_number:n { \zhnum_number:w #1 . \q_nil . \q_stop }
(End definition for \zhnumber This function is documented on page 3.)
```

`\zhnum_number_aux:nNn`

```
16 \cs_new_nopar:Nn \zhnum_number_aux:nNn
17 {
18   \IfNoValueTF {#1} { #2 {#3} }
19   { \group_begin: \zhnumsetup {#1} #2 {#3} \group_end: }
20 }
(End definition for \zhnum_number_aux:nNn)
```

`\zhnum_number:w` 先判断输入的是小数还是分数。

```
21 \cs_new_nopar:Npn \zhnum_number:w #1.#2.#3 \q_stop
22 {
23   \quark_if_nil:nTF {#2}
24     { \zhnum_integer_or_fraction:w #1 / \q_nil / \q_stop }
25     { \zhnum_decimal:nn {#1} {#2} }
26 }
```

(End definition for \zhnum_number:w)

`\zhnum_integer_or_fraction:w` 判断是否输入的是分数。

```
27 \cs_new_nopar:Npn \zhnum_integer_or_fraction:w #1/#2/#3 \q_stop
28 {
29   \quark_if_nil:nTF {#2}
30     { \zhnum_integer:f {#1} }
31     { \zhnum_fraction:w #2 \q_mark #1 ; \q_nil ; \q_stop }
32 }
```

(End definition for \zhnum_integer_or_fraction:w)

`\zhnum_fraction:w` 对分数进行预处理。

```
33 \cs_new_nopar:Npn \zhnum_fraction:w #1 \q_mark #2;#3;#4 \q_stop
34 {
35   \quark_if_nil:nTF {#3}
36     {
37       \zhnum_blank_to_zero:f {#1} \c_zhnum_parts_tl
38       \zhnum_blank_to_zero:f {#2}
39     }
40     {
41       \tl_if_blank:fF {#2} { \zhnumber {#2} \c_zhnum_and_tl }
42       \zhnum_blank_to_zero:f {#1} \c_zhnum_parts_tl
43       \zhnum_blank_to_zero:f {#3}
44     }
45 }
```

`\cs_generate_variant:Nn \tl_if_blank:nF { f }`

(End definition for \zhnum_fraction:w)

`\zhnum_decimal:nn` 对小数进行预处理。

```
47 \cs_new_nopar:Nn \zhnum_decimal:nn
48 {
49   \zhnum_blank_to_zero:f {#1} \c_zhnum_dot_tl
50   \tl_if_blank:fTF {#2} { \c_zhnum_zero_tl } { \zhdigits * {#2} }
51 }
```

`\cs_generate_variant:Nn \tl_if_blank:nTF { f }`

(End definition for \zhnum_decimal:nn)

`\zhnum_blank_to_zero:n` 输出小数的整数位。

```
53 \cs_new_nopar:Nn \zhnum_blank_to_zero:n
54 { \tl_if_blank:nTF {#1} { \zhnum_digit_map:n \c_zero } { \zhnumber {#1} } }
55 \cs_generate_variant:Nn \zhnum_blank_to_zero:n { f }
```

(End definition for \zhnum_blank_to_zero:n)

\zhnum 用于将 **L^AT_EX** 计数器按中文格式输出。

```
56 \DeclareExpandableDocumentCommand \zhnum { o m }
57 { \zhnum_number_aux:nNn {#1} \zhnum_counter:n {#2} }
58 \cs_new_nopar:Nn \zhnum_counter:n
59 {
60   \exp_args:Nc \token_if_int_register:NTF { c@#1 }
61     { \zhnum_integer:v { c@#1 } }
62     { \@nocounterr {#1} }
63 }
```

(End definition for \zhnum This function is documented on page 3.)

\zhnum_integer:n 对整数的处理。

```
64 \cs_new_nopar:Nn \zhnum_integer:n
65 {
66   \int_compare:nNt { \__int_get_sign:n {#1} \c_one } < \c_zero { \c_zhnum_minus_tl }
67   \zhnum_parse_number:f { \zhnum_trim_zeros:f { \zhnum_erase_separator:n {#1} } }
68 }
69 \cs_generate_variant:Nn \zhnum_integer:n { f , v , V }
(End definition for \zhnum_integer:n)
```

\zhnum_erase_separator:n 去掉分隔符和多余的 0。

```
\zhnum_trim_zeros:n
70 \cs_new_nopar:Nn \zhnum_erase_separator:n
71 { \cs_to_str:c { \tl_map_function:nN {#1} \zhnum_erase_separator_aux:N } }
72 \cs_new_nopar:Nn \zhnum_erase_separator_aux:N { \zhnum_if_number:NT {#1} {#1} }
73 \cs_new_nopar:Nn \zhnum_trim_zeros:n
74 {
75   \tl_if_empty:nTF {#1} \c_zero
76   {
77     \int_compare:nNtTF { \tl_head:n {#1} } = \c_zero
78     { \zhnum_trim_zeros:o { \use_none:n #1 } } {#1}
79   }
80 }
81 \prg_new_conditional:Nnn \zhnum_if_number:N { p , T , F , TF }
82 {
83   \if_int_compare:w \c_one < 1 #1 \exp_stop_f:
84   \prg_return_true: \else: \prg_return_false: \fi:
85 }
86 \cs_generate_variant:Nn \zhnum_trim_zeros:n { f , o }
87 \cs_generate_variant:Nn \cs_to_str:N { c }
(End definition for \zhnum_erase_separator:n and \zhnum_trim_zeros:n)
```

\zhnum_parse_number:n

```
\zhnum_parse_number:nn
88 \cs_new_nopar:Nn \zhnum_parse_number:n
89 {
90   \zhnum_parse_number:nff {#1}
91   { \tl_count:n {#1} }
92   { \int_mod:nn { \tl_count:n {#1} } \c_four }
93 }
94 \cs_new_nopar:Nn \zhnum_parse_number:nnn
95 {
96   \int_compare:nNtTF {#2} < \c_two
97   { \zhnum_digit_map:n {#1} }
98   {
99     \zhnum_split_number:fNNfn { \zhnum_insert_zeros:n {#3} #1 }
100     \c_true_bool \c_true_bool
101     {
102       \int_compare:nNtTF {#3} = \c_zero
103       { \int_eval:n { (#2) / \c_four - \c_one } }
104       { \int_div_truncate:nn {#2} \c_four }
105     }
106     { \c_zero }
107   }
108 }
109 \cs_generate_variant:Nn \zhnum_parse_number:n { f }
110 \cs_generate_variant:Nn \zhnum_parse_number:nnn { nff }
(End definition for \zhnum_parse_number:n and \zhnum_parse_number:nn)
```

\zhnum_insert_zeros:n 为了处理的方便,在整数前面补上适当的 0,使其位数可以被 4 整除。

```
111 \cs_new_nopar:Nn \zhnum_insert_zeros:n
112 {
113   \if_case:w \int_eval:n { #1 - \c_one } \exp_stop_f:
114   \or: \exp_after:wN \use_none:n
115   \or: \exp_after:wN \use_none:nn
```

```

116     \else: \exp_after:wN \use_none:nnn
117     \fi:
118     000
119 }
(End definition for \zhnum_insert_zeros:n)

```

\zhnum_split_number:nNNnn 将输入的整数由高位到低位,以四位为一段进行处理。

```

120 \cs_new_nopar:Nn \zhnum_split_number:nNNnn
121 {
122     \zhnum_split_number_aux:fnNNnn { \zhnum_number_item:nn {#1} {#5} }
123     {#1} #2#3 {#4} {#5}
124 }
125 \cs_new_nopar:Nn \zhnum_split_number_aux:nnNNnn
126 {
127     \int_compare:nNnTF {#1} = \c_zero { \use_i_ii:nnn }
128     {
129         \bool_if:NF #3 { \c_zhnum_zero_tl }
130         \zhnum_process_number:NNn #3#4 {#1}
131         \zhnum_scale_map:n { #5 - #6 }
132         \int_compare:nNnTF { \int_mod:nn {#1} \c_ten } = \c_zero
133         { \use_i_ii:nnn } { \zhnum_use_i_iii:nnn }
134     }
135     { \int_compare:nNnF {#5} = {#6} }
136     { {
137         \zhnum_split_number:nNNnf
138         {#2} \c_false_bool \c_true_bool {#5} { \int_eval:n { #6 + \c_one } }
139     } }
140     { {
141         \zhnum_split_number:nNNnf
142         {#2} \c_true_bool \c_false_bool {#5} { \int_eval:n { #6 + \c_one } }
143     } }
144 }
145 \cs_new_nopar:Nn \zhnum_use_i_iii:nnn {#1#3}
146 \cs_generate_variant:Nn \zhnum_split_number:nNNnn { fNNf , nNNnf }
147 \cs_generate_variant:Nn \zhnum_split_number_aux:nnNNnn { f }
(End definition for \zhnum_split_number:nNNnn)

```

\zhnum_number_item:nn 截取整数的其中四位数。

```

\zhnum_number_item_aux:nNNNN 148 \cs_new_nopar:Nn \zhnum_number_item:nn
149 { \zhnum_number_item_aux:nNNNN {#2} #1 \q_recursion_stop }
150 \cs_new_nopar:Nn \zhnum_number_item_aux:nNNNN
151 {
152     \int_compare:nNnTF {#1} = \c_zero
153     { \zhnum_recursion_stop:NNNNw #2#3#4#5 }
154     { \zhnum_number_item_aux:fNNNN { \int_eval:n { #1 - \c_one } } }
155 }
156 \cs_generate_variant:Nn \zhnum_number_item_aux:nNNNN { f }
157 \cs_new_nopar:Npn \zhnum_recursion_stop:NNNNw #1#2#3#4#5 \q_recursion_stop {#1#2#3#4}
(End definition for \zhnum_number_item:nn and \zhnum_number_item_aux:nNNNN)

```

\zhnum_process_number:NNn 对四位数字按情况进行处理。

```

\zhnum_process_number:NNNNNN 158 \cs_new_nopar:Nn \zhnum_process_number:NNn
159 { \zhnum_process_number:NNNNNN #3#1#2 }
160 \cs_new_nopar:Nn \zhnum_process_number:NNNNNN
161 {
162     \int_compare:nNnTF {#1} = \c_zero
163     { \bool_if:NF #6 { \c_zhnum_zero_tl } }
164     { \zhnum_digit_map:n {#1} \c_zhnum_thousand_tl }
165     \int_compare:nNnTF {#2} = \c_zero
166     { \int_compare:nNnT { #1 * (#3#4) } > \c_zero { \c_zhnum_zero_tl } }
167     {
168         \bool_if:nTF

```

```

169         { \l_zhnum_ancient_bool && \int_compare_p:nNn {#2} = \c_two }
170         { \zhnum_digit_map:n { #2 00 } }
171         { \zhnum_digit_map:n {#2} \c_zhnum_hundred_tl }
172     }
173     \int_compare:nNnTF {#3} = \c_zero
174     { \int_compare:nNnT { #2 * #4 } > \c_zero { \c_zhnum_zero_tl } }
175     {
176         \bool_if:nF
177         {
178             \int_compare_p:nNn {#3} = \c_one &&
179             \int_compare_p:nNn {#1#2} = \c_zero && #6 && #5
180         }
181         {
182             \bool_if:nTF
183             {
184                 \l_zhnum_ancient_bool &&
185                 ( \int_compare_p:nNn {#3} = \c_two ||
186                   \int_compare_p:nNn {#3} = \c_three ||
187                   \int_compare_p:nNn {#3} = \c_four )
188             }
189             { \zhnum_digit_map:n { #3 0 } \use_none:n }
190             { \zhnum_digit_map:n {#3} }
191         }
192         \c_zhnum_ten_tl
193     }
194     \int_compare:nNnF {#4} = \c_zero { \zhnum_digit_map:n {#4} }
195 }
(End definition for \zhnum_process_number:NNn and \zhnum_process_number:NNNNN)

```

\zhdigits 将输入的数字输出为中文数字串输出。

```

196 \DeclareExpandableDocumentCommand \zhdigits { s o m }
197 {
198     \IfBooleanTF {#1}
199     { \zhnum_digits_aux:nnN {#2} {#3} \zhnum_digits_zero_aux:N }
200     { \zhnum_digits_aux:nnN {#2} {#3} \zhnum_digits_null_aux:N }
201 }
202 \cs_new_nopar:Nn \zhnum_digits_aux:nnN
203 {
204     \IfNoValueTF {#1} { \tl_map_function:fn {#2} #3 }
205     { \group_begin: \zhnumsetup {#1} \tl_map_function:fn {#2} #3 \group_end: }
206 }
207 \cs_new_nopar:Nn \zhnum_digits_null:n { \tl_map_function:fn {#1} \zhnum_digits_null_aux:N }
208 \cs_generate_variant:Nn \zhnum_digits_null:n { V }
209 \cs_generate_variant:Nn \tl_map_function:nN { f }
(End definition for \zhdigits This function is documented on page 3.)

```

\zhnum_digits_null_aux:N 将输入的数字输出为中文数字串输出。

```

\zhnum_digits_zero_aux:N
\zhnum_digits_aux:NN
210 \cs_new_nopar:Nn \zhnum_digits_null_aux:N { \zhnum_digits_aux:NN \c_true_bool #1 }
211 \cs_new_nopar:Nn \zhnum_digits_zero_aux:N { \zhnum_digits_aux:NN \c_false_bool #1 }
212 \cs_new_nopar:Nn \zhnum_digits_aux:NN
213 {
214     \str_if_eq:x:nnTF {#2} . \c_zhnum_dot_tl
215     {
216         \str_if_eq:x:nnTF {#2} -
217         { \c_zhnum_minus_tl }
218         {
219             \zhnum_if_number:NT {#2}
220             {
221                 \bool_if:nTF { \int_compare_p:nNn {#2} = \c_zero }
222                 { \bool_if:NTF #1 \c_zhnum_null_tl \c_zhnum_zero_tl }
223                 { \zhnum_digit_map:n {#2} }
224             }

```

```

225     }
226   }
227 }
(End definition for \zhnum_digits_null_aux:N, \zhnum_digits_zero_aux:N, and \zhnum_digits_aux:NN)

```

\zhdate 输出中文日期。

```

228 \DeclareExpandableDocumentCommand \zhdate { s m }
229 {
230   \zhnum_date:w #2 \q_stop
231   \IfBooleanT {#1} { \zhweekday {#2} }
232 }
233 \cs_new_nopar:Npn \zhnum_date:w #1/#2/#3 \q_stop
234 {
235   \zhnum_check_time:Nn \zhnum_digits_null:n {#1} \c_zhnum_year_tl
236   \zhnum_check_time:Nn \zhnum_integer:f {#2} \c_zhnum_month_tl
237   \zhnum_check_time:Nn \zhnum_integer:f {#3} \c_zhnum_day_tl
238 }
(End definition for \zhdate This function is documented on page 2.)

```

\zhtoday 输出当天日期。

```

239 \cs_new_nopar:Npn \zhtoday
240 {
241   \zhnum_check_time:Nn \zhnum_digits_null:V \tex_year:D \c_zhnum_year_tl
242   \zhnum_check_time:Nn \zhnum_integer:V \tex_month:D \c_zhnum_month_tl
243   \zhnum_check_time:Nn \zhnum_integer:V \tex_day:D \c_zhnum_day_tl
244 }
(End definition for \zhtoday This function is documented on page 2.)

```

\zhnum_check_time:Nn 判断是用中文数字还是用阿拉伯数组。

```

245 \cs_new_nopar:Nn \zhnum_check_time:Nn
246 { \bool_if:NTF \l_zhnum_time_bool {#1} { \int_eval:n } {#2} }
(End definition for \zhnum_check_time:Nn)

```

\zhweekday 输出星期

```

247 \cs_new_nopar:Npn \zhweekday #1 { \zhnum_week_day:w #1 \q_stop }
(End definition for \zhweekday This function is documented on page 2.)

```

\zhnum_Zeller:nnn 用 Zeller 公式¹ 计算星期几。

```

\zhnum_Zeller_aux:Nnnn
\zhnum_two_digits:n
248 \cs_new_nopar:Nn \zhnum_Zeller:nnn
249 {
250   \int_compare:nNnTF
251     { #1 \zhnum_two_digits:n {#2} \zhnum_two_digits:n {#3} } > { 1582 10 04 }
252     { \zhnum_Zeller_aux:Nnnn \zhnum_Zeller_Gregorian:nnn }
253     { \zhnum_Zeller_aux:Nnnn \zhnum_Zeller_Julian:nnn }
254     {#1} {#2} {#3}
255 }
256 \cs_new_nopar:Nn \zhnum_Zeller_aux:Nnnn
257 {
258   \int_compare:nNnTF {#3} < \c_three
259     { #1 { #2 - \c_one } { #3 + \c_twelve } {#4} }
260     { #1 {#2} {#3} {#4} }
261 }
262 \cs_new_nopar:Nn \zhnum_two_digits:n
263 {
264   \int_compare:nNnT {#1} < \c_ten { 0 }
265   \int_eval:n {#1}
266 }
(End definition for \zhnum_Zeller:nnn, \zhnum_Zeller_aux:Nnnn, and \zhnum_two_digits:n)

```

¹http://en.wikipedia.org/wiki/Zeller's_congruence

\zhnum_Zeller_Gregorian:nnn 格里历(1582 年 10 月 15 日及以后)的计算公式

$$h = \left(q + \left\lfloor \frac{26(m+1)}{10} \right\rfloor + Y + \left\lfloor \frac{Y}{4} \right\rfloor + 6 \left\lfloor \frac{Y}{100} \right\rfloor + \left\lfloor \frac{Y}{400} \right\rfloor \right) \pmod{7}$$

其中 Y 为年, m 为月, q 为日; 若 $m = 1, 2$, 则令 $m += 12$, 同时 $Y --$ 。

```
267 \cs_new_nopar:Nn \zhnum_Zeller_Gregorian:nnn
268 {
269   \int_mod:nn
270   {
271     \int_eval:n {#3}
272     + \int_div_truncate:nn { 26 * ( \int_eval:n {#2} + \c_one ) } \c_ten
273     + \int_eval:n {#1}
274     + \int_div_truncate:nn {#1} \c_four
275     + \c_six * \int_div_truncate:nn {#1} \c_one_hundred
276     + \int_div_truncate:nn {#1} { 400 }
277   }
278   { \c_seven }
279 }
(End definition for \zhnum_Zeller_Gregorian:nnn)
```

\zhnum_Zeller_Julian:nnn 儒略历(1582 年 10 月 4 日及以前)的计算公式

$$h = \left(q + \left\lfloor \frac{26(m+1)}{10} \right\rfloor + Y + \left\lfloor \frac{Y}{4} \right\rfloor + 5 \right) \pmod{7}$$

```
280 \cs_new_nopar:Nn \zhnum_Zeller_Julian:nnn
281 {
282   \int_mod:nn
283   {
284     \int_eval:n {#3}
285     + \int_div_truncate:nn { 26 * ( \int_eval:n {#2} + \c_one ) } \c_ten
286     + \int_eval:n {#1}
287     + \int_div_truncate:nn {#1} \c_four
288     + \c_five
289   }
290   { \c_seven }
291 }
(End definition for \zhnum_Zeller_Julian:nnn)
```

\zhnum_week_day:w 用 Zeller 公式计算的结果 h 与实际星期的关系是 $d = h + 5 \pmod{7} + 1$ 。

```
292 \cs_new_nopar:Npn \zhnum_week_day:w #1/#2/#3 \q_stop
293 {
294   \if_case:w \int_eval:n { \zhnum_Zeller:nnn {#1} {#2} {#3} } \exp_stop_f:
295     \c_zhnum_sat_tl
296     \or: \c_zhnum_sun_tl
297     \or: \c_zhnum_mon_tl
298     \or: \c_zhnum_tue_tl
299     \or: \c_zhnum_wed_tl
300     \or: \c_zhnum_thu_tl
301     \or: \c_zhnum_fri_tl
302   \fi:
303 }
(End definition for \zhnum_week_day:w)
```

\zhtime 输出时间。

```
304 \cs_new_nopar:Npn \zhtime #1 { \zhnum_time:w #1 \q_stop }
305 \group_begin:
306 \char_set_lccode:nn { \; } { \: }
307 \tl_to_lowercase:n
308 {
309   \group_end:
310   \cs_new_nopar:Npn \zhnum_time:w #1 ; #2 \q_stop
```

```

311 {
312   \zhnum_check_time:Nn \zhnum_integer:f {#1} \c_zhnum_hour_tl
313   \zhnum_check_time:Nn \zhnum_integer:f {#2} \c_zhnum_minute_tl
314 }
315 }

```

(End definition for \zhnum This function is documented on page 2.)

\zhcurrtime 输出当前时间。

```

316 \cs_new_nopar:Npn \zhcurrtime
317 {
318   \zhnum_check_time:Nn \zhnum_integer:f
319   { \int_div_truncate:nn \tex_time:D { 60 } } \c_zhnum_hour_tl
320   \zhnum_check_time:Nn \zhnum_integer:f
321   { \int_mod:nn \tex_time:D { 60 } } \c_zhnum_minute_tl
322 }

```

(End definition for \zhcurrtime This function is documented on page 2.)

\zhnum_digit_map:n 阿拉伯数字与中文数字的映射。

```

323 \cs_new_nopar:Nn \zhnum_digit_map:n
324 { \tl_use:c { c_zhnum_ \zhnum_int_to_word:n {#1} _tl } }

```

(End definition for \zhnum_digit_map:n)

\zhnum_scale_map:n 大数系统的映射。

```

\zhnum_scale_map_loop:n 325 \cs_new_nopar:Nn \zhnum_scale_map:n
326 {
327   \cs_if_exist_use:cF { c_zhnum_scale_ \zhnum_int_to_word:n {#1} _tl }
328   { \zhnum_scale_map_hook:n {#1} }
329 }
330 \cs_new_nopar:Nn \zhnum_scale_map_loop:n
331 { \zhnum_scale_map:n { \int_mod:nn {#1} \g_zhnum_scale_int } }
332 \int_new:N \g_zhnum_scale_int
333 \int_set_eq:NN \g_zhnum_scale_int \c_eleven
334 \cs_new_eq:NN \zhnum_scale_map_hook:n \zhnum_scale_map_loop:n

```

(End definition for \zhnum_scale_map:n and \zhnum_scale_map_loop:n)

\zhnumExtendScaleMap

```

335 \NewDocumentCommand \zhnumExtendScaleMap { > { \TrimSpaces } o m }
336 {
337   \int_zero:N \l_tmpa_int
338   \clist_map_inline:nn {#2}
339   {
340     \int_incr:N \l_tmpa_int
341     \tl_set:Nx \l_tmpa_tl
342     { c_zhnum_scale_ \zhnum_int_to_word:n { \l_tmpa_int + \c_eleven } _tl }
343     \tl_if_exist:cF \l_tmpa_tl { \int_incr:N \g_zhnum_scale_int }
344     \tl_set:cn { \l_tmpa_tl } {##1}
345   }
346   \IfValueT {#1} { \cs_set:Nn \zhnum_scale_map_hook:n {#1} }
347 }

```

(End definition for \zhnumExtendScaleMap This function is documented on page 2.)

\zhnum_int_to_word:n 将整数转换成英文单词。

```

348 \cs_new_nopar:Nn \zhnum_int_to_word:n
349 {
350   \if_case:w \int_eval:n {#1} \exp_stop_f:
351     zero
352   \or: one
353   \or: two
354   \or: three
355   \or: four
356   \or: five
357   \or: six

```

```

358     \or: seven
359     \or: eight
360     \or: nine
361     \or: ten
362     \or: eleven
363     \else:
364     \int_case:nnn {#1}
365     {
366         { 20 } { twenty } { 30 } { thirty }
367         { 40 } { forty } { 200 } { two_hundred }
368     }
369     { \int_to_roman:n {#1} }
370 \fi:
371 }
(End definition for \zhnum_int_to_word:n)
    根据需要设置中文阿拉伯数字。
372 \keys_define:nn { zhnum / options }
373 {
374     - .tl_set:N = \l_zhnum_minus_tl      ,
375     -0 .tl_set:N = \l_zhnum_null_tl      ,
376     E2 .tl_set:N = \l_zhnum_hundred_tl   ,
377     E3 .tl_set:N = \l_zhnum_thousand_tl  ,
378     FE2 .tl_set:N = \l_zhnum_financial_hundred_tl ,
379     FE3 .tl_set:N = \l_zhnum_financial_thousand_tl ,
380 }
381 \clist_map_inline:nn
382 { 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 20 , 30 , 40 , 200 }
383 {
384     \keys_define:nn { zhnum / options }
385     { #1 .tl_set:c = { l_zhnum_ \zhnum_int_to_word:n {#1} _tl } }
386     \int_compare:nNnF {#1} > \c_ten
387     {
388         \keys_define:nn { zhnum / options }
389         { F#1 .tl_set:c = { l_zhnum_financial_ \zhnum_int_to_word:n {#1} _tl } }
390     }
391 }
392 \clist_map_inline:nn
393 { 4 , 8 , 12 , 16 , 20 , 24 , 28 , 32 , 36 , 40 , 44 }
394 {
395     \keys_define:nn { zhnum / options }
396     { E#1 .tl_set:c = { l_zhnum_scale_ \zhnum_int_to_word:n { #1 / 4 } _tl } }
397 }
398 \clist_map_inline:nn
399 {
400     dot , and , parts , year , month , day , weekday , hour , minute
401     mon , tue , wed , thu , fri , sat , sun
402 }
403 { \keys_define:nn { zhnum / options } { #1 .tl_set:c = { l_zhnum_ #1 _tl } } }

```

```

\zhnum_parse_config: 将配置文件中的中文数字保存起来。
\zhnum_check_simp:nn 404 \cs_new_nopar:Nn \zhnum_parse_config:
\zhnum_check_financial:nn 405 {
    \zhnum_set_zero: 406 \prop_map_function:NN \g_zhnum_cfg_map_prop \zhnum_check_simp:nn
    \zhnum_set_week_day: 407 \prop_map_function:NN \g_zhnum_cfg_map_prop \zhnum_check_financial:nn
    408 \zhnum_set_zero:
    409 \zhnum_set_week_day:
    410 }
    411 \cs_new_nopar:Nn \zhnum_check_simp:nn
    412 {
    413     \zhnum_check_simp_aux:nn {#1} {#2}
    414     \prop_get:NnNT \g_zhnum_cfg_map_finan_prop {#1} \l_tmpa_tl
    415     { \exp_args:NnV \zhnum_check_simp_aux:nn { financial_ #1 } \l_tmpa_tl }

```

```

416     }
417 \cs_new_nopar:Nn \zhnum_check_simp_aux:nn
418 {
419     \prop_get:NnNTF \g_zhnum_cfg_map_var_prop { #1 _trad } \l_tmpa_tl
420     {
421         \prop_get:NnNTF \g_zhnum_cfg_map_var_prop { #1 _simp } \l_tmpb_tl
422         {
423             \tl_set:cx { l_zhnum_ #1 _tl }
424             {
425                 \exp_not:n { \bool_if:NTF \l_zhnum_simp_bool }
426                 { \exp_not:V \l_tmpb_tl } { \exp_not:V \l_tmpa_tl }
427             }
428         }
429         {
430             \tl_set:cx { l_zhnum_ #1 _tl }
431             {
432                 \exp_not:n { \bool_if:NTF \l_zhnum_simp_bool }
433                 { \exp_not:n {#2} } { \exp_not:V \l_tmpa_tl }
434             }
435         }
436     }
437     { \tl_set:cn { l_zhnum_ #1 _tl } {#2} }
438 }
439 \cs_new_nopar:Nn \zhnum_check_financial:nn
440 {
441     \prop_get:NnNTF \g_zhnum_cfg_map_finan_prop {#1} \l_tmpa_tl
442     {
443         \tl_set:cx { c_zhnum_ #1 _tl }
444         {
445             \exp_not:n { \bool_if:NTF \l_zhnum_normal_bool }
446             { \exp_not:c { l_zhnum_ #1 _tl } }
447             { \exp_not:c { l_zhnum_financial_ #1 _tl } }
448         }
449     }
450     { \tl_set:cx { c_zhnum_ #1 _tl } { \exp_not:c { l_zhnum_ #1 _tl } } }
451 }
452 \cs_new_nopar:Nn \zhnum_set_zero:
453 {
454     \tl_set:Nx \l_zhnum_zero_tl
455     {
456         \exp_not:n { \bool_if:nTF \l_zhnum_null_bool }
457         { \exp_not:V \l_zhnum_null_tl } { \exp_not:V \l_zhnum_zero_tl }
458     }
459 }
460 \cs_new_nopar:Nn \zhnum_set_week_day:
461 {
462     \tl_set:Nx \l_zhnum_mon_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_one_tl }
463     \tl_set:Nx \l_zhnum_tue_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_two_tl }
464     \tl_set:Nx \l_zhnum_wed_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_three_tl }
465     \tl_set:Nx \l_zhnum_thu_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_four_tl }
466     \tl_set:Nx \l_zhnum_fri_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_five_tl }
467     \tl_set:Nx \l_zhnum_sat_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_six_tl }
468     \tl_set:Nx \l_zhnum_sun_tl { \exp_not:N \c_zhnum_weekday_tl \exp_not:V \l_zhnum_day_tl }
469     \clist_map_inline:nn { mon , tue , wed , thu , fri , sat , sun }
470     { \tl_set:cx { c_zhnum_ ##1 _tl } { \exp_not:c { l_zhnum_ ##1 _tl } } }
471 }

```

(End definition for \zhnum_parse_config: and others.)

\zhnum_load_cfg: 根据选定编码载入配置文件。

```

472 \cs_new:Nn \zhnum_load_cfg:n
473 {
474     \cs_if_exist:NT \CJK@makeActive

```

```

475     {
476         \int_compare:nNnTF { \char_value_catcode:n {"080} } = \active
477         { \bool_set_false:N \l_zhnum_set_CJK_active_bool }
478         { \bool_set_true:N \l_zhnum_set_CJK_active_bool \CJK@makeActive }
479     }
480     \prop_clear:N \g_zhnum_cfg_map_prop
481     \prop_clear:N \g_zhnum_cfg_map_var_prop
482     \prop_clear:N \g_zhnum_cfg_map_finan_prop
483     \file_input:n { zhnumber - #1 .cfg }
484     \zhnum_parse_config:
485     \bool_if:nT { \cs_if_exist_p:N \CJK@makeInactive && \l_zhnum_set_CJK_active_bool }
486     { \CJK@makeInactive }
487 }
(End definition for \zhnum_load_cfg:)

```

```

\zhnum_set_digits_map:nn
\zhnum_set_digits_map:nnn 488 \cs_new_protected_nopar:Nn \zhnum_set_digits_map:nn
\zhnum_set_financial_map:nn 489 { \prop_put:Nnn \g_zhnum_cfg_map_prop {#1} {#2} }
\zhnum_set_financial_map:nnn 490 \cs_new_protected_nopar:Nn \zhnum_set_digits_map:nnn
491 {
492     \prop_put_if_new:Nnn \g_zhnum_cfg_map_prop {#1} {#3}
493     \prop_put:Nnn \g_zhnum_cfg_map_var_prop {#1_#2} {#3}
494 }
495 \cs_new_protected_nopar:Nn \zhnum_set_financial_map:nn
496 { \prop_put:Nnn \g_zhnum_cfg_map_finan_prop {#1} {#2} }
497 \cs_new_protected_nopar:Nn \zhnum_set_financial_map:nnn
498 {
499     \prop_put_if_new:Nnn \g_zhnum_cfg_map_finan_prop {#1} {#3}
500     \prop_put:Nnn \g_zhnum_cfg_map_var_prop { financial_#1_#2 } {#3}
501 }
(End definition for \zhnum_set_digits_map:nn and others.)

```

```

\g_zhnum_cfg_map_prop
\g_zhnum_cfg_map_var_prop 502 \prop_new:N \g_zhnum_cfg_map_prop
\g_zhnum_cfg_map_finan_prop 503 \prop_new:N \g_zhnum_cfg_map_var_prop
504 \prop_new:N \g_zhnum_cfg_map_finan_prop
(End definition for \g_zhnum_cfg_map_prop, \g_zhnum_cfg_map_var_prop, and \g_zhnum_cfg_map_finan_prop)

```

encoding 宏包设置选项。

```

style 505 \keys_define:nn { zhnum / options }
null 506 {
reset 507     encoding .choice_code:n =
508     {
509         \tl_gset:Nx \c_zhnum_encoding_tl
510         { \exp_args:NV \tl_expandable_lowercase:n \l_keys_choice_tl }
511         \zhnum_load_cfg:n { \c_zhnum_encoding_tl }
512     } ,
513     encoding .generate_choices:n = { UTF8 , GBK , Big5 } ,
514     encoding .default:n = { GBK } ,
515     encoding / Bg5 .meta:n = { encoding = Big5 } ,
516     encoding / unknown .code:n =
517     { \msg_error:nnx { zhnumber } { encoding-invalid } { \l_keys_value_tl } } ,
518     style .multichoice: ,
519     style / Normal .code:n =
520     {
521         \bool_set_false:N \l_zhnum_ancient_bool
522         \bool_set_true:N \l_zhnum_normal_bool
523     } ,
524     style / Financial .code:n =
525     {
526         \bool_set_false:N \l_zhnum_ancient_bool
527         \bool_set_false:N \l_zhnum_normal_bool

```

```

528     } ,
529     style / Ancient      .code:n =
530     {
531         \bool_set_true:N \l_zhnum_ancient_bool
532         \bool_set_true:N \l_zhnum_normal_bool
533     } ,
534     style / Simplified   .code:n = { \bool_set_true:N \l_zhnum_simp_bool } ,
535     style / Traditional .code:n = { \bool_set_false:N \l_zhnum_simp_bool } ,
536     style                .default:n = { Normal , Simplified } ,
537     null                 .bool_set:N = \l_zhnum_null_bool ,
538     time .choice: ,
539     time / Chinese       .code:n = { \bool_set_true:N \l_zhnum_time_bool } ,
540     time / Arabic        .code:n = { \bool_set_false:N \l_zhnum_time_bool } ,
541     time                 .default:n = { Arabic } ,
542     reset                .code:n = { \zhnum_load_cfg:n { \c_zhnum_encoding_tl } } ,
543 }
544 \msg_new:nnn { zhnumber } { encoding-invalid }
545 {
546     The~encoding~'#1'~is~invalid,~please~check~it~over.\\
547     Available~encoding~are~'UTF8',~'GBK'~and~'Big5'.\\
548 }

```

(End definition for encoding and others. These functions are documented on page 3.)

\zhnumsetup 在文档中设置 zhnumber 的接口。

```

549 \NewDocumentCommand \zhnumsetup { m }
550 {
551     \keys_set:nn { zhnum / options } { #1 }
552     \tex_ignorespaces:D
553 }

```

(End definition for \zhnumsetup This function is documented on page 2.)

初始化设置和执行宏包选项。

```

554 \keys_set:nn { zhnum / options } { style , time }
555 \ProcessKeysOptions { zhnum / options }

```

如果没有选定编码,则根据引擎自动设置编码。

```

556 \ExplSyntaxOn
557 \tl_if_exist:NF \c_zhnum_encoding_tl
558 {
559     \exp_args:Nnx \keys_set:nn { zhnum / options }
560     {
561         encoding =
562         {
563             \bool_if:nTF { \xetex_if_engine_p: || \luatex_if_engine_p: }
564             { UTF8 } { GBK }
565         }
566     }
567 }
568 \ExplSyntaxOff
569 </package>

```

4 中文数字配置文件

```

1 <*config>
2 <!*big5>
3 \zhnum_set_digits_map:nnn { minus } { simp } { 负 }
4 \zhnum_set_digits_map:nnn { minus } { trad } { 負 }
5 </!big5>
6 <*big5>
7 \zhnum_set_digits_map:nn { minus } { 負 }
8 </big5>
9 \zhnum_set_digits_map:nn { zero } { 零 }

```

```

10 <*/big5>
11 \zhnum_set_digits_map:nn { null } { 〇 }
12 </!big5>
13 <*/big5>
14 \zhnum_set_digits_map:nn { null } { 〇 }
15 </big5>
16 \zhnum_set_digits_map:nn { one } { 一 }
17 \zhnum_set_digits_map:nn { two } { 二 }
18 \zhnum_set_digits_map:nn { three } { 三 }
19 \zhnum_set_digits_map:nn { four } { 四 }
20 \zhnum_set_digits_map:nn { five } { 五 }
21 \zhnum_set_digits_map:nn { six } { 六 }
22 \zhnum_set_digits_map:nn { seven } { 七 }
23 \zhnum_set_digits_map:nn { eight } { 八 }
24 \zhnum_set_digits_map:nn { nine } { 九 }
25 \zhnum_set_digits_map:nn { ten } { 十 }
26 \zhnum_set_digits_map:nn { hundred } { 百 }
27 \zhnum_set_digits_map:nn { thousand } { 千 }
28 \zhnum_set_digits_map:nn { twenty } { 廿 }
29 \zhnum_set_digits_map:nn { thirty } { 卅 }
30 \zhnum_set_digits_map:nn { forty } { 卌 }
31 \zhnum_set_digits_map:nn { two_hundred } { 兩 }
32 <*/big5>
33 \zhnum_set_digits_map:nnn { dot } { simp } { 点 }
34 \zhnum_set_digits_map:nnn { dot } { trad } { 點 }
35 </!big5>
36 <*/big5>
37 \zhnum_set_digits_map:nn { dot } { 點 }
38 </big5>
39 \zhnum_set_digits_map:nn { and } { 又 }
40 \zhnum_set_digits_map:nn { parts } { 分之 }
41 \zhnum_set_digits_map:nn { scale_zero } { }
42 <*/big5>
43 \zhnum_set_digits_map:nnn { scale_one } { simp } { 万 }
44 \zhnum_set_digits_map:nnn { scale_one } { trad } { 萬 }
45 \zhnum_set_digits_map:nnn { scale_two } { simp } { 亿 }
46 \zhnum_set_digits_map:nnn { scale_two } { trad } { 億 }
47 </!big5>
48 <*/big5>
49 \zhnum_set_digits_map:nn { scale_one } { 萬 }
50 \zhnum_set_digits_map:nn { scale_two } { 億 }
51 </big5>
52 \zhnum_set_digits_map:nn { scale_three } { 兆 }
53 \zhnum_set_digits_map:nn { scale_four } { 京 }
54 \zhnum_set_digits_map:nn { scale_five } { 垓 }
55 \zhnum_set_digits_map:nn { scale_six } { 秭 }
56 \zhnum_set_digits_map:nn { scale_seven } { 穰 }
57 <*/big5>
58 \zhnum_set_digits_map:nnn { scale_eight } { simp } { 沟 }
59 \zhnum_set_digits_map:nnn { scale_eight } { trad } { 溝 }
60 \zhnum_set_digits_map:nnn { scale_nine } { simp } { 涧 }
61 \zhnum_set_digits_map:nnn { scale_nine } { trad } { 澗 }
62 </!big5>
63 <*/big5>
64 \zhnum_set_digits_map:nn { scale_eight } { 澗 }
65 </big5>
66 \zhnum_set_digits_map:nn { scale_ten } { 正 }
67 <*/big5>
68 \zhnum_set_digits_map:nnn { scale_eleven } { simp } { 载 }
69 \zhnum_set_digits_map:nnn { scale_eleven } { trad } { 載 }
70 </!big5>
71 <*/big5>

```

```

72 \zhnum_set_digits_map:nn { scale_eleven } { 載 }
73 </big5>
74 \zhnum_set_digits_map:nn { year } { 年 }
75 \zhnum_set_digits_map:nn { month } { 月 }
76 \zhnum_set_digits_map:nn { day } { 日 }
77 <!*big5>
78 \zhnum_set_digits_map:nnn { hour } { simp } { 时 }
79 \zhnum_set_digits_map:nnn { hour } { trad } { 時 }
80 </!big5>
81 <!*big5>
82 \zhnum_set_digits_map:nn { hour } { 時 }
83 </big5>
84 \zhnum_set_digits_map:nn { minute } { 分 }
85 \zhnum_set_digits_map:nn { weekday } { 星期 }
86 \zhnum_set_financial_map:nn { null } { 零 }
87 \zhnum_set_financial_map:nn { zero } { 零 }
88 \zhnum_set_financial_map:nn { one } { 壹 }
89 \zhnum_set_financial_map:nn { two } { 貳 }
90 <!*big5>
91 \zhnum_set_financial_map:nnn { three } { simp } { 叁 }
92 \zhnum_set_financial_map:nnn { three } { trad } { 參 }
93 </!big5>
94 <!*big5>
95 \zhnum_set_financial_map:nn { three } { 參 }
96 </big5>
97 \zhnum_set_financial_map:nn { four } { 肆 }
98 \zhnum_set_financial_map:nn { five } { 伍 }
99 <!*big5>
100 \zhnum_set_financial_map:nnn { six } { simp } { 陆 }
101 \zhnum_set_financial_map:nnn { six } { trad } { 陸 }
102 </!big5>
103 <!*big5>
104 \zhnum_set_financial_map:nn { six } { 陸 }
105 </big5>
106 \zhnum_set_financial_map:nn { seven } { 柒 }
107 \zhnum_set_financial_map:nn { eight } { 捌 }
108 \zhnum_set_financial_map:nn { nine } { 玖 }
109 \zhnum_set_financial_map:nn { ten } { 拾 }
110 \zhnum_set_financial_map:nn { hundred } { 佰 }
111 \zhnum_set_financial_map:nn { thousand } { 仟 }
112 \ExplSyntaxOff
113 </config>

```


代码索引

斜体的数字表示对应项说明所在的页码,下划线的数字表示定义所在的代码行号,而直立的数字表示对应项使用时所在的行号。

Symbols	
\:	306
\;	306
\@ifpackagelater	9
\@nocounterr	62
\\	4, 5, 6, 546, 547
__int_get_sign:n	66
A	
\active	476
B	
\bool_if:NF	129, 163
\bool_if:nF	176
\bool_if:nT	485
\bool_if:NTF	222, 246, 425, 432, 445
\bool_if:nTF	168, 182, 221, 456, 563
\bool_set_false:N	477, 521, 526, 527, 535, 540
\bool_set_true:N	478, 522, 531, 532, 534, 539
C	
\c_eleven	333, 342
\c_false_bool	138, 142, 211
\c_five	288
\c_four	92, 103, 104, 187, 274, 287
\c_one	66, 83, 103, 113, 138, 142, 154, 178, 259, 272, 285
\c_one_hundred	275
\c_seven	278, 290
\c_six	275
\c_ten	132, 264, 272, 285, 386
\c_three	186, 258
\c_true_bool	100, 138, 142, 210
\c_twelve	259
\c_two	96, 169, 185
\c_zero	54, 66, 75, 77, 102, 106, 127, 132, 152, 162, 165, 166, 173, 174, 179, 194, 221
\c_zhnum_and_tl	41
\c_zhnum_day_tl	237, 243
\c_zhnum_dot_tl	49, 214
\c_zhnum_encoding_tl	509, 511, 542, 557
\c_zhnum_fri_tl	301
\c_zhnum_hour_tl	312, 319
\c_zhnum_hundred_tl	171
\c_zhnum_minus_tl	66, 217
\c_zhnum_minute_tl	313, 321
\c_zhnum_mon_tl	297
\c_zhnum_month_tl	236, 242
\c_zhnum_null_tl	222
\c_zhnum_parts_tl	37, 42
\c_zhnum_sat_tl	295
\c_zhnum_sun_tl	296
\c_zhnum_ten_tl	192
\c_zhnum_thousand_tl	164
\c_zhnum_thu_tl	300
\c_zhnum_tue_tl	298
\c_zhnum_wed_tl	299
\c_zhnum_weekday_tl	462, 463, 464, 465, 466, 467, 468
\c_zhnum_year_tl	235, 241
\c_zhnum_zero_tl	50, 129, 163, 166, 174, 222
\char_set_lccode:nn	306
\char_value_catcode:n	476
\CJK@makeActive	474, 478
\CJK@makeInactive	485, 486
\clist_map_inline:nn	338, 381, 392, 398, 469
\cs_generate_variant:Nn	46, 52, 55, 69, 86, 87, 109, 110, 146, 147, 156, 208, 209
\cs_if_exist:NT	474
\cs_if_exist_p:N	485
\cs_if_exist_use:cF	327
\cs_new:Nn	472
\cs_new_eq:NN	334
\cs_new_nopar:Nn	15, 16, 47, 53, 58, 64, 70, 72, 73, 88, 94, 111, 120, 125, 145, 148, 150, 158, 160, 202, 207, 210, 211, 212, 245, 248, 256, 262, 267, 280, 323, 325, 330, 348, 404, 411, 417, 439, 452, 460
\cs_new_nopar:Npn	21, 27, 33, 157, 233, 239, 247, 292, 304, 310, 316
\cs_new_protected_nopar:Nn	488, 490, 495, 497
\cs_set:Nn	346
\cs_to_str:c	71
\cs_to_str:N	87
D	
\DeclareExpandableDocumentCommand	13, 56, 196, 228
E	
\else:	84, 116, 363
\encoding	505
reset	3
\exp_after:wN	114, 115, 116
\exp_args:Nc	60
\exp_args:NnV	415
\exp_args:Nnx	559
\exp_args:NV	510
\exp_not:c	446, 447, 450, 470
\exp_not:N	462, 463, 464, 465, 466, 467, 468
\exp_not:n	425, 432, 433, 445, 456
\exp_not:V	426, 433, 457, 462, 463, 464, 465, 466, 467, 468
\exp_stop_f:	83, 113, 294, 350
\ExplSyntaxOff	112, 568
\ExplSyntaxOn	556
F	
\fi:	84, 117, 302, 370
\file_input:n	483
G	
\g_zhnum_cfg_map_finan_prop	414, 441, 482, 496, 499, 502, 504
\g_zhnum_cfg_map_prop	406, 407, 480, 489, 492, 502, 502
\g_zhnum_cfg_map_var_prop	419, 421, 481, 493, 500, 502, 503
\g_zhnum_scale_int	331, 332, 333, 343
\group_begin:	19, 205, 305
\group_end:	19, 205, 309
I	
\if_case:w	113, 294, 350

\if_int_compare:w	83	\msg_error:nxx	517
\IfBooleanT	231	\msg_new:nnn	2, 544
\IfBooleanTF	198		
\IfNoValueTF	18, 204	N	
\IfValueT	346	encoding	1
time	2	\NewDocumentCommand	335, 549
\int_case:nnn	364	\null	505
\int_compare:nNnF	135, 194, 386		
\int_compare:nNnT	66, 166, 174, 264	O	
\int_compare:nNnTF		\or:	114, 115, 296, 297, 298, 299, 300, 301, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362
... 77, 96, 102, 127, 132, 152, 162, 165, 173, 250, 258, 476			
\int_compare_p:nNn	169, 178, 179, 185, 186, 187, 221	P	
\int_div_truncate:nn	104, 272, 274, 275, 276, 285, 287, 319	\prg_new_conditional:Nnn	81
\int_eval:n	103, 113, 138, 142, 154, 246, 265, 271, 272, 273, 284, 285, 286, 294, 350	\prg_return_false:	84
\int_incr:N	340, 343	\prg_return_true:	84
\int_mod:nn	92, 132, 269, 282, 321, 331	\ProcessKeysOptions	555
\int_new:N	332	\prop_clear:N	480, 481, 482
\int_set_eq:NN	333	\prop_get:NnNT	414
\int_to_roman:n	369	\prop_get:NnNTF	419, 421, 441
\int_zero:N	337	\prop_map_function:NN	406, 407
		\prop_new:N	502, 503, 504
K		\prop_put:Nnn	489, 493, 496, 500
\keys_define:nn	372, 384, 388, 395, 403, 505	\prop_put_if_new:Nnn	492, 499
\keys_set:nn	551, 554, 559		
		Q	
L		\q_mark	31, 33
\l_keys_choice_tl	510	\q_nil	15, 24, 31
\l_keys_value_tl	517	\q_recursion_stop	149, 157
\l_tmpa_int	337, 340, 342	\q_stop	15, 21, 24, 27, 31, 33, 230, 233, 247, 292, 304, 310
\l_tmpa_tl	341, 343, 344, 414, 415, 419, 426, 433, 441	\quark_if_nil:nTF	23, 29, 35
\l_tmpb_tl	421, 426		
\l_zhnum_ancient_bool	169, 184, 521, 526, 531	R	
\l_zhnum_day_tl	468	\RequirePackage	11, 12
\l_zhnum_financial_hundred_tl	378	\reset	505
\l_zhnum_financial_thousand_tl	379		
\l_zhnum_five_tl	466	S	
\l_zhnum_four_tl	465	\str_if_eq_x:nnTF	214, 216
\l_zhnum_fri_tl	466	\style	505
\l_zhnum_hundred_tl	376		
\l_zhnum_minus_tl	374	T	
\l_zhnum_mon_tl	462	\tex_day:D	243
\l_zhnum_normal_bool	445, 522, 527, 532	\tex_ignorespaces:D	552
\l_zhnum_null_bool	456, 537	\tex_month:D	242
\l_zhnum_null_tl	375, 457	\tex_time:D	319, 321
\l_zhnum_one_tl	462	\tex_year:D	241
\l_zhnum_sat_tl	467	\tl_count:n	91, 92
\l_zhnum_set_CJK_active_bool	477, 478, 485	\tl_expandable_lowercase:n	510
\l_zhnum_simp_bool	425, 432, 534, 535	\tl_gset:Nx	509
\l_zhnum_six_tl	467	\tl_head:n	77
\l_zhnum_sun_tl	468	\tl_if_blank:fF	41
\l_zhnum_thousand_tl	377	\tl_if_blank:fTF	50
\l_zhnum_three_tl	464	\tl_if_blank:nF	46
\l_zhnum_thu_tl	465	\tl_if_blank:nTF	52, 54
\l_zhnum_time_bool	246, 539, 540	\tl_if_empty:nTF	75
\l_zhnum_tue_tl	463	\tl_if_exist:cF	343
\l_zhnum_two_tl	463	\tl_if_exist:NF	557
\l_zhnum_wed_tl	464	\tl_map_function:fN	204, 205, 207
\l_zhnum_zero_tl	454, 457	\tl_map_function:nN	71, 209
\luatex_if_engine_p:	563	\tl_set:cn	344, 437
		\tl_set:cx	423, 430, 443, 450, 470
M		\tl_set:Nx	341, 454, 462, 463, 464, 465, 466, 467, 468
\msg_error:nn	10	\tl_to_lowercase:n	307
		\tl_use:c	324

\token_if_int_register:NTF	60	\zhnum_number:n	14, 15
\TrimSpaces	335	\zhnum_number:w	15, 21, 21
style	2	\zhnum_number_aux:nNn	14, 16, 16, 57
U			
null	3	\zhnum_number_item:nn	122, 148, 148
\use_i_ii:nnn	127, 133	\zhnum_number_item_aux:fNNNN	154
\use_none:n	78, 114, 189	\zhnum_number_item_aux:nNNNN	148, 149, 150, 156
\use_none:nn	115	\zhnum_parse_config:	404, 404, 484
\use_none:nnn	116	\zhnum_parse_number:f	67
X			
\xetex_if_engine_p:	563	\zhnum_parse_number:n	88, 88, 109
Z			
\zhcurrtime	2, 316, 316	\zhnum_parse_number:nff	90
\zhdate	2, 228, 228	\zhnum_parse_number:nn	88
\zhdigits	1, 3, 50, 196, 196	\zhnum_parse_number:nnn	94, 110
\zhnum	1, 3, 56, 56	\zhnum_process_number:NNn	130, 158, 158
\zhnum_blank_to_zero:f	37, 38, 42, 43, 49	\zhnum_process_number:NNNNN	158, 159, 160
\zhnum_blank_to_zero:n	53, 53, 55	\zhnum_recursion_stop:NNNw	153, 157
\zhnum_check_financial:nn	404, 407, 439	\zhnum_scale_map:n	131, 325, 325, 331
\zhnum_check_simp:nn	404, 406, 411	\zhnum_scale_map_hook:n	328, 334, 346
\zhnum_check_simp_aux:nn	413, 415, 417	\zhnum_scale_map_loop:n	325, 330, 334
\zhnum_check_time:Nn	235, 236, 237, 241, 242, 243, 245, 245, 312, 313, 318, 320	\zhnum_set_digits_map:nn	7, 9, 11, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 37, 39, 40, 41, 49, 50, 52, 53, 54, 55, 56, 64, 66, 72, 74, 75, 76, 82, 84, 85, 488, 488
\zhnum_counter:n	57, 58	\zhnum_set_digits_map:nnn	3, 4, 33, 34, 43, 44, 45, 46, 58, 59, 60, 61, 68, 69, 78, 79, 488, 490
\zhnum_date:w	230, 233	\zhnum_set_financial_map:nn	86, 87, 88, 89, 95, 97, 98, 104, 106, 107, 108, 109, 110, 111, 488, 495
\zhnum_decimal:nn	25, 47, 47	\zhnum_set_financial_map:nnn	91, 92, 100, 101, 488, 497
\zhnum_digit_map:n	54, 97, 164, 170, 171, 189, 190, 194, 223, 323, 323	\zhnum_set_week_day:	404, 409, 460
\zhnum_digits_aux:NN	210, 210, 211, 212	\zhnum_set_zero:	404, 408, 452
\zhnum_digits_aux:nnN	199, 200, 202	\zhnum_split_number:fNNfn	99
\zhnum_digits_null:n	207, 208, 235	\zhnum_split_number:nNNnf	137, 141
\zhnum_digits_null:V	241	\zhnum_split_number:nNNnn	120, 120, 146
\zhnum_digits_null_aux:N	200, 207, 210, 210	\zhnum_split_number_aux:fnNNnn	122
\zhnum_digits_zero_aux:N	199, 210, 211	\zhnum_split_number_aux:nnNNnn	125, 147
\zhnum_erase_separator:n	67, 70, 70	\zhnum_time:w	304, 310
\zhnum_erase_separator_aux:N	71, 72	\zhnum_trim_zeros:f	67
\zhnum_fraction:w	31, 33, 33	\zhnum_trim_zeros:n	70, 73, 86
\zhnum_if_number:N	81	\zhnum_trim_zeros:o	78
\zhnum_if_number:NT	72, 219	\zhnum_two_digits:n	248, 251, 262
\zhnum_insert_zeros:n	99, 111, 111	\zhnum_use_i_iii:nnn	133, 145
\zhnum_int_to_word:n	324, 327, 342, 348, 348, 385, 389, 396	\zhnum_week_day:w	247, 292, 292
\zhnum_integer:f	30, 236, 237, 312, 313, 318, 320	\zhnum_Zeller:nnn	248, 248, 294
\zhnum_integer:n	64, 64, 69	\zhnum_Zeller_aux:Nnnn	248, 252, 253, 256
\zhnum_integer:V	242, 243	\zhnum_Zeller_Gregorian:nnn	252, 267, 267
\zhnum_integer:v	61	\zhnum_Zeller_Julian:nnn	253, 280, 280
\zhnum_integer_or_fraction:w	24, 27, 27	\zhnumber	1, 3, 13, 13, 41, 54
\zhnum_load_cfg:	472	\zhnumExtendScaleMap	2, 335, 335
\zhnum_load_cfg:n	472, 511, 542	\zhnumsetup	2, 19, 205, 549, 549
		\zhtime	2, 304, 304
		\zhtoday	2, 239, 239
		\zhweekday	2, 231, 247, 247