1 Introduction

The aim of the numspell package is to spell the cardinal and ordinal numbers from 0 to $10^{67} - 1$ (i.e. maximum 66 digits).

Currently, the supported languages are English, French, German, Hungarian and Italian. The spelling will happen in the current language.

The numspell package requires the services of the following packages: xstring, etoolbox, pdftexcmds.

Load the package as usual, with

\usepackage{numspell}

2 Commands

\numspell[(zeros)]{num}

Spelling the cardinal number $n = \langle num \rangle \cdot 10^{\langle zeros \rangle}$, where $0 \leq n \leq 10^{67} - 1$. The default value of $\langle zeros \rangle$ is 0. For example

\numspell{12000} → twelve thousand
\numspell[3]{12} → twelve thousand
\numspell[6]{12} → twelve million
\numspell[63]{1} → one vigintillion

\thenumspell

The \numspell stores the result in this command. For example

\numspell{12000}; \thenumspell → twelve thousand; twelve thousand
\numspell{1}; \numspell{2}; \thenumspell → one; two; two

\numspellsave{name}

It generates the \thenumspell{name} command, which saves the current \thenumspell. For example

\numspell{1};
\numspellsave{MyNum}
\numspell{2};
\thenumspell;
\thenumspellMyNum

one; two; two; one

\numspeldashspace{length}

In the number spelling, the spaces around the dashes are flexibility for the optimal hyphenation. Its value is 0pt plus \langle length \rangle. The default value of \langle length \rangle is 2pt. For example
It works like \numspell, but the number spelling will not be printed. In other words, the following two lines are equivalent:

\numspell*[\langle zeros \rangle\{\langle num \rangle\}]
\thenumspell

For example
\numspell*[1]
\numspellsave{MyNum}
\numspell*[2]
\thenumspell;\thenumspell MyNum

two; one

\Numspell*[\langle zeros \rangle\{\langle num \rangle\}]

It works like \Numspell, but the first letter will be capital. For example

\Numspell{12000} \rightarrow Twelve thousand
\Numspell[3]{12} \rightarrow Twelve thousand
\Numspell[6]{12} \rightarrow Twelve million
\Numspell[63]{1} \rightarrow One vigintillion

\Numspell*[\langle zeros \rangle\{\langle num \rangle\}]

It works like \Numspell, but the number spelling will not be printed. In other words, the following two lines are equivalent:

\Numspell*[\langle zeros \rangle\{\langle num \rangle\}]
\thenumspell

For example
\Numspell*[1]
\numspellsave{MyNum}
\Numspell*[2]
\thenumspell;\thenumspell MyNum

Two; One

\ordnumspell*[\langle zeros \rangle\{\langle num \rangle\}]

Spelling the ordinal number \( n = \langle num \rangle \cdot 10^{\langle zeros \rangle} \), where \( 0 \leq n \leq 10^{67} - 1 \). The default value of \langle zeros \rangle is 0. For example
It works like \ordnumspell, but the number spelling will not be printed. In other words, the following two lines are equivalent:
\ordnumspell\{\{zeros\}\}(\{num\})
\ordnumspell\{\{zeros\}\}(\{num\})\thenumspell

For example
\ordnumspell\{1\}
\numspellsave\{MyNum\}
\ordnumspell\{2\}
\thenumspell\;\thenumspell\{MyNum\}

second; first

3 Commands for English language

\numspellUS
By default, the number spelling will happen in British English, if the english language is active. This command changes it to American English. For example
\numspellUS\numspell\{1012345\} → one million, twelve thousand, three hundred forty-five

\numspellGB
Using the \numspellUS command, you can rechange it to British English by this command. For example
\numspellUS\numspell\{1012345\}\numspellGB\numspell\{1012345\}
one million, twelve thousand, three hundred forty-five
one million, twelve thousand and three hundred and forty-five
4 Commands for French language

The following commands only work, if \texttt{french} language is active.

\textbf{\texttt{\textbackslash numspellpremiere}}

By default, \texttt{\textbackslash ordnums{1}} → premier,
but \texttt{\textbackslash numspellpremiere\textbackslash ordnums{1}} → première

\textbf{\texttt{\textbackslash numspellpremier}} (default)
\texttt{\textbackslash numspellpremiere\textbackslash ordnums{1};}
\texttt{\textbackslash numspellpremier\textbackslash ordnums{1}}
première; premier

5 Commands for Hungarian language

The following commands only work, if \texttt{magyar} language is active.

\textbf{\texttt{\textbackslash anumspell\{zeros\}\{num\}}}

It works like \texttt{\textbackslash numspell}, but the number spelling will start with Hungarian definite article. For example

\begin{itemize}
\item \texttt{\textbackslash numspell{1}} → az egy
\item \texttt{\textbackslash numspell{2}} → a kettő
\end{itemize}

\textbf{\texttt{\textbackslash anumspell*[zeros]\{num\}}}

It works like \texttt{\textbackslash anumspell}, but the number spelling will not be printed. In other words, the following two lines are equivalent:

\begin{itemize}
\item \texttt{\textbackslash anumspell\{zeros\}\{num\}}
\item \texttt{\textbackslash anumspell*[zeros]\{num\}\thenumspell}
\end{itemize}

For example

\begin{itemize}
\item \texttt{\textbackslash anumspell*[1} \numspellsave{MyNum} \textbackslash anumspell*[2} \thenumspell \thenumspell{MyNum}
\end{itemize}

a kettő; az egy

\textbf{\texttt{\textbackslash Anumspell\{zeros\}\{num\}}}

It works like \texttt{\textbackslash anumspell}, but the first letter will be capital. For example

\begin{itemize}
\item \texttt{\textbackslash Anumspell{1}} → Az egy
\item \texttt{\textbackslash Anumspell{2}} → A kettő
\end{itemize}

\textbf{\texttt{\textbackslash Anumspell*[zeros]\{num\}}}

It works like \texttt{\textbackslash Anumspell}, but the number spelling will not be printed. In other words, the following two lines are equivalent:

\begin{itemize}
\item \texttt{\textbackslash Anumspell\{zeros\}\{num\}}
\item \texttt{\textbackslash Anumspell*[zeros]\{num\}\thenumspell}
\end{itemize}

For example
\texttt{\numspell*{1}}
\texttt{\numspellsave{MyNum}}
\texttt{\numspell*{2}}
\texttt{\thenumspell;}
\texttt{\thenumspell{MyNum}}

A kettő; Az egy

\texttt{\aordnumspell[(zeros)]{(num)}}

It works like \texttt{\ordnumspell}, but the number spelling will start with Hungarian definite article. For example

\texttt{\aordnumspell{1}} \rightarrow \texttt{az első}
\texttt{\aordnumspell{2}} \rightarrow \texttt{a második}

\texttt{\aordnumspell*[(zeros)]{(num)}}

It works like \texttt{\aordnumspell}, but the number spelling will not be printed. In other words, the following two lines are equivalent:

\texttt{\aordnumspell[(zeros)]{(num)}}
\texttt{\aordnumspell*[(zeros)]{(num)}} \texttt{\thenumspell}

For example

\texttt{\aordnumspell*{1}}
\texttt{\numspellsave{MyNum}}
\texttt{\aordnumspell*{2}}
\texttt{\thenumspell;}
\texttt{\thenumspell{MyNum}}

a második; az első

\texttt{\Aordnumspell[(zeros)]{(num)}}

It works like \texttt{\aordnumspell}, but the first letter will be capital. For example

\texttt{\Aordnumspell{1}} \rightarrow \texttt{Az első}
\texttt{\Aordnumspell{2}} \rightarrow \texttt{A második}

\texttt{\Aordnumspell*[(zeros)]{(num)}}

It works like \texttt{\Aordnumspell}, but the number spelling will not be printed. In other words, the following two lines are equivalent:

\texttt{\Aordnumspell[(zeros)]{(num)}}
\texttt{\Aordnumspell*[(zeros)]{(num)}} \texttt{\thenumspell}

For example

\texttt{\Aordnumspell*{1}}
\texttt{\numspellsave{MyNum}}
\texttt{\Aordnumspell*{2}}
\texttt{\thenumspell;}
\texttt{\thenumspell{MyNum}}

A második; Az első
6 Commands for Italian language

The following commands only work, if Italian language is active.

\numspellitmasculine (default)
The ordinal numbers will be printed in masculine form. For example
\ordnumspell{1} \rightarrow primo

\numspellitfeminine
The ordinal numbers will be printed in feminine form. For example
\numspellitfeminine\ordnumspell{1};
\numspellitmasculine\ordnumspell{1}
prima; primo

7 Examples

Example 1

\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ magyar, italian, ngerman, french, english ]{babel}
\usepackage{numspell}
\usepackage[group-separator={,}]{siunitx}
\begin{document}
\def\mynum{123456789}
\noindent
In American English the spelling of \num{\mynum} is
\numspellUS\emph{\numspell{\mynum}''.}
\smallskip
\noindent
In British English the spelling of \num{\mynum} is
\emph{\numspell{\mynum}''.}
\smallskip
\noindent
In French the spelling of \num{\mynum} is
\selectlanguage{french}\emph{\numspell{\mynum}''.}
\smallskip
\noindent
In German the spelling of \num{\mynum} is
\selectlanguage{ngerman}\emph{\numspell{\mynum}''.}
\smallskip
\noindent
In Hungarian the spelling of \num{\mynum} is
\selectlanguage{magyar}\emph{\numspell{\mynum}''.}
\smallskip
\noindent
In Italian the spelling of \num{\mynum} is
\selectlanguage{italian}\emph{\numspell{\mynum}''.}
\end{document}

In American English the spelling of 123,456,789 is “one hundred twenty-three million, four hundred fifty-six thousand, seven hundred eighty-nine”.
In British English the spelling of 123,456,789 is “one hundred and twenty-three million, four hundred and fifty-six thousand and seven hundred and eighty-nine”.

In French the spelling of 123,456,789 is “cent vingt-trois millions quatre cent cinquante-six mille sept cent quatre-vingt-neuf”.

In German the spelling of 123,456,789 is “einhundertdreizehn Millionen vierhundertsechsundfünfzigtausendzweihundertachtzehn”.

In Hungarian the spelling of 123,456,789 is “százhuszonnégymillió-négyszázötvenhátezer-hétszáznégyvenkilenc”.

In Italian the spelling of 123,456,789 is “centoventitré milioni quattrocentocinquantaesimila settecentotantanove”.

Example 2

\documentclass{article}
\usepackage{numspell}
\usepackage[group-separator=,]{siunitx}
\begin{document}
\def\mynum{123456789012345678901234567890123456789012345678901234567890}
\Numspell{\mynum}, that is \num{\mynum}.
\end{document}

One hundred and twenty-three vigintillion, four hundred and fifty-six novemdecillion, seven hundred and eighty-nine octodecillion, twelve septendecillion, three hundred and forty-five sexdecillion, six hundred and seventy-eight quindecillion, nine hundred and one quattuordecillion, two hundred and thirty-four tredecillion, five hundred and sixty-seven duodecillion, eight hundred and ninety undecillion, one hundred and twenty-three decillion, four hundred and fifty-six nonillion, seven hundred and eighty-nine octillion, twelve septillion, three hundred and forty-five sextillion, six hundred and seventy-eight quintillion, nine hundred and one quadrillion, two hundred and thirty-four trillion, five hundred and sixty-seven billion, eight hundred and ninety million, one hundred and twenty-three thousand and four hundred and fifty-six, that is 123,456,789,012,345,678,901,234,567,890,123,456,789,012,345,678,901,234,567,890,123,456.

Example 3

\documentclass{article}
\usepackage{numspell}
\newcounter{mycount}
\makeatletter
\begin{document}
The \@whilenum\value{mycount}<51 \do{\ordnumspell{\themycount}\stepcounter{mycount}, \ }\dots
\end{document}

Example 4

\documentclass{article}
\usepackage{numspell}
\newcounter{mycount}
\def\themycount{\numspell{\arabic{mycount}}}
\makeatletter
\begin{document}
\Numspell{0},
@whilenum\value{mycount}<30
\do{\stepcounter{mycount}\themycount, }\dots
\end{document}

Nought, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, twenty-one, twenty-two, twenty-three, twenty-four, twenty-five, twenty-six, twenty-seven, twenty-eight, twenty-nine, thirty, …

8 Limitations

Do not use the \numspell, \numspell*, \Numspell, \Numspell*, etc. commands inside \MakeUppercase and sectioning commands. An example for the illustration of the problem:

\documentclass{article}
\usepackage{hyperref,numspell}
\pagestyle{headings}
\begin{document}
\section{The \ordnumspell{123} factor}
\MakeUppercase{\numspell{123}}
\newpage
Text
\end{document}

The bugs:

1. You can see it on the page 1: “one hundred and twenty-three”
   Required: “ONE HUNDRED AND TWENTY-THREE”

2. You can see it on the heading: “THE one hundred and twenty-third FACTOR”
   Required: “THE ONE HUNDRED AND TWENTY-THIRD FACTOR”

3. You can see it on the pdf bookmark: “The 123 factor”
   Required: “The one hundred and twenty-third factor”

The solution is very easy:

\ordnumspell*{123}
\section{The \thenumspell\ factor}
\numspell*{123}
\MakeUppercase{\thenumspell}