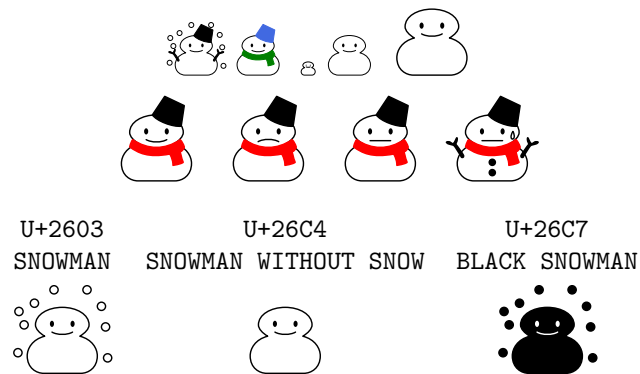


The `scsnowman` package v1.2c

Hironobu Yamashita

2018/01/15

The \LaTeX package `scsnowman` provides a command `\scsnowman`, which can display many variants of snowmen. This package utilizes `TikZ` for drawing snowmen.



The package is maintained on GitHub:

- <https://github.com/aminophen/scsnowman>

Contents

1	The History of Snowman in Unicode	3
2	Variation of Snowman among Actual Fonts	3
3	Introduction to <code>scsnowman</code> Package	5
4	Command Options	5
4.1	Scaling and Adjustment Options	5
4.2	Design Options	5
5	Changing the Default	6
6	Adding User-defined Snowman Shapes	6

7	Funny Usages	7
7.1	Changing Item Labels and QED Symbols	7
7.2	Drawing “ <i>Kagami-mochi</i> ”	7

1 The History of Snowman in Unicode

In October 1991, the first volume of the Unicode standard was published. Since then, there was a code point assigned to a character “snowman”; U+2603.

U+2603 SNOWMAN
= snowy weather

It seems that the shape of its reference glyph in Unicode 1.0.0 was taken from “Ryumin”, which was developed by Morisawa (a famous font vendor in Japan). A few years later, the reference glyph has sometimes been changed to another; however, there had been only one “snowman” in Unicode until 2009.

In October 2009, Unicode 5.2 was published. In this volume, two “snowman” code points were added; U+26C4 and U+26C7.

U+26C4 SNOWMAN WITHOUT SNOW
= light snow
U+26C7 BLACK SNOWMAN
= heavy snow

According to the code chart, the origin of these two characters is ARIB STD-B24 (Data Coding and Transmission Specification for Digital Broadcasting;¹), which was established by Association of Radio Industries and Business in Japan. Since then, it can be said that the old code point U+2603 has been given an implicit meaning of “SNOWMAN WITH SNOW”. The reference glyphs were also changed at that time.

2 Variation of Snowman among Actual Fonts

Since the shapes of the reference glyphs used in the Unicode code charts are not prescriptive, the actual fonts have a wide variety of glyph designs. However, when it comes to snowmen, the variation between fonts is enormous. This variation is very interesting, however, on the other hand, problematic.

Table 1 shows the variety of “snowman” in actual fonts. The snowman in “IPA Mincho (IPA 明朝)” from Information-technology Promotion Agency is very similar to the one in “Ryumin (リユウミン)” from Morisawa. However, in “MS Mincho (MS 明朝)” from Microsoft, the snowman wears a black hat instead of white one. In “Kozuka Mincho (小塚明朝)” from Adobe Systems Inc., he/she wears a muffler instead of a hat. Moreover, it doesn’t snow in “Hiragino Mincho (ヒラギノ明朝)” from SCREEN Graphic and Precision Solutions Co., Ltd. It is natural that some fonts developed before 2009 have a “snowman without snow” glyph in the code point U+2603, however, it can be a problem when we have to transfer the exact information to others.

¹http://www.arib.or.jp/tyosakenkyu/kikaku_hoso/hoso_std-b024.html; Abstract in PDF format (both Japanese and English) are available.

Table 1: The variety of “snowman” in actual fonts

	U+2603	U+26C4	U+26C7
IPAex 明朝			
MS 明朝			
小塚明朝 Pr6N Regular			
ヒラギノ明朝 ProN W3			
VL ゴシック			

3 Introduction to scsnowman Package

The L^AT_EX package `scsnowman` provides a command `\scsnowman`, which can display many variants of snowmen. This package depends on TikZ package for drawing snowman images.

To use this package, load it in preamble:

```
\usepackage{scsnowman}
```

In the main document, use `\scsnowman` command to print a snowman: ☺. By default, the snowman is “plain” style, without any decoration such as snow, a hat or a muffler.

4 Command Options

You can customize the style of a snowman using the optional argument. The syntax is

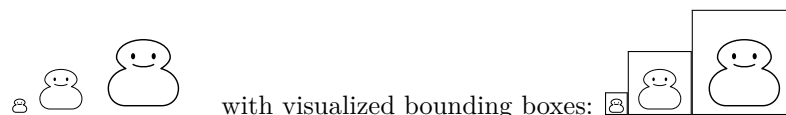
```
\scsnowman[⟨key-value list⟩]
```

4.1 Scaling and Adjustment Options

The following *keys* are available:

`scale`, `adjustbaseline`

The key `scale` takes a scale factor. The origin of scaling is set at the lower left corner of the bounding box. For example, `\scsnowman`, `\scsnowman[scale=3]` and `\scsnowman[scale=5]` give:



When the key `adjustbaseline` is specified (or, `adjustbaseline=true`), the base line of the in-line snowman will be adjusted to that of the surrounding texts. This will be helpful when a snowman appears to be “floating in the air.”

Text,	<code>\scsnowman[scale=1]%</code>	<code>\scsnowman[scale=4]%</code>	
	<code>\scsnowman[scale=7],</code>		
T.\par			
Text,	<code>\scsnowman[adjustbaseline,scale=1]%</code>	<code>\scsnowman[adjustbaseline,scale=4]%</code>	
	<code>\scsnowman[adjustbaseline,scale=7],</code>		
T.			

Text, ☺

, T.

Text, ☺

, T.

4.2 Design Options

Following *keys* take a *value* which specifies color. When the *value* is omitted, it reacts to the surrounding text color:

body, eyes, mouth, sweat, arms, hat, muffler, buttons, snow,
broom

The following *keys* require one specific *value*:


shape, mouthshape

The key **shape** specifies the outline shape of the snowman body. Currently, only **shape=normal** is officially available, but you can define and use other shapes using `\usescsnowmanlibrary` command (described later, section 6).

The key **mouthshape** takes one of the followings: **smile**, **tight** or **frown**.

Here are some examples:

```
\scsnowman[scale=2,body,hats=red,muffler=blue]
\scsnowman[scale=3,hats,snow,arms,buttons]
\scsnowman[scale=2,muffler=red,arms,broom=brown]
\scsnowman[scale=2,mouthshape=frown,hats=green]
{\color{blue}
\scsnowman[scale=2,body,hats=red,muffler=blue]
\scsnowman[scale=3,hats,snow,arms,buttons]
\scsnowman[scale=2,muffler=red,arms,broom=brown]
\scsnowman[scale=2,mouthshape=frown,hats=green]}
```



5 Changing the Default


The package default is the “plain” style snowman. This default can be changed by using `\scsnowmandefault` command. The syntax is

```
\scsnowmandefault{<key-value list>}
```

The available *keys* are the same as those in `\scsnowman`.

Here are some examples:

```
\scsnowmandefault{scale=5,hats=red}
\scsnowman
\scsnowman[body,muffler=blue]
\scsnowman[hats=green,snow]
```



6 Adding User-defined Snowman Shapes

Any users can define and use custom snowman shape definitions. Here is a description of adding a shape named `myfavorite`.

1. Prepare a snowman definition file `scsnowman-myfavorite.def` and put it into \TeX MF tree (e.g. `texmf-local/tex/latex/scsnowman/`). For the format of snowman definition files, please refer to `scsnowman-normal.def`.
2. Use `\usescsnowmanlibrary` command (don’t forget *sc!*) to load it.

```
\usepackage{scsnowman}
\usescsnowmanlibrary{myfavorite}
```

3. You can use the shape by `\scsnowman[shape=myfavorite]` command.

If you have created a fancy snowman, please contact me, so that I can incorporate it into the official release!

7 Funny Usages

7.1 Changing Item Labels and QED Symbols

For those who want more snowmen in the documents, currently `scsnowman` provides the following additional commands:

```
\makeitemsnowman:
Change item labels in itemize environment to snowmen 🍷🍷🍷.
The command \makeitemother restores the default, usually •-*.

\makeqedsnowman:
Change the QED symbol in proof environment to a snowman 🍷.
The package amsthm is required. The command \makeqedother
restores the default, usually □.
```

These commands can be used wherever you want, and are effective within the current group.

Here are some examples:

<pre>\begin{itemize} \makeitemsnowman \item Foo X. \begin{itemize} \item Bar A. \begin{itemize} \item Baz P. \item Baz Q. \end{itemize} \item Bar B. \item Bar C. \end{itemize} \end{itemize} \makeqedsnowman \begin{theorem} Given two line segments whose lengths are a and b respectively, there is a real number r such that $b=ra$. \end{theorem} \begin{proof} To prove it by contradiction try and assume that the statement is false, proceed from there and at some point you will arrive to a contradiction. \end{proof}</pre>	<pre>🍷 Foo X. 🍷 Bar A. 🍷 Baz P. 🍷 Baz Q. 🍷 Bar B. 🍷 Bar C.</pre> <p>Theorem 1. <i>Given two line segments whose lengths are a and b respectively, there is a real number r such that $b = ra$.</i></p> <p><i>Proof.</i> To prove it by contradiction try and assume that the statement is false, proceed from there and at some point you will arrive to a contradiction. 🍷</p>
--	--

The names of these commands are, of course, named after the \LaTeX `\makeatletter` and `\makeatother`;-)

7.2 Drawing “*Kagami-mochi*”

Using `scsnowman` package, you can also draw “*kagami-mochi*” (mirror rice cake). It is a traditional Japanese New Year decoration, which usually consists of two round “*mochi*” (rice cakes), the smaller placed atop the larger, and a “*daidai*” (a Japanese bitter orange) with an attached leaf on top.

Following *keys* are implemented for this usage:²

`mikan`, `leaf`

The *key* `leaf` is effective only when `mikan` is specified. Here is an example:

```
\scsnowmandefault{scale=5.5}  
\scsnowman[eyes=false,mouth=false,mikan=orange,leaf=green]
```



Version History

This is the summary of changes. For more detail, see GitHub repository.

Version 0.1	2015-12-13	First public version on GitHub
Version 0.8	2016-08-08	Second public version on GitHub: new variants <code>buttons</code> , <code>mouthshape</code> , <code>sweat</code> are added
Version 1.0	2016-12-22	First CTAN release
Version 1.1	2017-01-22	Add a new key <code>adjustbaseline</code> Update documentation
Version 1.2	2017-08-08	Default color reacts to surrounding text color Support vertical writing on (u)pTeX and LuaTeX-j Add a new key <code>shape</code> Add a new command <code>\usescsnowmanlibrary</code>
Version 1.2a	2017-11-25	Fix a bug in scaling with <code>adjustbaseline</code>
Version 1.2b	2018-01-05	Add new keys <code>mikan</code> , <code>leaf</code> for drawing ‘kagami-mochi’ Stopped loading <code>amsthm</code> by default
Version 1.2c	2018-01-15	Add a new key <code>broom</code>

References

- [1] 雪だるまの親子関係 — Mac OS X の文字コード問題に関するメモ
- [2] ヒラギノの雪だるまは、なぜ寂しそうなのか — Mac OS X の文字コード問題に関するメモ
- [3] いろいろなゆきだるま — TeX Alchemist Online
- [4] 「TeX でゆきだるま」をもっとたくさん — Acetaminophen’s diary
- [5] Unicode の例の雪だるまは多分アレ — マクロツイーター
- [6] TeX でゆきだるまを “もっともっと” たくさん — Acetaminophen’s diary
- [7] 夏といえば、やっぱり「ゆきだるま」！ — Acetaminophen’s diary
- [8] How do I redefine the QED symbol to be a Unicode character? — TeX – LaTeX Stack Exchange

²Strictly speaking, the orange on top of rice cakes should be “*daidai*”; however, a “*mikan*” is often substituted for the original “*daidai*”.