Ponysay

by Mattias Andrée (maandree), et al.

This manual is for ponysay (version 3.0.3).

Copyright © 2012-2016 Mattias Andrée, et al.

'For me! For my friends! **For EQUESTRIA!**'

Table of Contents

1 Overview

`ponysay` displays an image of a My Little Pony pony saying a message provided by the user in a terminal, or a quote from the show My Little Pony: Friendship is Magic (MLP:FiM). Historically `ponysay` was a wrapper for cowsay, but has since version 2.1 become an independent reimplementation of `cowsay`.

If a message is not provided, e.g. by piping, it accepts standard input. The pony quoting the given message is printed on standard output.

`ponythink` is to `ponysay` as `cowthink` is to `cowsay`.

`ponysay` is generally used to decorate your terminal with a random pony from a shell start-up script. But if you know anypony else who likes ponies you can always make screen-shots of `ponysay --q` executions and communicate that way over e-mail.

2 Invoking `ponysay`

The format for running the `ponysay` program is:

    ponysay [*option*...] [--] [*message*]
    ponythink [*option*...] [--] [*message*]

Running `ponysay` will print a speech balloon, `ponythink` will print a thought balloon. Otherwise `ponysay` and `ponythink` is the same thing.

`ponysay` supports the following options:

`--`   Parse the following arguments as parts of *message*.

`-h`
`--help`   Show summary of options.

`+h`
`++help`   Show summary of options, and use colours even if stdout is piped.

`-v`
`--version`
             Show version of program.

`-f PONY`
`--file PONY`
`--pony PONY`
             Specify the pony that should printed, this can either be a file name or a pony name printed by `ponysay -l`. This option can be used multiple times to specify a set of ponies from which one will be selected randomly. If no pony is specified one will be selected randomly.

             If you have `util-say` installed, you can use .png-files as the arguments for this options.

             In versions earlier than version 2.0, the if the pony were a file name it had to include a '/'. This is not longer required and any existing pony name supersedes file names.

`+f PONY`
`++file PONY`
`++pony PONY`
             Just as `+f`, but it uses extra (non-MLP:FiM) ponies instead of standard (MLP:FiM) ponies

`-F PONY`
`--any-file PONY`
`--anyfile PONY`
`--any-pony PONY`
`--anypony PONY`
             This option combines `-f` and `+f`.

`-q PONY`
`--quote PONY`
             By using this option, a pony will be printed with quotes from her in My Little Pony: Friendship is Magic. The pony will be selected randomly, unless at least one pony is added as an argument to `-q`. If one or more ponies are added as an argument to `-q`, the pony will be selected randomly from that set of ponies. This option requires the extension `ponyquotes4ponysay`, which is included by default since version 1.2.

             The argument can be a file name, but only if it ends with `.pony`.

```
--f [PONY...]
--files [PONY...]
--ponies [PONY...]
```
        Variadic variant of `-f`, meaning that all arguments added after this one will parsed as an argument to this option. Additionally, those options are added to `-f`.

```
++f [PONY...]
++files [PONY...]
++ponies [PONY...]
```
        Variadic variant of `+f`, meaning that all arguments added after this one will parsed as an argument to this option. Additionally, those options are added to `+f`.

        An important feature of this options, is that you can but it in the end of the command line, without any argument to get a random non-MLP:FiM pony. However, altough it is not nice, since version 3.0, `+f` can also be unargumented if at the end of the command line.

```
--F [PONY...]
--any-files [PONY...]
--anyfiles [PONY...]
--any-ponies [PONY...]
--anyponies [PONY...]
```
        This option combines `--f` and `++f`.

```
--q [PONY...]
--quotes [PONY...]
```
        Variadic variant of `-q`, meaning that all arguments added after this one will parsed as an argument to this option. Additionally, those options are added to `-q`.

        An important feature of this options, is that you can but it in the end of the command line, without any argument to get a quote from any pony with a quote. However, altough it is not nice, since version 3.0, `-q` can also be unargumented if at the end of the command line.

```
-b STYLE
--bubble STYLE
--balloon STYLE
```
        Specify the balloon style that should used, this can either be a file name or a balloon name printed by `ponysay -B`. This option can be used multiple times to specify a set of styles from which one will be selected randomly. If no balloon style is specified a fallback style will be used.

```
-W COLUMN
--wrap COLUMN
```
        Specify the screen column where the message should be wrapped, this is by default 65, as with `cowsay`. The balloon's extra width is taken into consideration.

        If the argument is not a number, but starts instead with `n` (for 'none' or 'no'), no wrapping is done, and if it starts with `i` (for 'inherit') the width of the terminal is used.

        `n` and `i` is case insensitive, so you may use `N` and `I` instead. Additionally, typo correction is for QWERTY (and QWERTZ) and Dvorak is built in to `ponysay`; the nearest key, either to the left or to the right, depending on which hand is used to press the key, is also allowed.

`-c`
`--compress`
`--compact`
>   Compress the message in the same way `cowsay` does, that is basically without multiple spaces, and only paragraphs separations. Using this options will mean that you cannot display `figlet` and `TOIlet` style messages.

`-l`
`--list`   Lists all installed ponies. The ponies which have quotes, i.e. can be used with the `-q` option, will be marked by being printed in bold or bright (depending on the terminal.)

`-L`
`--altlist`
`--symlist`
>   Lists all installed ponies. The ponies which have quotes, i.e. can be used with the `-q` option, will be marked by being printed in bold or bright (depending on the terminal.) This options differs from `-l` by printing alternative names (symbolic links) inside brackets after their target ponies.

`+l`
`++list`   Just as `-l`, except it lists extra (non-MLP:FiM) ponies instead of standard (MLP:FiM) ponies.

`+L`
`++symlist`
`++altlist`
>   Just as `-L`, except it lists extra (non-MLP:FiM) ponies instead of standard (MLP:FiM) ponies.

`-B`
`--bubblelist`
`--balloonlist`
>   Prints a list of all balloon styles.

`-A`
`--all`   List all ponies, MLP:FiM and non-MLP:FiM, in this case the first list are MLP:FiM and the second are non-MLP:FiM.

`+A`
`++all`
`--symall`
`--altall`   List all ponies names, including alternatives, these from MLP:FiM and non-MLP:FiM. The first list are the MLP:FiM and the second one are non-MLP:FiM.

`-o`
`--pony-only`
`--ponyonly`
>   Print just the pony, nothing else like the speech balloon. Naturally the `ponysay` will not wait for a message from stdin.

`-i`
`--info`   By adding this flag you will get a metadata for a pony printed, rather than the pony itself. The output will beformated with bold tag names. The output will be wrapped according to the `-W` option.

`+i`

`++info`   This works just like the `-i` option, except the pony will use the output has her
message rather that just print that information.

`-r RESTRICTION`

`--restrict RESTRICTION`

This option is used to restrict which ponies can be randomly select based one their
metadata. The restrict is given is disjunctive normal form, and can hence express
any logical combination, however only for tags with one entry. For tags with multiple
values all values are tested and of one of them passes a test passes.

The argument for `--restrict` is a `+` separated list of values that all must be satisfied
for a pony to be qualified for random selection. The option `--restrict` can be used
multiply times, only one of them need to be satisfied for a pony to qualified for
random selection.

A value in the argument is a combination of the tag name and tag value on the form
`NAME=VALUE`. Additionally if the tag names ends with a question mark (`?`) the tag is
satsified if the tag is missing; if the value starts with a bang (`!`) the test is inverted.
Using just a bang means that the test passes for and only for all ponies with the
tag definied; using the question mark and a empty value means that the test passes
for all ponies; finally, using the question mark and just a bang for the value means
that the test passes for and only for all ponies without the tag definied.

For   most   shells,   if   not   all,   trick   to   not   need   to   use   disjunc-
tive   normal   form   is   to   use   `{ }`.   For   example   if   you   use
`--restrict={eye={blue,green,cyan}+coat={black,grey},coat=white}`
(note that there is no whitespaces) means that only ponies with white coat will be
randomly selected as will as ponies with black or grey coat provided that they have
either blue, green or cyan eyes.

`-X`

`--256-colours`

`--256colours`

`--x-colours`

Use `xterm`'s 256-colour support (supported by most X11 terminals), despite your
terminal's actual compatibilies.

`-V`

`--tty-colours`

`--ttycolours`

`--vt-colours`

Use Linux VT's compatbilies without KMS utilisation, despite your terminal's actual
compatibilies.

`-K`

`--kms-colours`

`--kmscolours`

Use Linux VT's compatbilies with KMS utilisation, despite your terminal's actual
compatibilies.

`+c`

`--colour ANSI-COLOUR`

Colour the balloon, including link and message (the parts that are not individually
specified.) The argument, should be a ANSI colour sequence without leading CSI
and without a tailing 'm', for example `1;31` will make it in red and bold (or bright
depending on the terminal.)

```
--colour-bubble
--colour-balloon ANSI-COLOUR
```
>   Just like `--colour`, but it only colours the balloon, without the message or link.

```
--colour-link ANSI-COLOUR
```
>   Just like `--colour`, but it only colours the balloon link.

```
--colour-msg
--colour-message ANSI-COLOUR
```
>   Just like `--colour`, but it only colours the message.

```
--colour-pony ANSI-COLOUR
```
>   Just like `--colour`, but it colours the pony. This colouring has no effect ony regular pony files, as it has its own colouring.

```
--colour-wrap
--colour-hyphen ANSI-COLOUR
```
>   Just like `--colour`, but it colours hyphen added by the word wrapping. By default this is red (31), if you want uncoloured use 0, without 0 or 39, the default 31 is presistent.

If neither `-q` is used nor any *message* is specified, `ponysay` will read the message from stdin (standard input); however, if no arguments are used and nothing is piped to stdin, a help message will be printed. If you want to use `ponysay` without arguments and enter the message by hand, you can run `cat | ponysay`.

If no pony is selected, `ponysay` will look for a `best.pony` file, this file should be a symbolic link to the pony you want as a default. If it is not a symbolic link, `-q` cannot determine which quotes to use.

3 Advanced usage of `ponysay`.

3.1 Extra information

If file descriptor 3 is definied when `ponysay` is executed, extra information is printed to it. The printed information includes the name of the pony file, the name of the balloon style file, and if definied in the pony file, file meta data and comment.

In most shells, a file descriptor 3 can defined using `3> FILE`, and linked to stderr using `3>&2`. For example, you can print the information to `~/info` by running `ponysay I\'m just the cutest pony! 3> ~/info`.

The message is not stored this way, for that you can use `tee`. However, if you use `-q` the quote file is printed to file descriptor 3.

3.2 Fortune cookies

If you have `fortune` installed – this program may be named `fortune-mod` in your GNU/Linux distributions package repository – you can run `fortune | ponysay` to get a random pony reading a random fortune cookie.

By adding `fortune | ponysay` to the end [easiest way] of your `~/.bashrc` – or equivalent for your shell if you do not use GNU Bash (standard shell for most distributions now a days) – you will get the effect described in the previous paragraph every time you open a terminal.

3.3 Ponification

You can ponify messages (i.e. replaces words search as 'everyone' with 'everypony') by using `fortune | ponypipe` instead of using `fortune`. `ponypipe` can be downloaded from `https://github.com/maandree/ponypipe`. Alternatively you can use `pinkie` (or `pinkiepie`), which can be downloaded from `https://github.com/maandree/pinkie-pie`, which is just `fortune | ponypipe`. There is also a large `sed` script, similar to `ponypipe`: `http://www.reddit.com/r/mylittlelinux/comments/srixi/using_ponysay_with_a_ponified_fortune_warning/` However I think `ponypipe` as better at replacing words than the `sed` script, but I haven't used the script so I wouldn't know for sure.

3.4 Running on TTY

If you use TTY and have a custom colour palette, you should also add to your `~/.bashrc`, before `fortune | ponysay`:

```
[ "$TERM" = "linux" ] &&
    function ponysay
    {   /usr/bin/ponysay "$@"
        #RESET PALETTE HERE
    }
```

You should read more about this in Section 5.1 [KMS ponies], page 12.

3.5 Running on `screen` and `tmux`

`screen` and `tmux` will adapt ANSI colour escape sequences to your terminal's capabilities. This means that if your terminal reports itself as `xterm` in `$TERM` ponies will lose their colours; they will only use the lower 16 colours instead of the top 240 colours. By default, almost all X terminals, including `xterm` and `mate-terminal` reports themselves as `xterm` in `$TERM`, and some

reports their actual name in `$COLORTERM`. So before opening `screen` or `tmux` you should set `$TERM` to `xterm-256color`, if you are using a terminal with support for `xterm`'s 256 colours; this can be done by adding to your `~/.bashrc`:

```
[ "$TERM" = "xterm" ] &&
    function screen
    {  TERM=xterm-256color /usr/bin/screen "$@"
    }


[ "$TERM" = "xterm" ] &&
    function tmux
    {  TERM=xterm-256color /usr/bin/tmux "$@"
    }
```

Alternatively, you can run `tmux` with the option `-2`: `tmux -2`. This also forces `tmux` to assume the terminal supports 256 colours.

3.6 `~/.ponysayrc`

If you have the file `~/.ponysayrc` (`.ponysayrc` in your home directory, the home directory can be spoofed by changing the system environment `HOME`,) the first thing `ponysay` does is running that file. This can be used for modifing environment variables (see Chapter 4 [Environment variables], page 10). For your convience this can be done by modifying the map `env`. The code in `~/.ponysayrc` must be written in Python 3.

For example if you want to set the `PONYSAY_SHELL_LINES` to 5, but only 1 if you are using Linux VT (TTY), your `~/.ponysayrc` may look like this:

```
if env[TERM] == 'linux':
    env[PONYSAY_SHELL_LINES] = 1
else:
    env[PONYSAY_SHELL_LINES] = 5
```

You can examine the source code of `ponysay` to figure out some nice hacking you may want to do, everything in the source code can be used directly as long as it is defined before `~/.ponysayrc` is interpreted.

`~/.ponysayrc` is a fallback for `~/.config/ponysay/ponysayrc`, which in turn is a fallback for `${XDG_CONFIG_HOME}/ponysay/ponysayrc`. If neither of those exist, `/etc/ponysayrc` is used if that exists.

3.7 Narcissistic ponies

The following will not work if you have line breaks in you file names, but if you do have that, you may want to rethink that as it will usually cause problems for programs, especially for shell scripts.

The command `__pony=$(ponysay -o 3>&1 1>/dev/null | grep ^pony\ file: | sed -e s/^pony\ file:\ //g) && (ponysay -of "$__pony" | ponythink -Wn -f "$__pony")` will give you a pony thinking of herself. The command works on GNU Bash, but may not work on less POSIX compatible shells. It works by first getting a random pony and then uses the extra information printed to the file descriptor 3 (see Section 3.1 [Extra information], page 7) to fetch the file name with help of `grep` and `sed`. The file name is stored in a shell variable.

It then the pipes output from one execution of ponysay into the second, using the stored file name in both. This does not work on FISH shell because of POSIX incompatibility.

Ponysay can use just about anything as a message because it quarantines the message's ANSI escape sequences, including colour. And is Unicode aware (including combining characters) and ANSI escape sequence aware.

Naturally this means that you can also make ponies think of eachother, for example: `ponysay -f rarity -b round 'My little Spiky-wiky' | ponythink -f spikefloat -b unicode -W n`

4 Environment variables

`ponysay` supports the follow environment variables:

PONYSAY_BOTTOM

Under TTY (Linux VT), if the output is larger the the screen's height, only the beginning is printed, leaving two blank lines. If you want the bottom to be printed rather the the beginning you can export `PONYSAY_BOTTOM` with the value `yes`, `y` or `1`.

PONYSAY_SHELL_LINES

Under TTY (Linux VT), if the output is larger than the screen's height, two lines are left blank. If you want more, or less, blank lines you can export `PONYSAY_SHELL_LINES` with the value of how many blank lines you want. Naturally this takes effect eve n if the output is not actually larger than the screen.

PONYSAY_FULL_WIDTH

You can export `PONYSAY_FULL_WIDTH` with the value `yes`, `y` or `1`, if you do not want the output to be truncated on the width to fit the terminal.

PONYSAY_TRUNCATE_HEIGHT

Export `PONYSAY_TRUNCATE_HEIGHT` with the value `yes`, `y` or `1`, if you want to truncate the output on the height even if you are not running `ponysay` under TTY.

PONYSAY_UCS_ME

Export `PONYSAY_UCS_ME` with the value `yes`, `y` or `1`, if you want [simulated] symlink to pony files using Universal Character Set in their names. Otherwise pony files uses only ASCII. If you want to remove the ASCII:ised names export `PONYSAY_UCS_ME` with the value `harder`, `h` or `2` instead.

If you have not enabled this, UCS names are not usable, suggested or listed. If you use `yes` UCS names will be usable, suggested and listed. If you use `harder` ASCII:ised names will not be suggested or listed, but they will still be usable.

PONYSAY_KMS_PALETTE
PONYSAY_KMS_PALETTE_CMD

`PONYSAY_KMS_PALETTE` or `PONYSAY_KMS_PALETTE_CMD` is used to tell ponysay how your TTY palette looks, this feature lets you get the best images in TTY if you have Kernel Mode Setting (KMS) support.

See Section 5.1 [KMS ponies], page 12, for information on how to use this.

PONYSAY_TYPO_LIMIT

ponysay is able to auto correct misspelled pony names and balloon style name. Without consideration for transpositioning, the distance between two words are measured in the number of edits needed to get from one word to the other, with weighting on some character changes used to favour spellos over typos.

By default if the weighted distance is greater than 5 for the closest words, auto correction ignored. This limit can be changed by exporting the limit to `PONYSAY_TYPO_LIMIT`; setting the limit to zero will disable auto correction.

PONYSAY_WRAP_HYPHEN

You can export what ponysay should use instead of a hyphen when wrapping messages. The hythen is red by default, if you want to change the colour or other formating, should should do so using the option `--colour-hyphen` (`--colour-wrap`).

PONYSAY_WRAP_LIMIT

Defines how long a word mush be to be hyphenated. This is used for to wrap words that are long so the output gets as pretty as possible. This s not the only condition

under which a word can be hyphenated, it can also be hyphenated if the word cannot fit otherwise. The default value is 8.

PONYSAY_WRAP_EXCEED
Defines how much a word must exceed the wrapping point to be hyphenated. This setting is used togather with PONYSAY_WRAP_LIMIT. The default value is 5.

5 Optional features

5.1 KMS ponies

KMS ponies is an optional feature that required that you have `util-say>=3` (`util-say<2` for `ponysay<2.1` and `util-say<3` for `ponysay<3`) installed. It lets TTY users that have a custom TTY colour palette and KMS support get best TTY images that can be display at the current state of the art. KMS is supported on most computers, but due to lack of published specifications Nvidia drivers does not support KMS. Other video cards can have problems too like 'gma500_gfx' due to lack of propper drivers and correct KMS support. `util-say` can be downloaded at `https://github.com/maandree/util-say`.

To use this feature your `~/.bashrc` (or equivalent for your shell) must keep track of your colour palette; it is not possible for a program to ask to terminal. Either the shell should export a palette string to `$PONYSAY_KMS_PALETTE` or you should export a command to can get the palette string to `$PONYSAY_KMS_PALETTE_CMD`. The palette string should be the stream which sets the colour palette to the terminal when `echo`:ed; preferably, to increase speed and reduce cache usage, it should be consistent every time it is exported for every colours palette. So you may want to keep it sorted, always be in either upper case or lower case, and not contain an character that is not used to set the colour palette.

Assuming you have a function in your `~/.bashrc`, to reset the colour palette to what you set it to last time in the terminal, named `reset-palette`, your `~/.bashrc` should, for example, contain:

```
[ "$TERM" = "linux" ] &&
    function ponysay
    {   export PONYSAY_KMS_PALETTE="$(reset-palette)"
        exec ponysay "$@"
    }
```

KMS ponies uses `/var/cache/ponysay/` or, if missing, `~/.cache/ponysay/` for cache space. You may also want to read Section 7.1 [Fill KMS cache], page 19.

6 Pony metadata

Pony files can contain metadata tags and a multiline comment. The following are the standard tags (comma separated lists may have whitespace surrounding the comma [,]):

GROUP NAME

If a pony file contains multiple ponies, it *should* have a *GROUP NAME* tag. The tag is a comma seperated list of the recognised names of the ponies as a groups, if the list is empty the tag value must be (none). An officiallity tag should be added to each name.

NAME

Every pony file should have this tag, one entry for each pony on the file. The value of the tag *must* be the pony's most common name as used on the TV show (or other source). If the pony's name have not been mentioned the value must be (not mentioned).

OTHER NAMES

If a pony in the pony file has other names then the one in *NAME* it *should* have this take for this pony. Any pony in the file (in case of multiple ponies) that do not need this tag should use the value (none).

The tag is a comma seperated list of alternative (to *NAME* names for the pony, each name should have an officiallity tag.

APPEARANCE

This tag specifies in which episode the pony first appeared. It reasonable to specify it even for ponies that appears in every episode.

For uniformity the format S%sE%e %t [%P] is recommended; [ ] denotes and optional part, optional in the sence that it does not apply the every episode, but it *should* be used if applicable. %s is the series (season) number in two digits, %e is the episode number in two digits. %t is the episode title and should use the standardised title format for the used format however without surrounding quotes if the used language has that, in the unlike event that [or] is present in the title it should be backslashed (\[, \]). %P is the part in the format [Part %p], where [ ] *does not* denote and optional part but rather is verbatim, and %p is the part number in one digit (well, if the part number is not 10 or higher).

The standard way to format titles in American English is the same as in British English, however it is not fully standardised. Capitalisation of the first word, and all other words, except for articles, prepositions, conjunctions, and forms of 'to be' is recommended.

Be aware that MLP:FiM episodes use American spelling which include a rather uncommon way to write for examples abbrevations (like for example Mr. instead of Mr), this may however not be the case for non-MLP:FiM episodes. And if there are not series (season) the series number defaults to 1, however other numbers and tags (which the part number is) may be added if required.

KIND

This tag decribes what kind of pony a pony is, it is a comma seperated lower case list, and it cannot be empty, by it can be (but shouldn't) skipped for every pony in the image.

Every fitting value should be used, however an alicorn (also known as alacorn, winged unicorn, acorn, pegasus unicorn, unipeg, unisus, horned pegasus, wing-horn, allacorn, cerapter pterippus, aquillacorn, pegasos aithiopikos, alate unicorns, or pegacorn (really all those name has been used throw history to define a unicorn with wing or pegasus with a horn)) should have the values alicorn and pony, but neither

`pegasus` nor `unicorn` or another of the possible therms mentioned. Earth ponies should have the value `pony` and `earth` not `earth pony`.

The standard values are (you may use other ones if fitting):

- `unicorn` For unicorn like `rarity` or presummed unicorns.
- `pegasus` For pegasi like `fluttershy`.
- `alicorn` For Alicorn, Winged pegasus, Pegacorn, etc. like `celestia`.
- `earth` For earth ponies like `applejack`.
- `pony` As a generic in case of unknow or sujetable like `headlesshorse`.
- `changeling` Changeling like `queenchrysalis` or `thorax`.
- `crystal` Any cristal pony like `twilightcrystal` or animal in they crystal form like `spikecrystal`.
- `seapony` Any seapony or by extenssion an orcapony from either novel, movies, comics, the series or the fandom like `sealyra`.
- `breezie` Breezies showed on the series or movies, this kind is based on the G3 breezies, but with they unique traits like speak in a nordic accent; examples are `seabreeze` or `twilightbreezie` from main ponydir.
- `animal` Applies to Spike and any other animal.
- `item` Applies to `tom` and Pinkamina's imaginary friends.
- `batpony` Applies to `royalnightguard` and others bat winged pony a like creatures, *Word of good* say these ponies are a different species but merchandice call them `pegasus`, so you need use `pegasus` and `batpony` comma separated one each other for uniformity. For other side `flutterbat` is just `flutershy` plus a fuit bat but not a natural `batpony` so this tag shall not be with she.

GROUP

This tag decribes which groups a pony classifies under, it is a comma seperated lower case list, and it cannot be empty, by it can be (but shouldn't) skipped for every pony in the image.

The standard values are (you may use other ones if fitting):

- `mare` Adult female pony.
- `stallion` Adult male pony.
- `filly` Female pony child.
- `colt` Male pony child.
- `baby` Baby ponies like `poundcake`.
- `dragon` Dragons (Spike, Crakle, Ember, Basil and the other dragons).
- `mane` The mane characters (also known as main characters [unponified] or protagonists).
- `wildlife` Wildlife, for example timberwolfs.
- `pet` A pony's pet, Spike does not count because Twilight does not play with him during pony–pet play dates.
- `royal` Royal pony, either by birth, marriage, or conquer (i.e. the old school style). Shining Armour is royal by marriage, but his biological family doesn't become royal by this, and Twilight Sparkle is because Princess Celestia give that tittle because she fullfilled they destiny.
- `griffon` Aplies to griffon (also called griffin) like Gilda and others.
- `villain` Villains, normally minions to antagonists or recurring bad ponies. Applies to changelings.

- `antagonist` (applies to: Nightmare Moon, Gilda, Discord, Chrysalis, Sombra, Tirek) Antagonists are also known as archivillians or archenemies. Nightmare Moon, Discord and Chrysalis are such, but Gilda also counts as one for her debut.
- `reconciled` (applies to Trixie, Discord, et al.) Any `villain` that change to the good side (aka become a good citizen or help the mane 6) count as one even if return to the dark side later, not count if they help for fulfill they malevolent plant.
- `deuteragonist` (applies to: the cutiemark cruisers) Deuteragonists are secondary characters, these are (as of seasson 3) only the Cutiemark Cruisers. The requirement is that thay are somewhat regular characters with dedicated episodes, but are not protagonists.
- `tritagonist` (applies to: Celestia, Luna, Cadance, Shining Armor, Spike) Important characters (excluding Derpy) that are neither protagonists, deuteragonists nor antagonists.
- `background` Background characters are not characters that are neither protagonists, deuteragonist, tritagonist, antagonist nor pets. They do not need to be strictly background characters, for example Big Mac and Cheerilee classifies under this group, as they are not too important to be considered tritagonists (as of seasson 3).
- `voiced` (only used together with background) Only `background` characters can be `voiced`. The additional requirement is that they have said something (ponies comics can also be voiced, in this case they need a minimal of one big dialoge text).
- `imaginary` Imaginary ponies (or other animal), in this group classify Tom and Pinkamina's imaginary friends for example.

BALLOON

For each balloon in the file (a pony file can have more than one balloon, but that is not common) their should one tag entry. There are four values that can be used:

- `top` The common setup, the balloon is at the top of the image
- `bottom` The balloon is at the bottom of the image
- `right` The balloon is neither at the top or at the bottom of the image, but is placed to the right of the pony
- `inside` The balloon is somewhere as inside the image

LINK ON   Files with only one pony *should not* use this tag. Specifies to which pony the link is connected, it is a number, starting from 1.

If a file contains Fluttershy and Pinkie (in that order, i.e. Pinkie is to the right of or below Fluttershy) and the link is connected to Pinkie, than the value should be 2.

In the rare case that the file contains multiple links (and multi ponies), the metadata should contains multiple entries of this tag, one entry for each link sorted in the order of the linkes placement in the image, in the same way ponies are ordered.

LINK   In the rare case that the file contains multiple links the metadata should contains multiple entries of this tag, one entry for each link sorted in the order of the linkes placement in the image, in the same way ponies are ordered.

The value for this tag must be either `regular` or `mirrored`. `regular` applies to linkes with NNE–SSW (\) direction. `mirrored` applies to linkes with NNW–SSE (/) direction, in version 3.0.1 only `rainbowdrag` uses this.

COAT   The name of the colour (as best estimated by you), in lowercase, that the pony's
           coat have. If the creature is (for example) a dragon, the colour of the scales is used.
           Common colour names are preferable. Only one colour should be named, but the
           name may describe a colour combination.

MANE   The name of the colour (as best estimated by you), in lowercase, that the pony's
           mane have. Common colour names are preferable. Only one colour should be named,
           but the name may describe a colour combination like `rainbow` for Rainbow Dash
           mane colour schema or `pastel` for Princess Celestia.

EYE   The name of the colour (as best estimated by you), in lowercase, that the pony's eyes
           have. Common colour names are preferable. Only one colour should be named, but
           the name may describe a colour combination. If the eyes are closed in the picture,
           use `close` in addition the the actual eye colour, separated by a comma.

AURA   The name of the colour (as best estimated by you), in lowercase, that the pony's
           magic aura have. Common colour names are preferable. Only one colour should be
           named, but the name may describe a colour combination.

           The magic aura is the colourisation around items that are affected by magic.

           If the pony file have multiple ponies, some with magicial abilities and some without,
           the ponies without magicial abilies should use the value (`no magic`). If the pony
           has magicial abilies but without an aura, use the value (`invisible`).

           Only humans [here we must call ourself humans rather than ponies, otherwise the
           next sentence does not make sense] can see the magic aura, describe the colour that
           we humans see, not ponies and other creatures in the TV Show [proof, see S01E11
           Winter Wrap Up and S02E25-26 A Canterlot Wedding] (Presumably Discord can
           see Magic too).

DISPLAY   This tag describes how a pony is places in the image. The standard values are:
           - `full` Full body)
           - `head` Just the head
           - `down` Upside down
           - `left` Pony is looking to our left
           - `right` Pony is looking to our right
           - `front` Pony is looking at us. `front` Can be combined with `left` and `right`,
             but `left` and `right` nor `full` and `head` cannot be combined.

WIDTH   The width of the pony image measured in text columns.

HEIGHT   The height of the pony image measured in text lines, this include the balloon (oc-
           cupies one line) even if it the first line with nothing else on that line.

BALLOON TOP
           The number of lines at the beginning of the pony image that should be skipped if
           the balloon is not printed.

BALLOON BOTTOM
           The number of lines at the end of the pony image that should be skipped if the
           balloon is not printed.

POSE   One word (preferably) to distinguish the pony files from other pony files with the
           same *MASTER*. It is preferable that this is written in bare infinitive. Master files
           should try to specify this tag but are not required to, however, non-master files
           (slave files) are required to specify this tag. **This tag is important for the extras
           feature to function.**

BASED ON
Either the name of a pony that the pony is based. If the original pony is not from MLP:FiM, the name a value that pony's MEDIA tag can used inside brackets, for example `(Tumblr)`, after the original pony's name. If the pony is not based on any pony the value `(original)` can be used. If the ponies is based on multiple ponies, make a comma separated list.

MASTER This tag refers to the pony file that is not named with extra attributes. For example, all files where Shining Armor is the (sole) speaking pony the this tag should be `shiningarmor`, except for in `shiningarmor.pony` where this tag may be omitted. **This tag is important for the ponyquotes feature to work.**

WING This is added manually only if needed, is the colour of wings from a pegasus or other alated creature (i.e.: Chrysalis).

SOURCE This tag specifies from where the pony image (not the file itself) originates. If the source is unknown the value should be `(unknown)`, if a GitHub user draw it the the value should be that user inside square brackets (in case of multiple artists, the tag is comma seperated list). Otherwise the source should be specified in any reasonable manner.

In order the claim authorship (the GitHub user value) it image must have been written from scratch (using templates is okay) or must be a major edit of another image. Just converting (including fixing the colours) an image (for example from the Internet or a screenshot) with or without removing the background is not enough.

MEDIA This tag *must not* be used for MLP:FiM ponies, but only for extraponies. It specifies the media from where the pony (not the image) originates.

LICENSE Which licence applies to the image? Full name and version should be used. In case of multiple license there should be one entry for each license. Omit this tag is the license is "not known".

The are two special cases here where this is no license. In which case it either uses regular copyright, in which case use the value `(regular)`, or everyone is the copyright holder (for example Public Domain), in which case use the value `(public)`.

FREE Is the image fully free? (For example Linux-libre is fully free, but not regular Linux.) The value *must* either be `yes`. `sharable` or `no`, or the tag must be omitted.

- `yes` mean free like Linux-libre
- `no` mean non free for `libre` distro
- `sharable` mean that you need permission from the author of the original image (or consept art like `aurora`) for inclusion with free ponies.

This is the most important tag as it helps us build a fully free version that can be officially distributed on FSF endorsed GNU/Linux distributions (GNU/Linux-libre) without problems.

Duplicate tags should be ordered in the order of the pony they describe from top-left to bottom-right in the image. It is important that if there are for example three ponies the image then all used tags that depends on the number of ponies in the image is used three times.

'Officiallity tag' refers the an annotation added to a tag value's list element. If the value is unofficial the string `(unofficial)` is appended (preferable with leading whitespace) to the element. If it is official the appended string is of the format `(official, %c)` (the brackets are verbatim), where `%c` is a comment. For example Chrysalis' name has not been mentioned in the show, however it is used in the manuscript and comics, therefore a pony file with Chrysalis should have the (partial) metadata:

    NAME: (not mentioned)

OTHER NAMES: Chrysalis (official, in manuscript and comic)

7 The tool chest

The tool chest is a collection of subcommands under the command `ponysay-tool`, its purpose is to provide tools to ponysay relevant actions that is not printing ponies (like the commands `ponysay` and `ponythink`).

7.1 Fill KMS cache

Before reading this section you may want to read the earlier section Section 5.1 [KMS ponies], page 12.

Invoking the command `ponysay-tool --kms` (no additional options are available) will pre-generate all kmsponies for your current TTY palette. This is useful if your computer is not fast enough, for you, at converting a pony to a kmspony. As the kmsponies may change between versions (noted in the change log if it happens) you may want to run this commmend after installing a new version of `ponysay`. Ponies that are already in the cache with the current KMS version will not be re-generated. May not work in all KMS drivers due to KMS inconsistences.

7.2 Metadata pasting

`ponysay-tool` allows you to copy, remove, stash and apply stashed pony metadata (but not merging, that must be done by hand.) The following commands does not support additional options.

`ponysay-tool --edit-rm PONY-FILE` will remove all metadata from the file `PONY-FILE`. To just remove some data you must use `ponysay-tool --edit PONY-FILE` or do it by hand. Note that you always use pony file, not pony names.

`ponysay-tool --edit-stash PONY-FILE` will print all metadata from a file to stdout. Cherry-picking cannot be done.

`ponysay-tool --edit-apply PONY-FILE` replace all metadata in a file with the metadata used provided in stdin.

To copy the metadata from one pony to another (and remove the old metadata) you will have to pipe the stashing and the applying command: `ponysay-tool --edit-stash SOURCE-PONY-FILE | ponysay-tool --edit-apply TARGET-PONY-FILE` (yes this is the trix mentioned on the man pages)

7.3 Editing metadata

`ponysay-tool` allows you to edit the metadata in a pony file by running `ponysay-tool --edit PONY-FILE`, where `PONY-FILE` is the pony file to edit, not the pony name. No additional options are available.

`ponysay-tool --edit PONY-FILE` is interative and opens an editor inspired by GNU Emacs. The tool will give you the standard tags to fill and will automatically fill in *HEIGHT* and *WIDTH* for you without allowing you to editing those two tags. Additionally the editor will print the pony at the right side of the terminal with the name of the file you are editing.

The commands the editor use is a small subset of the standard commands in GNU Emacs. Currently the commands are only coded for xterm (just about all terminals except Linux VT.) *C-x* means *x* with `control` held down. *M-x* means *x* with `alt` (`meta`) held down.

C-space
C-@   Set mark; only if mark is set and is at the same position as the point (cursor) the mark is deactivated. A mark creates a text select, it cannot span between lines.

C-k   Cut out the rest of the line and add it to the kill ring.

C-w   Cut out selected text and add it to the kill ring.

M-w   Add the selected text to the kill ring and unset the mark.

C-y   Paste (yank) text from the kill ring.

M-y   Cycle in the kill ring.

C-o   Insert a next line below the current line and go to it. This is useful if you want to
              add another entry for a tag.

C-j
enter   Go to next line, create a new line if at last line.

C-n
down   Go to next line, do not create a new line if at last line.

C-p
up   Go to previous line.

C-f
right   Go to next column.

C-b
left   Go to previous column.

home   Go to the beginning of the line.

end   Go to the end of the line.

backspace
C-h
C-?   Remove the previous character on the same line.

delete   Remove the current character on the same line.

insert   Enter or exit override mode.

C-x C-x   Swap the mark and the point.

C-x C-s   Save your changes.

C-x C-c   Exit the editor, do not forget to save if you have made changes.

7.4 Metadata collections

Pony metadata collection files are used by `ponysay` to by just reading one file per directory
determine all pony files metadata and determine which ponies will pass the `--restrict` option
when ponies are randomly selected.

A metadata colletion file's content a list, of pony files with and their corresponding metadata
as a map from tag name to tag value set, serialised with Python's cPickle module.

Running `ponysay-tool --metadata PONY-DIR` will generate the file `metadata` with the se-
rialised information. For use by the installer, the files to include can be explicity declared
appending their basename to the command.

7.5 Dimension files

Pony dimension files are used by `ponysay` to determine the size of all ponies and use that
information to determine which ponies fit the terminal and may be randomly selected.

Running `ponysay-tool --dimensions PONY-DIR` will generate three files `widths`, `heights`
and `onlyheights` to the directory `PONY-DIR`, the contain optimised information about the
widths, heigths and heights with printed without the balloon, respectively, for each pony the
the directory. For use by the installer, the files to include can be explicity declared appending
their basename to the command.

7.6 Pony browsing

Running `ponysay-tool --browse PONY-DIR`, or `ponysay-tool -b PONY-DIR` will display all ponies in `PONY-DIR` for you. You can limit the listed ponies by using the option `--restrict`, or `-r`, that works the same was in with the commands `ponysay` and `ponythink`. See Chapter 2 [Invoking ponysay], page 2, for more infomation about the `--restrict` option.

In this browser you will on the right side have all pony files, in your selected directory, listed except those that does not match your `--restrict` settings. In the rest of the free space, the pony select in the list is centered. You can move the pony, in case it is too big, by using the arrows keys with `control` held down, or using the `W`, `A`, `S`, `D` keys (for QWERTY and QWERTZ layout,) or with the `<` (or `Ä`), `A`, `O`, `E` keys (for Dvorak and Svorak layout.) To recenter the pony press `C-l` (`dl` with `control` held down.)

Browse between ponies using the arrow keys or with `C-n` and `C-p`, for next pony and previous pony, respectivily. Additionally, `Q` can be used list quotes for pony, and `I` for metadata; press the key again to return the pony browsing.

The tool can be exited using the key combinations `C-q` or `C-x C-c`.

8 Limitations

8.1 Terminals

Ponysay works perfectly on `xterm`, `xterm` like terminals including `putty`, settings may however need to be customised for Unicode Character Set (UCS) support, but less well, depending on font, on VTE based terminals including `mate-terminal`.

On Linux's native terminal Linux VT (TTY) it works less well, and not good at all without Kernel Mode Setting (KMS) support. See `https://github.com/erkin/ponysay/issues/1` for more information. `ponysay` clears the screen before printing to TTY, this is because if your graphics driver supports KMS, the colours will be messed by when the ponies position moves on the screen, this is also reason why the output is truncated on the height in TTY by default.

Most terminals have support for 256 colours, we do however only use the top 240 colours; this is because the lower 16 colours are usually, in contrast to the top 240, customised. We assume that the top 240 colours have their standard values. In TTY with KMS support we don't have any actual limit (except for 2^{24} + full transparency.)

`ponysay` works perfectly on `xterm`, `urxvt` and `putty`, but `rxvt`, `mrxvt` and `Eterm` do not have UTF-8 support and are currently not supported. Additionally `aterm` have neither UTF-8 support nor 256 colour support, and is therefore not yet supported.

Due to extreme limitations in `9term` `ponysay` will never be able to run on it.

Any Terminal runing in VT-50 compatibility mode lost many of the VT100 terminals capabilities like UTF-8 and 256 colour support, making impossible for `ponysay` run in that emulated mode indiferent if the terminal used can support those two requisites.

8.2 GNU Hurd

`ponysay` should work just fine on GNU/Hurd, except for in the native virtual terminal (TTY). Hurd's terminal is limited to 16 colours and does not provide the capaility of modifing.

If we are lucky it may be possible draw pictures, in full resultions, as you can in linux; which is currently not inplemented in `ponysay`. Another, not yet implemented possiblity, is to use low resolution with ACSII character for colour interpolation.

8.3 Cowsay

This section describes the limitation of `cowsay`, but since version 2.1 `cowsay` is no longer used because of it. So none of the following limitations are present anymore.

When `cowsay` determines the length of a word it measures in number of bytes (in UTF-8), therefore non-ASCII words will malformat the balloon with the message.

Further, `cowsay` does not recognise ANSI escape sequences, therefore, using colours and text styling in messages will also malformat the balloon with the message.

`cowsay` does not support balloon, including the link between the message and the pony, customisation, other than using `cowthink`. However you can modify `cowsay` (written Perl, so you can edit the installed files) to make the balloon look different, maybe using box drawing characters.

`cowsay` does not support setting the minimum size of the balloon, both directions on the balloon–pony links. or any other placement of the balloon than at the top to the left.

9 Problems and requests

9.1 External bugs

There is only one known bug that may occour that is external to ponysay, meaning that it is a bug that can occour with ponysay, but is not actually a bug in ponysay. This bug is common for programs that prints a lot of colours, even with `cat`; it is only known to happen for VTE-based terminals, such as `mate-terminal`, `gnome-terminal` and `terminator`.

The bug, is that lines (often no more than one line) can skipped when all lines move up one step, the next line is skipped instead, and so one; or that an escape sequence is interpreted as pure text (this is common in GNU Emacs, however GNU Emacs does not support programs such as `ponysay`.)

This bug can often be suppressed by piping the output to `cat` multiple times when using a VTE-based terminal (or always if you prefer). The use of VTE-based terminal can often be determined by checking for the environment variable *COLORTERM*, to which VTE-based terminal usually export their name (some reported another terminals name.)

If you want to do this in GNU Bash you can add this (with possible modifications depending on what you also have done with `ponysay`) code sample to your `~/.bashrc` file.

```
[[ ! "$COLORTERM" = "" ]] &&
function ponysay
{
    exec ponysay "$@" | cat | cat | cat | cat
}
```

It is important for this bug workaround that `cat` is unbuffered, which is default in GNU's version of `cat`, but not in Unix's version. If this does not work, test adding the option `-u` to `cat`.

9.2 Reporting bugs

If you find a bug in `ponysay`, install the last version from `https://github.com/erkin/ponysay`, and if it is still present, please report it at `https://github.com/erkin/ponysay/issues`. Please be as descriptive as possible, as it will help us verify it solve it faster. Wrong authorship, author name, alias and source count as bug, but in this case you need provide evidence that we are wrong, and who are the real artist, we not want fix a mistaken with another one.

9.3 Requesting ponies

If you want I specific pony added, ask us at `https://github.com/erkin/ponysay/issues` and we will add it. To speed the up the process, if possible, supply good pictures. Full visibly, transparent background, and pixelated are the properties that makes a picture good.

10 Dependencies

We have provided a script that should run one most, if not all shells, named
`./dependency-test.sh` that will help you track down any missing package. The script
works in `bash`, `dash` and `zsh`, but not in `fish`, so case you `sh` links to `fish`, run `bash
dependency-test.sh` (or with one of the other compatible shells.)

10.1 Required runtime dependencies

`coreutils`
> `stty` is used to determine the size of the terminal.

`python>=3`[1]
> `ponysay` is written in pure Python 3.

10.2 Optional runtime dependencies

`util-say>=3`
> For improved TTY support for user with custom colour palette and KMS support.
> It can be downloaded at `https://github.com/maandree/util-say`. If this is used
> `chmod` from `coreutils` is also required.

> For the purpose of simplifying for pony contributors, `ponysay` supports using .png-
> images (note that the file must not miss the `.png` at the end of the file name) in
> addition to .pony-files or pony names.

10.3 Package building dependencies

`python>=3`[2]
> Required to run the `./setup.py` file, which is also invoked from the make script.

`gzip`   Used for compressing manuals. (Optional, standard)

`xz`   Used for compressing manuals. (Optional, non-standard)

`coreutils`
> For set ownership on directories at build time. For set simbolic links on files.

`texinfo` Used to compile this `info` manual. (Optional, standard)

`info`[3]   Used to install this `info` manual with `install-info`. (Optional, standard)

10.4 Dependencies for pony providers

`bash`   Required to run `dev/dist.sh`.

`coreutils`
> `ln` and `readlink` are used in the `ttyponies` subscript of `dev/dist.sh`.

`util-say>=3`
> Used by `dev/dist.sh ttyponies` to build ttyponies from xterm ponies. It can be
> downloaded at `https://github.com/maandree/util-say`.

---
[1] Sometimes distributed as `python3` rather than `python`.
[2] Sometimes distributed as `python3` rather than `python`.
[3] Normally a part of `texinfo`.

11 Installing

11.1 From upstream

11.1.1 Installations basics

Before installing `ponysay`, make sure your system have the packages listed under Section 10.1 [Required runtime dependencies], page 24, and Section 10.3 [Package building dependencies], page 24, installed.

Tarballs can be downloaded at `https://github.com/erkin/ponysay/tarball/master` for bleeding edge, or from `https://github.com/erkin/ponysay/tags` for releases.

If you have `git` you can `clone` the project URL `https://github.com/erkin/ponysay.git`.

In the terminal, `cd` into the ponysay directory and execute `./setup.py --freedom=parital install` or `python3 ./setup.py --freedom=parital install`. This will install `ponysay` into `/usr`, normally meaning you need to run as root, e.g. by running `sudo ./setup.py --freedom=parital install`.

The `--freedom` option and manditory, if you only want completely free ponies, use `--freedom=strict` instead of `--freedom=partial`.

Now you will be to use ponysay, run: `ponysay "I am just the cutest pony!"`, or if have a specific pony in your mind: `ponysay -f pinkie "Partay!~"`.

`ponysay` comes with this `info` manual and a man page in section 6, `man 6 ponysay` (or just `man ponysay`). The man page is also available in Spanish, Swedish and Turkish: `man -L es 6 ponysay`, `man -L sv 6 ponysay`, `man -L tr 6 ponysay`. To install the localised manual, add the option `--with-man-es`, `--with-man-sv` or `--with-man-tr` when running `./setup.py`.

11.1.2 Custom installations

With the exception for with `--with-everything` and `--with-nothing`, every option that starts with `--with-` or `--without-` exists in both variants. `--with-` options install parts of the package. `--without-` options skips installation of parts of the packages. With the same exception, `--without-` options take not arguments and `--with-` optionally takes an argument, if no argument is provided a default argument is implied.

The configuration script recognised the following options, the default values for options with arguments are written after the equality sign (=) in the option:

`--everything`
`--with-everything`
> Install everything that is not explicity excluded.

`--minimal`
> Install only the essentials. Note that this can vary depending on version. Currently this means that the commands, xterm ponies and legal documents is installed.

`--nothing`
`--with-nothing`
> Install nothing, except legal documents, that is not explicity included.

`--with-ponysay`
`--with-ponysay-command=/usr/bin/ponysay`
> Install the `ponysay` command, and set file name. (Default)

`--with-ponythink`
`--with-ponythink-command=/usr/bin/ponythink`
> Install the `ponythink` command, and set file name. (Default)

```
--with-ponysay-tool
--with-ponyponysay-tool-command=/usr/bin/ponyponysay-tool
```
> Install the `ponysay-tool` command, and set file name. (Default)

```
--with-shared-cache=/var/cache/ponysay
```
> Install a user shared cache, this is only used by KMS ponies so far. (Default)

```
--with-bash
--with-bash-completion=/usr/share/bash-completion/completions/ponysay
```
> Install auto-completion for installed commands in GNU Bash. Select the file name
> for the installed script for the ponysay command, the other commands modifies this
> file name. (Default)

```
--with-fish
--with-fish-completion=/usr/share/fish/completions/ponysay.fish
```
> Install auto-completion for installed commands in Friendly interactive shell. Select
> the file name for the installed script for the ponysay command, the other commands
> modifies this file name. (Default)

```
--with-zsh
--with-zsh-completion=/usr/share/zsh/site-functions/_ponysay
```
> Install auto-completion for installed commands in the zsh shell. Select the file name
> for the installed script for the ponysay command, the other commands modifies this
> file name. (Default)

```
--with-shell
--with-shell-completion=/usr/share
```
> Macro for `--with-bash`, `--with-fish` and `--with-zsh`. The argument is the used
> share/ directory that all shells have in common.

```
--with-pdf
--with-pdf-manual=/usr/doc
```
> Install PDF manual, and select directory for it.

```
--with-pdf-compression
--with-pdf-manual-compression=gz
```
> Compress PDF manual, select compression by file name extension. This option does
> not imply `--with-pdf`. (Default)

```
--with-info
--with-info-manual=/usr/share/info
```
> Install `info` manual, and select directory for it. (Default)

```
--with-info-install
--with-info-manual-install=My Little Ponies for your terminal
```
> Use `install-info` when installing `info` manual. Set the description for the manual.
> This option does not imply `--with-info`. (Default)

```
--with-info-compression
--with-info-manual-compression=gz
```
> Compress `info` manual, select compression by file name extension. This option does
> not imply `--with-info`. (Default)

```
--with-man-en
--with-manpage-en
--with-man-manual-en
--with-en-man
--with-en-manpage
--with-en-man-manual=/usr/share/man
```
> Install English `man` manual. Set directory for `man` manuals. (Default)

```
--with-man-es
--with-manpage-es
--with-man-manual-es
--with-es-man
--with-es-manpage
--with-es-man-manual=/usr/share/man
```
> Install Spanish `man` manual. Set directory for `man` manuals.

```
--with-man-sv
--with-manpage-sv
--with-man-manual-sv
--with-sv-man
--with-sv-manpage
--with-sv-man-manual=/usr/share/man
```
> Install Swedish `man` manual. Set directory for `man` manuals.

```
--with-man-tr
--with-manpage-tr
--with-man-manual-tr
--with-tr-man
--with-tr-manpage
--with-tr-man-manual=/usr/share/man
```
> Install Turkish `man` manual. Set directory for `man` manuals.

```
--with-man
--with-manpage
--with-man-manual
```
> Macro for all `--with-man-LANG`.

```
--with-man-en-compression
--with-manpage-en-compression
--with-man-manual-en-compression
--with-en-man-compression
--with-en-manpage-compression
--with-en-man-manual-compression=gz
```
> Compress English `man` manual, select compression by file name extension. This option does not imply `--with-man-en`. (Default)

```
--with-man-es-compression
--with-manpage-es-compression
--with-man-manual-es-compression
--with-es-man-compression
--with-es-manpage-compression
--with-es-man-manual-compression=gz
```
> Compress Spanish `man` manual, select compression by file name extension. This option does not imply `--with-man-es`. (Default)

```
--with-man-tr-compression
--with-manpage-tr-compression
--with-man-manual-tr-compression
--with-tr-man-compression
--with-tr-manpage-compression
--with-tr-man-manual-compression=gz
```
        Compress Turkish `man` manual, select compression by file name extension. This option does not imply `--with-man-tr`. (Default)

```
--with-man-compression
--with-manpage-compression
--with-man-manual-compression
```
        Macro for all `--with-man-LANG-compression`.

```
--man-section-ponysay
--man-sectionpage-ponysay
--ponysay-man-section
--ponysay-manpage-section=6
```
        Change the section for the `ponysay` manpage.

```
--man-section-cowsay
--manpage-section-cowsay
--cowsay-man-section
--cowsay-manpage-section=1
```
        Change the section for the `cowsay` manpage.

```
--man-section-fortune
--manpage-section-fortune
--fortune-man-section
--fortune-manpage-section=6
```
        Change the section for the `fortune` manpage.

```
--with-ponies
```
        Install standard xterm ponies. (Default)

```
--with-ttyponies
```
        Install standard tty ponies. (Default)

```
--with-extraponies
```
        Install extra xterm ponies. (Default)

```
--with-extrattyponies
```
        Install extra tty ponies. (Default)

```
--with-quotes
```
        Install pony quotes. (Default)

```
--with-balloons
```
        Install balloon styles. (Default)

```
--with-ucs
--with-ucs-names
```
        Install UCS pony names. (Default)

```
--without-custom-env-python
```
        Let the installer set the `env` name for `python` in `ponysay`. (Default)

```
--with-custom-env-python=python3
```
        Set the `env` name for `python` in `ponysay`.

`--prefix=/usr`
> Set a prefix to all implicit directories.

`--private`
> Change all implicit configurations to fit local user a installation for the current user.

`--opt`   Change all implicit directories to fit installation to `/opt`.

`--bin-dir`
`--bindir=/usr/bin`
> Set the system's directory for command executables.

`--lib-dir`
`--libdir=/usr/lib/ponysay`
> Set the system's directory for non-command executables. Currently their is not non-executable library, so this options has no effect, but bleeding edge distributors should specify it if it differs from prefered.

`--libexec-dir`
`--libexecdir=/usr/libexec/ponysay`
> Set the system's directory for non-command executables. Currently their is not non-command executables, so this options has no effect, but bleeding edge distributors should specify it if it differs from prefered.

`--share-dir`
`--sharedir=/usr/share`
> Set the system's directory for resource files.

`--sysconf-dir`
`--sysconfdir=/etc`
> Set the system's local specific configuration directory.

`--cache-dir`
`--cachedir=/var/cache`
> Set the system's directory for cache directories.

`--dest-dir`
`--destdir=`
> Set off environment for installation.

`--linking=symbolic`
> Set how to link identical files. Directories cannot be hard linked on most systems, therefore directories or always symbolically linked of hard linked is specified. Recognised arguments are `copy`, `hard` and `symbolic`. `copy` implies that files and directories are not linked, but duplicated. `ponysay -L` will give the same output as `ponysay -l` if `copy` or `hard` is used. This is because it does link reading and not content or inode comparison.

`--freedom=MANDITORY!`
> Set your freedom. If you the any of the values `strict`, `full`, `true` or `yes`, the setup will make sure that only completly free parts of the package is installed. This should be used (`--freedom=strict`) on distributions for GNU endorsed (endorsable) GNU/Linux-libre distributions.
>
> If you do not want this, will need to explicity say so (you do also need to say if you do want it) by using either of the values `sloppy`, `partial`, `false` or `no`.

Recognised compressions are `gz` which uses `gzip -9`, and `xz` which uses `xz -9e`. `xz` is still exotic to most programs, using it is not recommended. Distributors are strongly disencouraged to compression for the PDF manual and should use `--without-pdf-compression`.

You can run `./setup.py [OPTIONS] view` to make sure everything is correct before building and installing.

11.2 Package repositories

11.2.1 Arch Linux

The official Arch Linux package repositories contains `ponysay` as `community/ponysay` (developer maintained). The Arch Linux User Repository (AUR) contains a bleeding edge git version of `ponysay` as `ponysay-git` (user maintained).

11.2.2 Arch Linux ARM

`community/ponysay` from Arch Linux (Section 11.2.1 [Arch Linux], page 30) is also available for Arch Linux ARM.

11.2.3 Chakra

Chakra users can install from (CCR) a stable version named a `ponysay` (developer maintained Arch Linux mirror), additionally a git verion of ponysay is available as `ponysay-git` (developer maintained Arch Linux mirror).

11.2.4 Debian GNU/Linux and Ubuntu

A .deb file is available at `http://roryholland.co.uk/misc.html#ponysay` (user maintained), and PPA:s can be found at `https://launchpad.net/~vincent-c/+archive/ppa` (user maintained) and `https://launchpad.net/~blazemore/+archive/ponysay` (user maintained). Note: Look Availabily before use one of these ppa, and check version because some are not updated often.

11.2.5 Gentoo Linux

Gentoo users can use the overlay `https://github.com/etu/aidstu-overlay`, which contains `ponysay` as `games-misc/ponysay` (developer maintained).

11.2.6 Source Mage GNU/Linux

The spell `util/ponysay` (user maintained) is available in Grimoire for Source Mage GNU/Linux. Note: They skip impar versions.

11.3 Exotic operating systems

An "exotic operating system" is a operating system that is not GNU (GNU/Linux ("Linux") and GNU/Hurd are GNU distro:s.)

Mac OS X users can use the `ponysay` Homebrew (`https://github.com/mxcl/homebrew`) formula to install `ponysay`.

Ponysay is also reported to be able to run on Windows 8 through Cygwin, provided that `python3` is installed. It will probably also run on any other version of Windows through Cygwin. Additionally among the preinstalled fonts in Windows 8; Consolas 10pt, and larger, can be used for almost perfect ponies (they may be just a little distorted on the height), however Consolas is only able print the ASCII based balloons.

Ponysay also runs on Windows Subsystem for Linux, be sure to install the latest Windows updates to enable true colour console support, then follow the install instructions for Ubuntu in a bash console.

11.4 Uninstalling

If you did not install `ponysay` with a package manager, but rather manually from the upstream, you can uninstall it by running `make uninstall`.

Well written package manages will uninstall files that the package is no longer using, i.e. if deleted, moved or renamed. To uninstall files that are not longer used, by the currently installed version you will need that versions `Makefile`. To perform an uninstallation of old files run `make uninstall-old`.

Note: make is no longer supported, instead use `setup.py` a `python 3` wrapper whit same syntaxis as the old make.

12 Inner workings

12.1 Pony anatomy

The pony files are simple raw output data that can be printed to the terminal, except it contains scalar variables. The pony images consists of white space, lower half blocks [U+2584], upper half blocks [U+2580] and ANSI colour sequences (CSI m), and, in TTY, colour value change sequences (OSI P).

Variables are recalled by putting the variable's name between two dollar signs (`$var$`), and are stored by putting the variable's name followed by the value between two dollar signs and with a equality sign between the name and the value (`$var=value$`). Variable names cannot include equality signs, but the value can; dollar signs can be used by placing an ESC character before the dollar sign.

There are three predefined variables: `$$` (empty variable name), `$\$` and `$/$`. `$$` has a dollar sign (`$`) as its value, while `$\$` and `$/$` contains the characters for the link to the balloon directed in the same direction as the variable name's slash. New for Ponysay 3.1 is `$X$`, which is a cross if the `$/$` directed balloon link and the `$\$` directed balloon link.

Variables whose name begin with `balloon` are parsed as balloon inserts, it can be either `balloon`, `balloonX`, `balloon,Y` or `balloonX,Y`, whether `X` is the minimum width of the balloon and `Y` is the minimum height of the balloon. New in Ponysay 3.0 is that the `X` can also be an range of columns, it contains of two numbers, the preferable start column, from the column that variables is placedon, the other number is the minimum width of the balloon. The two values are separated either by a `l`, a `r` or a `c`. If `l` is used the the balloon is printed as normal, except that it if wrapping is enabled and the balloon whould exceed the wrapping column, the balloon continues to fill on its left, at most as much as the position value. If `r` is used, the balloon fills the it's left first and then to its right. If `c` is used the balloon will try the fill on its left and right side equally.

Prior to version 2.1 the pony files were cow files used by `cowsay`, they are partial Perl-scripts that assign a value to a scalar variable named `$the_cow`. Cow files use a predefined scalar variable named `$thoughts`, these are used to create a link between the message and the pony. The message (and the balloon) itself was printed by `cowsay` and is not defined in the cow files.

12.2 Pony metadata extension

New in ponysay 3.0 is pony metadata, this feature is not supported in `util-say` (at least not yet). It extends the previously described[1] format of the pony files, by letting you specify details about the pony image, and the pony itself, as well as adding comments.

The metadata entry must be at the absolute beginning of the file (UTF-8 signture excluded), and is the file must be encoded in UNIX line breaks. The metadata entry begins with a line with exactly 3 dollar signs and nothing else (`$$$`), and end in the same way direct follow by the pony image starting from the next line.

A metadata tag consists of a tag name in upper case and a tag value, with a colon (`:`), optionally with surrounding regular spaces or tab spaces, but at least one regular space or tab space directly after the colon. The name can only consist of A to Z (upper case ASCII letters) and regular spaces. All tab spaces in the tag names and values are handled as regular spaces. Multiple tag names can be used multiple times or can be completely skipped. There are only a few tags, namely *BALLOON TOP*, *BALLOON BOTTOM*, *MASTER* and *FREE*, that absolutely should not be used muliple tag, but nor should *WIDTH* and *HEIGHT*; a general rule

---

[1] Section 12.1 [Pony anatomy], page 32

is that a tag desribing a pony should be duplicated exactly as many times as there are ponies in the image.

Any line that does not conform to the format of a tag line is a part of the comment field. Leading line breaks in the comment field is ignored.

12.3 Pony quote infrastructure

When compiling, pony quotes are built to `quotes/`, the file names are lists of ponies joined with plus signs (`+`) – the pony names are the same as the pony files, except they do not end with `.pony` – with a index at the end, and a full stop (`.`) before the index.

The source files are located in `ponyquotes/`, where their is a file named `ponies`. This file is called the pony map, and is the basis for how the compiled files are named. In the ponymap ponies with the same quotes are on the same line join together with plus signs (`+`), if the lines because too long for file names the line is split into multiple lines with the first pony in common.

In `ponyquotes/` there are also quote files, each contain just one quote, just as when compiled to `quotes/`. The source quote files are identical to the compiled quote files, except that their name contains just the first pony.

12.4 Balloon style files

Balloon style files are located in the directory `balloons/`, the ones ending with `.say` applies to `ponysay` and the ones ending with `.think` applies to `ponythink`.

Balloon style consists of 20 strings. Each string is defined on separate lines, by their name and their value separated with a colon (`name:value`), if the name is empty it continues the last one on a new line in the value. Only 10 of the strings may be multi-lined: *nw*, *nnw*, *n*, *nne*, *ne*, *sw*, *ssw*, *s*, *sse* and *se*.

The following strings are used, and must be defined in the files:

\   The character for the link to the balloon directed as \.

/   The character for the link to the balloon directed as /.

ww   The beginning of the balloon's line where the message is located if and only if the message contains only one line.

ee   The end of the balloon's line where the message is located if and only if the message contains only one line.

nw   The top left corner of the balloon.

nnw   If both this string and the *nne* string fits between the top corners, this is printed directly to the right of the top left corner.

n   The top edge of the balloon.

nne   If both this string and the *nnw* string fits between the top corners, this is printed directly to the right of the top left corner.

ne   The top right corner of the balloon.

nee   The end of the balloon's line where the message's first line is located if and only if the message contains more than one line.

e   The right edge of the balloon.

see   The end of the balloon's line where the message's last line is located if and only if the message contains more than one line.

se   The bottom right corner of the balloon.

sse   If both this string and the *ssw* string fits between the bottom corners, this is printed directly to the left of the bottom right corner.

s   The bottom edge of the balloon.

ssw   If both this string and the *sse* string fits between the bottom corners, this is printed directly to the right of the bottom left corner.

sw   The bottom left corner of the balloon.

sww   The beginning of the balloon's line where the message's last line is located if and only if the message contains more than one line.

w   The left edge of the balloon.

nww   The beginning of the balloon's line where the message's first line is located if and only if the message contains more than one line.

12.5 Printing in TTY with KMS

Since Linux VT (TTY) does not have capabilities for returning the position of the cursor, the screen must always be cleared before printing the ponies to make sure the pony's colours is not lost, i.e. reduced to mare 16 colours, during print. The colours are reduced if the pony's position on the screen is changed. This is only relevant with KMS support. The clear the screen we print "\e[H\e[2J" (\e is ESC) in at beginning. "\e[H" places the cursor at the beginning of the screen, and "\e[2J" clears everything on the screen after, and including at, the cursor. If we would use "\ec" (that is a reset), we would also turn off num. lock and caps. lock.

12.6 Truncation

Ponysay supports three type of output truncations, cutting away overflow on the right and truncation the height by either keeping the bottom or keeping the top. By default the latest is enabled under TTY, cutting away overflow on the right is always enabled by default.

Truncating the height in TTY is required under Kernel Mode Setting (KMS) support to keep the colours from being messed up when the ponies is moved in the screen during print. Prior to version 2.0 this was done either by piping to `head` (keeps the top) or by piping to `tail` (keeps the bottom.) `head` and `tail` takes as argument the number of lines to keep at most.

The size of the terminal, measured in characters, is fetched from `stty size`, which returns `HEIGHT WIDTH`, and `cut` it the used to get either the height or the width. This requires only GNU Coreutils; earlier `tput rows` and `tput cols` were used, this however required, the only de facto standard, package `ncurses`, some shells have environment variables for this.

Since version 2.1 truncation is done internally in the Python script, before that it was done in a custom C program `truncater`, that was installed to `/usr/lib/ponysay/truncater`. It recognised UTF-8 ANSI escape sequences, including OSI P and CSI m, which is essential for the truncation to be correct. It also expands tabs to every eighth column and resets the background colour when needed, and writes ANSI escape sequences that are on the left side of the truncation. The truncater stops CSI sequences on the first ASCII letter (`[a-zA-Z]`), but also stops escape sequences after the first character after the initial escape if it is not either `[` (CSI) or `]` (OSI). In the previous, C, program it supported UTF-8 by assuming that bytes do not match `10xxxxxx` and only those bytes were visible. This now fixed internally in Python, but has also been improved to exclude combining characters from the set of visible characters. Another difference is that the background colours are not reset, instead ANSI colours after the truncation point are still printed.

12.7 Languages

Before version 2.0 `ponysay` was written primarily in GNU Bash script; the truncater was however written in C, because it is simple, fast, does not pose addition dependencies, and is easy to do byte hacking in.

Sometimes shell is too slow, in these cases Perl was used; Perl was already required by `cowsay`, it is also similar to shell, but also supports hash tables.

However since version 2.0 we were trying to move from all there languages and only use Python 3, which as been accomplished in version 2.1.

12.8 Shell auto-completion

To make it easier to write auto-completion for shells, `ponysay` supports the options `--onelist`, `++onelist`, `--Onelist` and `--quoters`, which has no short versions. To make it even easier we use `auto-auto-complete` (`https://www.github.com/maandree/auto-auto-complete`) to generate auto-completion scripts, currently it supports `bash`, `fish` and `zsh`, the built system uses that program to generate completion for each shell.

Executing `ponysay --onelist` will list every available standard (MLP:FiM) pony, independent of where it is located, the output is a sorted and consists only of one pony per line.

Executing `ponysay ++onelist` will list every available extra (non-MLP:FiM) pony, independent of where it is located, the output is a sorted and consists only of one pony per line.

Executing `ponysay --Onelist` will list every available standard pony as well as extra pony, independent of where it is located, the output is a sorted and consists only of one pony per line.

`ponysay --quoters` work just as `ponysay --onelist`, excepts it limits the ponies to those that have quotes. Ponies that have quotes, but does not exist, i.e. does not have a .pony-file, are not listed.

Auto-completion scripts should not suggest these options.

12.9 Universal Character Set

In earlier versions of `ponysay` only the output truncation supported Universal Character Set, though handcoded UTF-8 character counting. Now `ponysay` lets Python decode the data, Python store all 31 bits of a character in as one character, not in UTF-16 as some other languages does, this means that the code is agnostic to the character encoding. However in Unicode 6.1 their are four ranges of combining characters, these do not take up any width in proper terminal, we therefore have a class in the code named `UCS` that help us take them into consideration when determine the length of a string.

Some ponies have names that contain non-ASCII characters, read about it in Chapter 4 [Environment variables], page 10. The UCS names are stored in the file `share/ucsmap`, in it lines that are not empty and does not start with a hash (#) are parsed, and contains a UCS name and a ASCII:ised name. The UCS name comes first, followed by the ASCII:ised name that the UCS name should replace or link towards. The two names are separated by and simple left to right arrow character [U+2192], optionally with surrounding white space.

It is important that the UCS names are stored in a file and not in file names, because it can cause problems on some platforms.

13 Contributing

13.1 Providing ponies

Most pony images are browser ponies or desktop ponies, browser ponies is a port of desktop ponies, implementing it in JavaScript. Browser ponies are available at `https://github.com/panzi/Browser-Ponies`. Desktop ponies are available at `http://desktop-pony-team.deviantart.com/`.

There is also a collection of ponies that are not yet pixelated in a Java reimplementation of the early Ponysay: `https://github.com/maandree/unisay/tree/develop/dev/newponies`

There is a checklist named `pony-checklist` at the `dev/` directory. You can use the check which ponies are added and which are not. Please update it when fit.

New ponies can be created from regular images by using util-say, which is available at `https://github.com/maandree/util-say`. Prior to version 2.1 of `ponysay`, `img2xterm` could be used, by since version 2.1 `ponysay` is using a new format that only util-say supports. `img2xterm` (`https://github.com/rossy2401/img2xterm`) was used in the early stage, but util-say tries to optimise the images in some aspects: as good as possible for low capability terminals, tries to place the pony–balloon link, displayed as good as possible when marked in the terminal (somewhat compromised by the first aspect,) and same width on all rows.

Using util-say:

```
img2ponysay -2 -- SOURCE_IMAGE > PONY_FILE

PONY_FILE should end with .pony and be localed in ponies/,
or extraponies/ if the pony is not a MLP:FiM pony.

Omit -2 if the source image does not use double pixel size.

For more information is available in util-say's info manual.
img2ponysay is a legacy command that uses the default settings of
ponytool for converting a image file to a pony file.
```

If you have util-say installed, which is required to build ponies, you can use PNG files as argument the for `ponysay -f`, this requires that the file is named `.png` at the end.

The following `bash` code will print the palette the ponies (the terminals) use:

```
c=16
while ((c < 256)); do
    echo -en "\e[48;5;${c}m  \e[49m"
    c=$(( $c + 1 ))
    if (( $(( c % 36 )) == 16 )); then
        echo
    fi
done; echo
```

For the palette to be correct, which is especially important when you draw ponies, you must not redefine the colours in the range 16 to 255 (inclusive).

When a pony is added please also add a ttypony version, i.e. the pony files used in TTY, but if you don't please state so in the pull request so we do not miss the create it; the simplest way to do this is to run `dev/dist.sh ttyponies` after adding the ponies to `ponies/`, running `dev/dist.sh ttyponies` will build (or rebuild) all ttyponies with a pony present in `ponies/`, and creates all needed symlinks.

To be able to run `dev/dist.sh ttyponies` you must have the packages listed under Section 10.4 [Dependencies for pony providers], page 24.

Also when adding new ponies, please map them up in the file `ponyquotes/ponies`. If the pony is a new pony without any other alternative image just add it to a new line, without `.pony`, preferably in its alphabetical position. If the file is a symlink add it to the same line as the target pony, and if the pony has and alternative image add it the the same line as that pony. Ponies on the same line are separated with a plus sign (`+`) without any white space. When a line is too long for a file name (this has happened to Pinkie Pie [`pinkie`],) it must be split into multiple lines, these lines should have their first pony file in common.

13.2 Pony naming guildlines

These are not rules, this are guildlines you can use when in doubt on how you want to name your ponies. That is, these are only suggestioned based on observion of current practice and discussions, it is probable that these suggestions are not optimal in complex cases.

Try to follow the MLP:FiM Wikia on `http://mlp.wikia.com/wiki/List_of_ponies`, if it is in conflict with an authorised game, such as Gameloft, it is more likely that that game has make an error, especially ignore a game on palette mismatch, and the Wikia is probabily more agreed with fist party merchandice, fans and crew at same time.

For in the wikia the order of precedence is:

- The series itself: In they first run, and uncensored
- The crew: M. A. Larson, Sabrina "Sibsy", Lauren Faust (For the first and second seasson only), et al.
- First party merchandice: Blind Bags and colectable toys [Further duscution is taken after any renamed]
- Second party merchandise: Any that claim to by official whitout logo and all the fancy stuff [further discusion first]
- Three party merchandise: Any that use the toys or characters but only in they tags, on-line stores and soo [Only if the evidence and use are far extended that a further discusion for renaming or not is needed]
- Script and others: Sometimes a name can become official if was used in the script or in CC for the official DVD.
- Placeholder: Any fan given name from a voted process in a open procees on `http://reddit.com/r/listofponies` where anypony can vote or give sugestions fallow they rules. These ones *CAN* potentially becomed official names or/and influence subsecuent discutions about renaming ponies.

Sometimes there are background ponies with the same palette and cutie mark, but of different kinds (earth pony, unicorn, pegasus and so on). In these cases the Wikia often list them as the same pony, they are given the same name. When this happens you can name the pony that appears first in the show just the name and append either 'earth', 'unicorn' or 'pegasus' depending on the kind. If the pony is a background alicorn, it is a princess and is titled as such, you can bet your sweet flank the Wikia will have given her a royal name [look ´Princess Erroria' second appearence].

When there are many alternative names, but no official, use one you think is most recognised or your personal favourite. It with short names when the names are similar. Symbolic links can help in this last case.

14 Distributing

If you are planning on maintaining `ponysay` in your favourite operating system you should first read Section 10.1 [Required runtime dependencies], page 24, and Section 10.2 [Optional runtime dependencies], page 24. If your OS does not follow Filesystem Hierarchy Standard (FHS), e.g. installing amusement binaries in `/usr/games` instead of `/usr/bin` or only supporting `/opt` equivalent directories you should read about configurations in Section 11.1.2 [Custom installations], page 25.

Apart from this, you should configure `ponysay` before building it with the option `--everything`. Otherwise only the `info` manual and the English man page will be installed for documentation.

Please inform us about your distribution so we can list it so everypony can see it.

The following is a reference distribution written in Arch Linux's PKGBUILD format. It is not complete, proper, verbose enough or well written, it just contains the core of an stable `git` distribution.

```
pkgname=ponysay
pkgver=3.0.1
pkgrel=1
arch=('any')
pkgdesc="Cowsay reimplementation for ponies"
url="https://github.com/erkin/ponysay"
license=('custom:GPL3' 'FDL')
depends=('python>=3' 'coreutils')
optdepends=("util-say>=3: Improved TTY support with KMS and PNG files"
"auto-auto-completion: Give autocompletion function for ponysay to bash, zsh and fish
makedepends=('git' 'texinfo' 'info' 'gzip' 'python>=3')
source=("ponysay::git+https://github.com/erkin/ponysay")
nd5suns=('SKIP')

build()
{
    cd "$srcdir/ponysay"
    git checkout "$pkgver"

    ./setup.py --everything --without-pdf-compression  \
            --bin-dir=/usr/bin --dest-dir="$pkgdir" \
            --freedom=parital build

    # CHANGE --freedom=parital to --freedom=strict
    # FOR ONLY COMPLETELY FREE PONIES,
    # useful for GNU/Linux-libre distributions
}

package()
{
    cd "$srcdir/ponysay"
    ./setup.py DESTDIR=$pkgdir install
}
```

15 Terminology

MLP:FiM The television show My Little Pony: Friendship is Magic.

My Little Pony
> The successor to My Pretty Pony, the toy not the short story by Stephen King.

TTY
Linux VT Linux's native terminal emulator. The name TTY comes from the file names for the devices used for terminals by Linux VT, which is `/dev/tty*`.

KMS
Kernel Mode Setting
> A feature in Linux allowing mode setting in kernel-space, this gives the TTY, for example better colour support. I would go to Wikipedia for more information.

ttyponies Pony files used in TTY.

kmsponies Pony files generated for use in TTY with custom TTY colour palette and KMS support.

extraponies
extra ponies
> Pony files of ponies that are not a part of MLP:FiM.

standard ponies
> Pony files of ponies that are a part of MLP:FiM.

systemponies
sysponies Pony files located in `/usr/share/ponysay`.

homeponies
usrponies Pony files located in `${XDG_DATA_HOME}/ponysay` or `~/.local/share/ponysay` (fallback).

browser ponies
> A JavaScript program which is the source for most of our ponies. It is a port of *desktop ponies*.

desktop ponies
> A program for Windows (need `.NET framework 4`), OSX (need `Mono 3.0.2` and `XQuartz`) and Unix (need `mono` and `mono-basic`) this say they README file, but is reported have really many problems or simply not work outside Windows) that run gif pixelated ponies like those ones on `ponysay` on your screen with interactivity one and each other, quotes and one or two games. Programmed by RoosterDragon with help of artists like Botchan, Jay Wright and many others artist.

ponification
> The process of converting English text to Equestrian English. The process if transforming a non-pony artowork, character history or another thing into a pony version of that.

Equestrian English
> The English dialect spoken by the ponies in MLP:FiM, the basic role is that it is American English with as many words and parts of words as possible exchanged to words having to do with ponies, including the work 'pony' itself. This is normally the language we, the developers, write in, except we may use another English, e.g. British English or Basic English, as the base language.

best.pony   The pony you think is [the] best pony. It should be a symlink pony. It is a feature
affecting the `-f`, `+f`, `-F` and `-q` options.

pony symlink
symlink pony
A pony file that is a symbolic link to another pony file. Symbolic links can be
created with the command `ln -s TARGET SYMLINK`.

ponyquotes
A feature enabling ponies to quote them self from MLP:FiM.

environment variables
Variables stored to the environment with the command `export VARIABLE=VALUE`.
The variable name is often written with the prefix `$` due to have they are read in
shell, using the command `echo $VARIABLE`.

UCS
Universal Character Set
The set of of character, develop by the Unicode Consortium. It defined a partially
filled space of 2^{31} characters, some of which are not glyphs.

combining characters
Character that have zero width and is used to compose characters with diacritical
when there is no precomposed character to use.

ASCII
ASCII character
American Standard Code for Information Interchange (ASCII) defines 128 charac-
ters, some are not glyphs. It contains control characters, basic punctuation, the
decimal digit, and lower case and upper case English alphabet characters `a-z`.

short options
Command line arguments starting with either exactly one hyphen (`-`) or exactly
one plus sign (`+`), and have exactly one character beyond that. They may be argu-
mentless, argumented, optionally argumented, or variadic (consumes all following
arguments).

long options
Command line arguments starting with either at least two hyphens (`-`) or at least
two plus signs (`+`), beyond that they have at least one character, but often at least
one work. They by be argumentless, argumented, optionally argumented, or variadic
(consumes all following arguments).

completion
auto-completion
shell completion
shell auto-completion
Provided by a shell dependent script, argument suggestion is provided of then by
pressing the tab key.

ANSI escape sequences
escape sequences
Character sequences starting with a ESC character, with a special interpretation for
terminals standardise by ANSI.

ANSI colour sequences
ANSI colours
colour sequences

ANSI escape sequences defining a colour or other formatting, known as CSI m, a sequence starting with CSI and ending with an `m`. This is extended to 256 colours, from 16 colours, by `xterm` which is de facto standardise.

CSI   The character combination ESC followed by `[`, used in standardised ANSI escape sequences.

OSI   The character combination ESC followed by `]`, used in non-standardised ANSI escape sequences.

Appendix A Change log

Version 3.0.3

- New ponies: `auntorange`, `caballeron`, `cocopommel`, `coriander`, `ember`, `flurryheart`, `fluttershybat`, `fluttershybathang`, `goldiedelicious`, `grace`, `kevin`, `kingthorax`, `moondancer`, `saffron`, `starlightglimmer`, `stellareclipse`, `sunburst`, `sunburstcape`, `thesmooze`, `thorax`, `tirek`, `treehugger`, `troubleshoes`, `twilightbreezie`, `whoanelly`
- New extraponies: `cocoachill`, `riverbeauty`
- Pony symlink added:
    - `cookiecrumbles` → `raritysmom`
    - `bettyboufant` → `raritysmom`
    - `hondoflanks` → `raritysdad`
    - `royalpin` → `pokeypierce`
    - `cloudyquartz` → `sue`
    - `cocopommel` → `misspommel`
    - `horsemd` → `doctop`
    - `fleurdelis` → `fleurdislee`
    - `blinkie` → `limestone`
    - `inky` → `marble`
    - `clyde` → `igneousrock`
    - `hughjelly` → `hughbertjellius`
    - `drcaballeron` → `caballeron`

    `orange` was renamed to `uncleorange` to not conflict with `auntorange`. `cocopommel` was renamed to `misspommel` in the serie and merchandise a symlink was provided pointing to the old name. `maudepie` was renamed to `maudpie` since that her real name, we just miswritted it.
- Code is cleaned and improved.
- Initial fullwidth CJK characters support.

Version 3.0.2

- New ponies: `ahuizotl`, `brucemane`, `cheesesandwich`, `coldheart`, `creampuff`, `cremebrulee`, `deepblue`, `fleetfoot`, `filmreel`, `flashsentry`, `hairytipper`, `ironwill`, `mane-iac`, `mantishy`, `maudepie`, `maybelle`, `misty`, `mule`, `nightmarerarity`, `peachbottom`, `poundfly`, `pumpkincake`, `rainbowdashcrystal`, `rainbowdrop`, `rainbowfim`, `seabreeze`, `shoeshine`, `shortround`, `strawberrycream`, `sunsetshimmer`, `tealove`, `theoldenpony`, `unclewing`, `yearling`, `wildflower`
- New extraponies: `aurora`, `aquarius`, `archlinux`, `aries`, `barbara`, `buttonmom`, `calamity`, `cancer`, `capricorn`, `childrenofthenight`, `chrome`, `coffeetalk`, `coffeewalk`, `drizzle`, `firefox`, `fluffle`, `freckles`, `fyreflyready`, `gemini`, `gnupony`, `internetexplorer`, `kingsley`, `kingsleybanner`, `leo`, `libra`, `lilmissrarity`, `littlepip`, `milky`, `milkylay`, `opera`, `oscura`, `pisces`, `princeartemis`, `reddit`, `robodash`, `sagittarius`, `scorpio`, `seabreeze`, `snowdrop-crew`, `solaris`, `starstruck`, `sweetiebot`, `taurus`, `twibrain`, `virgo`, `wiggles`
- Pony symlink added:
    - `walter` → `waltercoltchack`

- barbra → barbara
- buttonmash → highscore
- brad → flashsentry
- ie → internetexplorer
- dancefever → hairytipper
- snowheart → coldheart
- bulkbiceps → snowflake
- mannyroar → manticore

- Default value for `-W`, the message wrapping column, has been changed from 40 to 65, to wrap messages better.
- `nightmare` has renamed into `nightmaremoon` for concistency with the comic
- `pipsqueak` from ponies has renamed to `pipsqeakpirate` and they extraponies counterpart moved to ponies as now is canon they normal appearance.
- `ironwill` (3.0.1 and lower) has been renamed to `ironwillwalk` because now we have a non walk version to be master.
- `doctor` has ben reintegrated to ponies thanks to official comic issue 8 where `drhooves` is siting alongside `doctor` and in front of `celestia`
- Turkish manual page added.
- Swedish manual page added.
- `+h` (`++help`, `--help-colour`) added.
- `auto-auto-completion` now is no more part of `ponysay`, therefor you need install them and rebuild to get *auto-completion*, otherwise these *auto-completion* files aren't installed.

Version 3.0.1

- New ponies: `harshwhinny`
- All ponies has been reviewed and improved when needed.
- The `ponysay-tool` command is now installed.

Version 3.0

- New ponies: `applesplit`, `amira`, `babseed`, `bear`, `beautybrass`, `bigmacsleep`, `billneiigh`, `cadancecrystal`, `cadancescruffy`, `cloudchaser`, `descent`, `gingersnap`, `haakim`, `hayseed`, `jubileena`, `featherweight`, `fiddlesticks`, `flitter`, `lighningdust`, `midnightstrike`, `mrbreezy`, `orangebird`, `orangefrog`, `pansyshy`, `pinkiecrazyface`, `princesserroria`, `poundcake`, `raccoon`, `rainbowblitz`, `rarity`, `ravenearth`, `ravenunicorn`, `royalunicornguard`, `rumble`, `shiningarmorcrystal`, `sombra`, `spikecrystal`, `squirrel`, `sweetcream` (IDW Comic Issue #1), `trixieamulet`, `twilacorn`, `twilightcrystal`, `twilightfly`, `twilightpricess`, `twilightwings`, `twinkleshine`
- New extraponies: `donutpony`, `gleamingshield`, `hastelle`, `johndelancie`, `jristz`, `maandree`, `orion`, `pipsqueak`, `pardise`, `pizzapony`, `snowdrop`, `tempo`, `ticket`
- `lotusbloosom` has been renamed to `lotusblossom` (typo)
- `maredowellgallop` has been renamed to `maredowellgallop` (need a version named as master)
- `ironwillwalk` has been renamed to `ironwillwalk` (need a version named as master)
- Renamed option `-F` to `+f` and option `--F` to `++f`, `-F` and `--F` has new definitions.
- Environment variable `PONYSAY_TYPO_LIMIT` has been added.

- Environment variable `PONYSAY_WRAP_HYPHEN` has been added.
- Environment variable `PONYSAY_WRAP_LIMIT` has been added.
- Environment variable `PONYSAY_WRAP_EXCEED` has been added.
- Added support for `~/.ponysayrc` with the alternatives: `${XDG_CONFIG_HOME}/ponysay/ponysayrc` and `~/.config/ponysay/ponysayrc` as well as the global fallback `/etc/ponysayrc`
- `-f`, `+f` and `-q` may be unargumented if that are at the end of the command line.
- `ponysay-tool` is introduced, it can be used to edit, remove and copy pony meta data, and more.
- `ponysay-tool --kms` generates all kmsponies for the current TTY palette.
- Pony metadata tags *BALLOON TOP* and *BALLOON BOTTOM* can be used to specify how much extra height the balloon causes at the top and at the bottom of the pony.
- `${XDG_DATA_HOME}/ponysay/*` is allowed in favour of `${HOME}/.local/share/ponysay/*`
- Balloons can be have and explicit minimum column span with placement justification.
- Only ponies that fit the terminal will be randomly selected (for directory with pony dimension files generated), however if no pony fits, any of the can be randomly selected.
- Setup option `--sysconf-dir` with default value `/etc` added,
- New manditory setup option `--freedom`.
- Pony metadata options added: `--info`, `++info` and `--restrict`.
- `fillycelestia` and `filliestia` has been moved to `extraponies`.
- `shadowbolts` has been renamed to `nightingale` (shadowbolts split)
- `lily` has been renamed to `lilyvalley` (official name)
- `sweatiesing` has been renamed to `sweetising` (name consistency)
- `carecake` has renamed into `carrotcake` (official full name)
- `peppermoon` has renamed to `pepermoon` (typo)
- `maria` has been renamed to `danger` (name given by the author)
- `maliot` renamed to `melilot` (typo)
- Pony symlink added:
    - `lily` → `lilyvalley`
    - `sweetiedrops` → `bonbon`
    - `carecake` → `carrotake`
    - `berrydreams` → `blueberry`
- The license has been changed to the GNU General Public License version 3+, from WTFPL 2.

Version 2.9.1

- Bug fix: correction in the -W option broke the -o option.

Version 2.9

- New ponies: `pinkieumbrelahatfear`, `twilighttime`
- New extraponies: `molestia` (Tumblr)
- The option `-q` works like `-f` and `-F`, it takes one argument, and may be used multiple times for more arguments.
- The old option `-q` is renamed to `--q`.

- The options `--f` and `--F` has been added.
- Weighted distance for autocorrection on pony names and boolean style name is set to 5, rather than unlimited. Currently this cannot be modified (without editing the source code.)
- If file descriptor 3 is definied when `ponysay` is executed, extra information is printed to it.
- Arguments starting with `n` or `i` is allowed for `-W`.

Version 2.8

- New ponies: `airheart`, `bastionyorsets`, `gustavelegrand`, `milkyway`, `peppermoon`, `pinkacopter`, `pinkiefly`, `pinkieparade`, `pinkieumbrellahat`, `shiningarmorwedding`, `soaringofficer`, `starlight`, `sunnyrays`, `sweatiesing`, `tenderheart`, `tom`, `twilightspike`, `zecorabalance`
- New extraponies: `applejack` (Tumblr), `applejack-63`, `artemis`, `blueberry`, `butterscotch`, `drhoovesdiscorded` (Tumblr), `duskshine`, `elusive`, `rainbowblitz`
- Pony symlink added:
  - `georgewashingtony` → `bastionyorsets`
- Support for explicit hyphenation using soft hyphens had been added to the word wrapper.
- Support for explicit non-word wrapping using non-breaking space had been added to the word wrapper.
- The word wrapper colours the inserted hyphens in red.
- Support for terminal capabilities emulation with the flags `-X`, `-V` and `-K`.
- Support for printing just the pony, using the flag `-o`.
- Colouring option flags are added.
- Automatic correction of incorrectly spelled pony names and balloon style names added.

Version 2.7

- New ponies: `basil`, `cloudkicker`, `cerberus`, `cow`, `derpysad`, `flowertrio`, `frederickhorseshoepin`, `horsemd`, `jeffletroski`, `jesuspezuna`, `joe`, `joetuxedo`, `manticore`, `meadownsong`, `meliot`, `pinkiegummydisguise`, `seaswirl`, `theodoredonaldkerabatsos`, `turf`, `waltercoltchak`
- New extraponies: `blueballblitz` (Varous fanfics, Shadowbolt), `drhooves1`, `drhooves2`, `drhooves3`, `drhooves4`, `drhooves5`, `drhooves6`, `drhooves7`, `drhooves8`, `drhooves9`, `drhooves10`, `drhooves11`, `nyx` (Fanfic: Past Sins), `nyxdisguised` (Fanfic: Past Sins), `pinkaminacupcake` (Fanfic)
- `cracky` is renamed to `crackle`.

Version 2.6

- New ponies: `applebloomdance`, `blueberry`, `blueberrycake`, `blueharvest`, `candylicious`, `cherrycola`, `cracky`, `cutiemarkcrusaders`, `derpybags`, `derpycloud`, `firestreak`, `hughjelly`, `lemonhearts`, `lyrabonbon`, `noi`, `pictureperfect`, `poppycock`, `quickfix`, `silverspeed`, `rainbowhurricane`, `rainbowshadowbolt`, `silverspeed`, `surprise` (wonderbolt), `thunderlane`, `timeturner`, `twilightthebearded`
- New extraponies: `faust` (alicorn), `maria` (Moonstuck, seapony), `posey` (Tumblr), `slanderpony`, `sparkler` (Tumblr), `twilight` (Tumblr)
- Pony symlink added:
  - `bonbonlyra` → `lyrabonbon`
  - `epona` → `quickfix`

- clockwork → quickfix
  - drhooves → timeturner
  - lotusbloosom → lotus
- doctor and doctornohat has become extraponies, because their mane style is differenct from in the TV show. And timeturner no longer links to any of them.
- ./configure and make is no longer support.

Version 2.5.1

- New extraponies: sealyra
- Build system as compatibility with standard GNU Make build system.

Version 2.5

- Brand new highly configurable build system.
- UTF-8 as I/O encoding is enforced. (Critical bug fix for ASCII locale users.)

Version 2.4

Nothing worth mentioning.

 Note: Identifies itself as version 2.3

Version 2.3

- Support for best.pony file.
- -q accepts file names.
- Improved Unicode support: treats combining characters as invisible.
- Optional support for UCS pony names.
- Pony files and balloon style files can be pipes (as well as sockets, doors and as always regular files.)
- Support cowsay style message compression.
- New ponies: blaze
- New extraponies: fyrefly (Tumblr), surprise (Tumblr), woona (moonstuck), woonanohat (moonstuck)
- Pony symlink added:
  - pinkieoink → oinkoinkoink
- Support for non-MLP:FiM ponies (known as extraponies). This is implemented with the options -F, +l, and +L corresponding to -f, -l, and -L.

Version 2.2

- Full support for arbitrary positioning of balloon in pony files.
- ANSI colour sequences in pony files are applied only to the pony image, not the balloon link or the balloon itself.
- Support for colours in the message.
- Support custom balloon styles using the option -b, -B will list all available. This list depends on whether you are invoking ponysay or ponythink

Version 2.1.1

Nothing worth mentioning.

Version 2.1

- `applebumkin` is renamed to `applebumpkin`.
- New ponies: `owlowiscious`, `purplehaze`
- Cowsay has be reimplemented, and have full Unicode support and support for `figlet` style messages.
- Deleted environment variables: `PONYSAY_COWSAY`, `PONYSAY_COWTHINK`
- You will need Python 3, but not GNU Bash, Perl or Cowsay.
- New .pony file format is used: `unisay`'s format instead of `cowsay`'s Perl based format. This includes arbitrary position of balloon, mirrored balloon links, and minimum size of balloon.

Version 2.0

- Makefile is generated by running `./configure`.
- All Perl scripts and almost all Bash are reimplemented in one Python 3 script.
- kmsponies4ponysay is included.

Version 1.4.1

- Code is repaired and more portable.

Version 1.4

- Make file is improved.

  Note: Identifies itself as version 1.3

Version 1.3

- New ponies: `forestspirit`, `hollydash`, `raggedy`, `rhyme`
- `sindy` is renamed to `powderrouge`.
- Pony symlink added:
    - `sindy` → `powderrouge`
- `PREFIX=/some-dir` can be used when invoking `make`, the default value is `/usr`

Version 1.2

- ponyquotes4ponysay is included.
- Support for extension: kmsponies4ponysay.
- Pony symlinks added:
    - `mrsparkle` → `nightlight`
    - `elsie` → `prettyvision`
- New ponies: `ace`, `blueblood`, `filthyrich`, `gingergold`, `hayfever`, `highscore`, `junebug`, `mrsparkle`, `persnickety`, `ponet`, `screwloose`, `tornadobolt`.
- `elsie` is renamed to `prettyvision`.
- Arbitrary spaces in `-f` argument is not longer accepted (it causes problems with file names including spaces.)

  Note: Identifies itself as version 1.1

Version 1.1

- Man pages are compressed before installation.
- `info` manual added.
- Shell completion for `ponythink` added, in addition to `ponysay`.
- `fish` completion added.
- `/usr/lib/ponysay` is used instead of `/usr/bin` for code used by the main script.
- `~/.local/share/ponysay` is used for private pony directories.
- `ncurses` is no longer needed for determining the screen's size, `coreutils` is used instead.
- Pony symlinks added:
    - `amethyststar` → `sparkler`
    - `berrypinch` → `ruby`
    - `craftycrate` → `boxxy`
    - `magnum` → `raritysdad`
    - `pearl` → `raritysmom`
    - `powderrouge` → `sindy`
    - `royalribbo` → `violet`
- New ponies: `blossomforth`, `bonvoyage`, `cadance`, `celestiasmall`, `changelingqueen`, `cherryberry`, `discordamused`, `discordpuppetmaster`, `fleurdelishair`, `fleurdelislay`, `owl`, `perrypierce`, `petunia`, `pinacolada`, `skyra`, `truffleshuffle`.
- Pony spelling removed: `fillycadence`.
- Pony symlink change: `perry` → {`pokey` → `perrypierce`}.
- Option `-L` added, lists ponies with symlink mapping.
- Support for extension: ponyquotes4ponysay.
- Accepts arbitrary spaces in `-f` argument.

Version 1.0

- Spanish translation of the man page is added.
- New ponies: `applecore`, `applejackscarecrow`, `bonbonstand`, `changeling`, `chrysalis`, `cottoncloudy`, `diamondmint`, `discord`, `fillycadence`, `flam`, `fleurdelis`, `flim`, `fluttershyshy`, `fluttershystare`, `lyrasit`, `oinkoinkoink` (is pinkie), `philomenaphoenix`, `pinkiecannon`, `pinkiecannonfront`, `pinkiecannonhappy`, `pinkiegummy`, `pinkiehugfluttershy`, `pinkiehugsfluttershy`, `pinkiepartycannon`, `pinkieprincess`, `pinkiesilly`, `pinkietongue`, `pinkiewhoops`, `pinkiewhoopseat`, `pinkiewhoopsout`, `rainbowdrag`, `rainbowsalute`, `rainbowshine`, `raritydrama`, `shiningarmor`, `shiningarmorguard`, `snowflake`, `spikemustache`, `stevenmagnet`, `stevenmagnettrue`, `twilightcrazyfromball`, `twilightrage`, `twilightzero`, `wildfire`.
- Pony symlinks added:
    - `djpon-3` → `vinyl`
    - `fillycadance` → `fillycadence`
    - `horsepower` → `snowflake`
- Improved TTY support: ponies have low colours resolution, instead of monochrome, when the high colour resolution is not available.

Version 0.10

- Man page manual added.
- The directories for pony directories are changed from `/usr/share` to `/usr/share/ponysay` and ˜ to `˜/.ponysay`.
- Pony symlinks added:
    - `carrottop` → `carrot`
    - `goldenharvest` → `carrot`
    - `harpass` → `lyra`
    - `heartstrings` → `lyra`
    - `lulamoon` → `trixie`
    - `minuette` → `colgate`
    - `noteworthy` → `blues`
    - `perry` → `pokey`
    - `pokeypierce` → `pokey`
    - `timeturner` → `doctornohat`
    - `trixielulamoon` → `trixie`
    - `twilightvelvet` → `mrssparkle`
- Support for truncating output on height, enabled by default under TTY.
- Environment variables added: `PONYSAY_FULL_WIDTH`, `PONYSAY_SHELL_LINES`, `PONYSAY_TRUNCATE_HEIGHT`, `PONYSAY_BOTTOM`.

Version 0.9

- Output truncated on width to fit screen.
- Support for TTY (Linux VT).
- `bash` completion added.
- `zsh` completion added.
- New ponies: `allie`, `archer`, `boxxy`, `carecake`, `cupcake`, `daringdo`, `davenport`, `fancypants`, `ironwillwalk`, `lily`, `lunafly`, `maredowellfly`, `maredowellgallop`, `master`, `mjolna`, `orange`, `raritysdad`, `raritysmom`, `royalnightguard`, `ruby`, `sparkler`, `violet`.

Version 0.8

- New ponies: `aloe`, `angle`, `applebloom`, `applebumkin`, `applefritter`, `berrypunch`, `bigmac`, `blinkie`, `blues`, `braeburn`, `caesar`, `candymane`, `caramel`, `cheerilee`, `cheerilee80`, `clyde`, `colgate`, `colton`, `daisy`, `derpystand`, `derpystandwing`, `diamondtiara`, `dinky`, `doctornohat`, `elsie`, `fido`, `fillycelestia`, `fillydash`, `fillydashfly`, `fillyjack`, `fillyjacktravel`, `fillypinkie`, `fillypinkiecurly`, `fillyrarity`, `fillyshy`, `fluttershygala`, `gilda`, `gildastand`, `granny`, `grannychair`, `grannysleep`, `gummy`, `hoity`, `horte`, `inky`, `laflour`, `lightning`, `lintsalot`, `lotus`, `mayor`, `mrssparkle`, `nightmare`, `opal`, `parasprite`, `philomena`, `photofinish`, `pinkamina`, `pinkiebounce`, `pinkiechicken`, `pinkiegala`, `pipsqueak`, `pokey`, `rainbowfly`, `rainbowgala`, `rainbowsleep`, `raindrops`, `rarityfly`, `raritygala`, `rarityponder`, `redheart`, `rocky`, `rose`, `rover`, `royalguard`, `sapphire`, `scootaloo`, `screwball`, `shadowbolts`, `silverspoon`, `silverstar`, `sindy`, `snails`, `snips`, `soarin`, `soigne`, `spike`, `spikefloat`, `spikelove`, `spot`, `stella`, `strongheart`, `sue`, `suedance`, `tank`, `trixiestage`, `trixiestand`, `turnip`, `twist`, `winona`.

Note: Identifies itself as version 0.7

Version 0.7

- New ponies: `carrot`, `octavia`, `trixie`, `vinyl`, `zecora`.
- Support for listing ponies with `-l` option.

Version 0.6

- New ponies: `bonbon`, `celestia`, `doctor`, `fillistia`, `spitfire`, `woona` (not moonstuck).
- Dropping usage of utility `which`, using `hash` instead.

  Note: Identifies itself as version 0.5

Version 0.5

- Using utility `which` to determine existence of `cowsay`.

Version 0.4

- `.cow` files are removed.
- Support for `-W` option.
- Select random pony if not specified.
- `-f` supports file names, and not only pony names.

Version 0.3

- Fixed use of `.pony` files.

Version 0.2

- Pony files end with `.pony` instead of `.cow`.
- `lyrasleep` is renamed to `lyra`.
- `.cow` files are kept but not used.
- `-h` prints proper help.

Version 0.1

First release.
- Includes the ponies: `applejack`, `derpy`, `derpysit`, `fluttershy`, `luna`, `lyrasleep`, `pinkie`, `rainbow`, `rarity`, `sweetie`, `twilight`.

Appendix B Ponysay contributors

Active developers and major contributors of ponysay:
- Erkin "erkin" Batu Altunbaş
- Mattias "maandree" Andrée
- Elis "etu" Axelsson
- Sven-Hendrik "svenstaro" Haase
- Pablo "jristz" Lezaeta
- Jan Alexander "heftig" Steffens

Patchers and other contributors of ponysay:
- Duane "Marneus68" Bekaert
- Kyah "L-four" Rindlisbacher
- James "rossy2401" Ross-Gowan
- Louis "kragniz" Taylor
- Daniel "gtmanfred" Wallace
- Jannis "sycoso"
- "spider-mario"

Appendix C Ponysay license

Ponysay is release by Erkin Batu Altunbaş et al.

Copyright © 2012, 2013, 2014 Erkin Batu Altunbaş et al.

Ponysay is Free Software (and Open Source) and in licensed under the terms of GNU General Public License version 3, or at your option, any later version.

You have the four essential freedoms:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbour (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

If you intend to redistribute ponysay or a fork of it commercially, it contains aggregated images, some of which may not be commercially redistribute, you would be required to remove those. To determine whether or not you may commercially redistribute an image make use that line `FREE: yes`, is included inside the image between two `$$$` lines and the `FREE` is and upper case and directly followed by the colon.

> ponysay is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> ponysay is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with ponysay, and attached to this document. If not, see <http://www.gnu.org/licenses/>.

Appendix D GNU General Public License

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. `http://fsf.org/`

Everyone is permitted to copy and distribute verbatim copies of this
license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

   "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

   To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

   A "covered work" means either the unmodified Program or a work based on the Program.

   To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

   To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

   An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a. The work must carry prominent notices stating that you modified it, and giving a relevant date.

   b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

   c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable

section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything

designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

    c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

    d. Limiting the use for publicity purposes of names of licensors or authors of the material; or

    e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

    f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

    You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

    A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

    A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

    Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

    In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

    If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

    If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRIT-

ING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PRO-
GRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR
IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE
RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH
YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
    WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MOD-
    IFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE
    TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
    CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE
    THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA
    BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
    PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
    PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED
    OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local
    legal effect according to their terms, reviewing courts shall apply local law that most closely
    approximates an absolute waiver of all civil liability in connection with the Program, unless
    a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public,
the best way to achieve this is to make it free software which everyone can redistribute and
change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start
of each source file to most effectively state the exclusion of warranty; and each file should have
at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or (at
your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it
starts in an interactive mode:

```
program Copyright (C) year name of author
```

```
       This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
       This is free software, and you are welcome to redistribute it
       under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html.

Appendix E GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
`http://fsf.org/`

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix F Concept and program index

G

H

I

K

L

M

N

O

P

Q

R

Appendix G Variable and option index

A

B

C

D

E

F

G

H

K

L

M

N

O

P

S

W