```
                           \\\///
                          /  _  _  \
                         (|  (.)(.)  |)
               .---------.000o--()--o000.---------.
               |                                  |
               |        __  _          _    _     |
               |       / _(_)  __ _ | | ___ | |_   |
               |      |  |_| |/ _` | |/ _ \ __|    |
               |      |   _| | (_| | |  __/ |_     |
               |      |_| |_|\__, |_|\___|\__|     |
               |            |___/                  |
               |                                  |
               '---------.ooo0------------------'
                         (   )   0ooo.
                          \ (     (   )
                           \_)     ) /
                                  (_/
```

Codeblock class: figlet

```
    sudo apt-get install figlet
    http://www.figlet.org

    runs:
    > figlet {im_opt} < code-text

    class->cmd
      figlet -> figlet
```

Metadata options
```
    imagine.im_out: img,fcb
    imagine.im_log: 4
    imagine.figlet.im_out: stdout,fcb
```
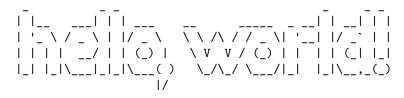
Figlet

figlet

```
     _____       __        __
    / __(_)___ _/ /__  / /_
   / /_/ / __ `/ / _ \/ __/
  / __/ / /_/ / /  __/ /_
 /_/ /_/\__, /_/\___/\__/
        /____/
```{.figlet im_opt="-f slant" caption="Figlet"}
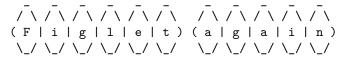figlet
```

hello, world!

```
 _          _ _                             _     _ _
| |__   ___| | | ___    __      _____  _ __| | __| | |
| '_ \ / _ \ | |/ _ \   \ \ /\ / / _ \| '__| |/ _` | |
| | | |  __/ | | (_) |   \ V  V / (_) | |  | | (_| |_|
|_| |_|\___|_|_|\___( )   \_/\_/ \___/|_|  |_|\__,_(_)
                    |/
```

```figlet
hello, world!
```

Figlet again.

```
  _   _   _   _   _   _      _   _   _   _   _
 / \ / \ / \ / \ / \ / \    / \ / \ / \ / \ / \
( F | i | g | l | e | t )  ( a | g | a | i | n )
 \_/ \_/ \_/ \_/ \_/ \_/    \_/ \_/ \_/ \_/ \_/
```

```{.figlet im_opt="-f bubble"}
Figlet again
```

Documentation

figlet -h

```
Usage: figlet [ -cklnoprstvxDELNRSWX ] [ -d fontdirectory ]
              [ -f fontfile ] [ -m smushmode ] [ -w outputwidth ]
              [ -C controlfile ] [ -I infocode ] [ message ]
```

man page

NAME
       FIGlet - display large characters made up of ordinary screen char-
       acters

SYNOPSIS
       figlet [-cklnoprstvxDELNRSWX] [-d fontdirectory]
              [-f fontfile] [-m layoutmode]
              [-w outputwidth] [-C controlfile]
              [-I infocode] [message]

DESCRIPTION
       FIGlet prints its input using large characters (called ``FIGchar-
       acters'')made up of ordinary screen characters (called ``sub-char-
       acters''). FIGlet output is generally reminiscent of the sort of
       ``signatures'' many people like to put at the end of e-mail and
       UseNet messages. It is also reminiscent of the output of some
       banner programs, although it is oriented normally, not sideways.

       FIGlet can print in a variety of fonts, both left-to-right and
       right-to-left, with adjacent FIGcharacters kerned and ``smushed''
       together in various ways.   FIGlet fonts are stored in separate
       files, which can be identified by the suffix ``.flf''. In systems
       with UTF-8 support FIGlet may also support TOIlet ``.tlf'' fonts.
       Most FIGlet font files will be stored in FIGlet's default font
       directory.

       FIGlet can also use ``control files'', which tell it to map cer-
       tain input characters to certain other characters, similar to the
       Unix tr command.   Control files can be identified by the suffix
       ``.flc''. Most FIGlet control files will be stored in FIGlet's
       default font directory.

       You can store FIGlet fonts and control files in compressed form.
       See COMPRESSED FONTS.

USAGE
       Just start up FIGlet (type ``figlet'') and then type whatever you
       want. Alternatively, pipe a file or the output of another command
       through FIGlet, or put input on the command line after the
       options. See EXAMPLES for other things to do.

OPTIONS
     FIGlet reads command line options from left to right, and only the
     last option that affects a parameter has any effect. Almost every
     option has an inverse, so that, for example, if FIGlet is custom-
     ized with a shell alias, all the options are usually still avail-
     able.

     Commonly-used options are -f, -c, -k, -t, -p and -v.

     -f fontfile
          Select the font. The .flf suffix may be left off of font-
          file, in which case FIGlet automatically appends it.
          FIGlet looks for the file first in the default font direc-
          tory and then in the current directory, or, if fontfile was
          given as a full pathname, in the given directory. If the
          -f option is not specified, FIGlet uses the font that was
          specified when it was compiled. To find out which font
          this is, use the -I3 option.

     -d fontdirectory
          Change the default font directory. FIGlet looks for fonts
          first in the default directory and then in the current
          directory. If the -d option is not specified, FIGlet uses
          the directory that was specified when it was compiled. To
          find out which directory this is, use the -I2 option.

     -c
     -l
     -r
     -x   These options handle the justification of FIGlet output.
          -c centers the output horizontally. -l makes the output
          flush-left. -r makes it flush-right.   -x (default) sets
          the justification according to whether left-to-right or
          right-to-left text is selected. Left-to-right text will be
          flush-left, while right-to-left text will be flush-right.
          (Left-to-right versus right-to-left text is controlled by
          -L, -R and -X.)

     -t
     -w outputwidth
          These options control the outputwidth, or the screen width
          FIGlet assumes when formatting its output. FIGlet uses the
          outputwidth to determine when to break lines and how to
          center the output. Normally, FIGlet assumes 80 columns so
          that people with wide terminals won't annoy the people they
          e-mail FIGlet output to. -t sets the outputwidth to the

7

terminal width. If the terminal width cannot be deter-
mined, the previous outputwidth is retained. -w sets the
outputwidth to the given integer. An outputwidth of 1 is a
special value that tells FIGlet to print each non-space
FIGcharacter, in its entirety, on a separate line, no mat-
ter how wide it is.

-p
-n   These options control how FIGlet handles newlines. -p puts
        FIGlet into ``paragraph mode'', which eliminates some
        unnecessary line breaks when piping a multi-line file
        through FIGlet.   In paragraph mode, FIGlet treats line
        breaks within a paragraph as if they were merely blanks
        between words. (Specifically, -p causes FIGlet to convert
        any newline which is not preceded by a newline and not fol-
        lowed by a space character into a blank.) -n (default)
        puts FIGlet back to normal, in which every newline FIGlet
        reads causes it to produce a line break.

-D
-E   -D switches to the German (ISO 646-DE) character set.
        Turns `[', `\' and `]' into umlauted A, O and U, respec-
        tively.   `{',  `|' and `}' turn into the respective lower
        case versions of these. `~' turns into s-z. -E turns off
        -D processing.   These options are deprecated, which means
        they probably will not appear in the next version of
        FIGlet.

-C controlfile
-N   These options deal with FIGlet controlfiles. A controlfile
        is a file containing a list of commands that FIGlet exe-
        cutes each time it reads a character. These commands can
        map certain input characters to other characters, similar
        to the Unix tr command or the FIGlet -D option. FIGlet
        maintains a list of controlfiles, which is empty when
        FIGlet starts up.   -C adds the given controlfile to the
        list. -N clears the controlfile list, cancelling the
        effect of any previous -C. FIGlet executes the commands in
        all controlfiles in the list. See the file figfont.txt,
        provided with FIGlet, for details on how to write a con-
        trolfile.

-s
-S
-k
-W

-o   These options control how FIGlet spaces the FIGcharacters
        that it outputs. -s (default) and -S cause ``smushing''.
        The FIGcharacters are displayed as close together as possi-
        ble, and overlapping sub-characters are removed. Exactly
        which sub-characters count as ``overlapping'' depends on
        the font's layoutmode, which is defined by the font's
        author. -k causes ``kerning''. As many blanks as possible
        are removed between FIGcharacters, so that they touch, but
        the FIGcharacters are not smushed. -W makes FIGlet display
        all FIGcharacters at their full width, which may be fixed
        or variable, depending on the font.

        The difference between -s and -S is that -s will not smush
        a font whose author specified kerning or full width as the
        default layoutmode, whereas -S will attempt to do so.

        If there is no information in the font about how to smush,
        or if the -o option is specified, then the FIGcharacters
        are ``overlapped''. This means that after kerning, the
        first subcharacter of each FIGcharacter is removed. (This
        is not done if a FIGcharacter contains only one subcharac-
        ter.)

-m layoutmode
        Specifies an explicit layoutmode between 1 and 63. Smush-
        modes are explained in figfont.txt, which also provides
        complete information on the format of a FIGlet font. For
        the sake of backward compatibility with versions of FIGlet
        before 2.2, -m0 is equivalent to -k, -m-1 is equivalent to
        -W, and -m-2 is equivalent to -s. The -m switch is nor-
        mally used only by font designers testing the various lay-
        outmodes with a new font.

-v
-I infocode
        These options print various information about FIGlet, then
        exit. If several of these options are given on the command
        line, only the last is executed, and only after all other
        command-line options have been dealt with.

        -v prints version and copyright information, as well as a
        ``Usage: ...'' line. -I prints the information corre-
        sponding to the given infocode in a consistent, reliable
        (i.e., guaranteed to be the same in future releases) for-
        mat.   -I is primarily intended to be used by programs that

use FIGlet. infocode can be any of the following.

-1 Normal operation (default).
      This infocode indicates that FIGlet should operate
      normally, not giving any informational printout,
      printing its input in the selected font.

0 Version and copyright.
      This is identical to -v.

1 Version (integer).
      This will print the version of your copy of FIGlet
      as a decimal integer. The main version number is
      multiplied by 10000, the sub-version number is mul-
      tiplied by 100, and the sub-sub-version number is
      multiplied by 1. These are added together, and the
      result is printed out. For example, FIGlet 2.2 will
      print ``20200'' , version 2.2.1 will print
      ``20201''. Similarly, version 3.7.2 would print
      ``30702''. These numbers are guaranteed to be
      ascending, with later versions having higher num-
      bers. Note that the first major release of FIGlet,
      version 2.0, did not have the -I option.

2 Default font directory.
      This will print the default font directory. It is
      affected by the -d option.

3 Font.
      This will print the name of the font FIGlet would
      use.   It is affected by the -f option. This is not
      a filename; the ``.flf'' suffix is not printed.

4 Output width.
      This will print the value FIGlet would use for out-
      putwidth, the number of columns wide FIGlet assumes
      the screen is. It is affected by the -w and -t
      options.

5 Supported font formats.
      This will list font formats supported by FIGlet .
      Possible formats are ``flf2'' for FIGfont Version 2
      .flf files and ``tlf2'' for TOIlet .tlf files.

If infocode is any other positive value, FIGlet will simply
exit without printing anything.

```
-L
-R
-X        These options control whether FIGlet  prints  left-to-right
          or  right-to-left.   -L selects left-to-right printing.  -R
          selects right-to-left printing.  -X (default) makes  FIGlet
          use whichever is specified in the font file.

          Once the options are read, if there are any remaining words
          on the command line, they  are  used  instead  of  standard
          input  as  the  source  of text.  This feature allows shell
          scripts to generate large letters without having  to  dummy
          up standard input files.

          An  empty  argument,  obtained  by  two  sequential quotes,
          results in a line break.
```

EXAMPLES

```
     To use FIGlet with its default settings, simply type

          example% figlet

     and then type whatever you like.

     To change the font, use the -f option, for example,

          example% figlet -f script

     Use the -c option if you would prefer centered output:

          example% figlet -c

     We have found that the most common use  of  FIGlet  is  making  up
     large  text  to  be  placed  in e-mail messages.  For this reason,
     FIGlet defaults to 80 column output.  If you  are  using  a  wider
     terminal, and would like FIGlet to use the full width of your ter-
     minal, use the -t option:

          example% figlet -t

     If you don't want FIGlet to smush FIGcharacters into  each  other,
     use the -k option:

          example% figlet -k

     If  figlet  gets its input from a file, it is often a good idea to
```

```
    use -p:

            example% figlet -p < myfile

    Of course, the above can be combined:

            example% figlet -ptk -f shadow < anotherfile
            example% figlet -cf slant

    Finally, if you want to have FIGlet take the input from  the  com-
    mand line instead of a file:

            example% figlet Hello world
```

  Other Things to Try
    On many systems nice effects can be obtained from the lean font by
    piping it through tr. Some you might want to try are the follow-
    ing:

```
            example% figlet -f lean | tr ' _/' ' ()'
            example% figlet -f lean | tr ' _/' './\\'
            example% figlet -f lean | tr ' _/' ' //'
            example% figlet -f lean | tr ' _/' '/  '
```

    Similar things can be done with the block font and many of the
    other FIGlet fonts.

COMPRESSED FONTS
    You can compress the fonts and controlfiles using the zip archiv-
    ing program. Place only one font or controlfile in each archive,
    and rename the archive file (which will have a name ending in
    .zip) back to .flf or .flc as the case may be. If you don't
    rename the file appropriately, FIGlet won't be able to find it.

    FIGlet does not care what the filename within the .zip archive is,
    and will process only the first file.

    The .zip format was chosen because tools to create and manipulate
    it are widely available for free on many platforms.

THE STANDARD FONTS
    Here are a few notes about some of the fonts provided with FIGlet.
    You can get many other font from the Web site
    http://www.figlet.org/   This location should also contain the
    latest version of FIGlet and other related utilities.
```

The font standard is the basic FIGlet font, used when no other font is specified. (This default can be changed when FIGlet is compiled on your system.) The controlfiles 8859-2, 8859-3, 8859-4, and 8859-9 are provided for interpreting those character sets, also known as ISO Latin-2 through Latin-5 respectively. The character set 8859-1 (ISO Latin-1) is FIGlet's default and requires no special controlfile.

Closely related are the fonts slant, shadow, small, smslant (both small and slanted), smshadow, (both small and shadowed), and big. These fonts support only Latin-1, except that big supports Greek FIGcharacters as well; the controlfiles frango (for Greek text written in Latin characters, so-called ``frangovlakhika''), and 8859-7 (for mixed Latin/Greek text) are provided.

The ivrit font is a right-to-left font including both Latin and Hebrew FIGcharacters; the Latin characters are those of the standard font. The available controlfiles are ilhebrew, which maps the letters you get by typing on a U.S. keyboard as if it were a Hebrew keyboard; ushebrew, which makes a reasonable mapping from Latin letters to Hebrew ones; and 8859-8, which supports mixed Latin/Hebrew text. Warning: FIGlet doesn't support bidirectional text, so everything will come out right-to-left, even Latin letters.

The fonts terminal, digital, and bubble output the input character with some decoration around it (or no decoration, in the case of terminal). The characters coded 128 to 159, which have varying interpretations, are output as-is. You can use the appropriate controlfiles to process Latin-2, Latin-3, or Latin-4 (but not Latin-5) text, provided your output device has screen or printer fonts that are appropriate for these character sets.

Two script fonts are available: script, which is larger than standard, and smscript, which is smaller.

The font lean is made up solely of `/' and `_' sub-characters; block is a straight (non-leaning) version of it.

The font mini is very small, and especially suitable for e-mail signatures.

The font banner looks like the output of the banner program; it is a capitals and small capitals font that doesn't support the ISO Latin-1 extensions to plain ASCII. It does, however, support the Japanese katakana syllabary; the controlfile uskata maps the

upper-case and lower-case Latin letters into the 48 basic katakana
characters, and the controlfile jis0201 handles JIS 0201X (JIS-
Roman) mixed Latin and katakana text. Furthermore, the banner
font also supports Cyrillic (Russian) FIGcharacters; the con-
trolfile 8859-5 supports mixed Latin and Cyrillic text, the con-
trolfile koi8r supports the popular KOI8-R mapping of mixed text,
and the controlfile moscow supports a sensible mapping from Latin
to Cyrillic, compatible with the moscow font (not supplied).

The fonts mnemonic and safemnem support the mnemonic character set
documented in RFC 1345. They implement a large subset of Unicode
(over 1800 characters) very crudely, using ASCII-based mnemonic
sequences, and are good for getting a quick look at UTF-8 unicode
files, using the controlfile utf8.

ENVIRONMENT
     FIGLET_FONTDIR
          If $FIGLET_FONTDIR is set, its value is used as a path to
          search for font files.

FILES
     file.flf   FIGlet font file
     file.flc   FIGlet control file

DIAGNOSTICS
     FIGlet's diagnostics are intended to be self-explanatory. Possi-
     ble messages are

               Usage: ...
               Out of memory
               Unable to open font file
               Not a FIGlet 2 font file
               Unable to open control file
               Not a FIGlet 2 control file
               "-t" is disabled, since ioctl is not fully implemented.

     This last message is printed when the -t option is given, but the
     operating system in use does not include the system call FIGlet
     uses to determine the terminal width.

     FIGlet also prints an explanatory message if the -F option is
     given on the command line. The earlier version of FIGlet, version
     2.0, listed the available fonts when the -F option was given.
     This option has been removed from FIGlet 2.1. It has been
     replaced by the figlist script, which is part of the standard
     FIGlet package.

ORIGIN
        ``FIGlet''  stands  for  ``Frank, Ian and Glenn's LETters''.
        Inspired by Frank's .sig, Glenn wrote (most of) it, and Ian
        helped.

        Most of the standard FIGlet fonts were inspired by signatures on
        various UseNet articles. Since typically hundreds of people use
        the same style of letters in their signatures, it was often not
        deemed necessary to give credit to any one font designer.

BUGS
        Very little error checking is done on font and control files.
        While FIGlet tries to be forgiving of errors, and should (hope-
        fully) never actually crash, using an improperly-formatted file
        with FIGlet will produce unpredictable output.

        FIGlet does not handle format characters in a very intelligent
        way.   A tab character is converted to a blank, and vertical-tab,
        form-feed and carriage-return are each converted to a newline. On
        many systems, tabs can be handled better by piping files through
        expand before piping through FIGlet.

        FIGlet output is quite ugly if it is displayed in a proportion-
        ally-spaced font. I suppose this is to be expected.

        Please report any errors you find in this man page or the program
        to <info@figlet.org>

WEBSITE AND MAILING LIST
        You can get many fonts which are not in the basic FIGlet package
        from the Web site http://www.figlet.org/   It should also contain
        the latest version of FIGlet and other utilities related to
        FIGlet.

        There is a mailing list for FIGlet for general discussions about
        FIGlet and a place where you can ask questions or share ideas with
        other FIGlet users. It is also the place where we will publish
        news about new fonts, new software updates etc.

        To subscribe or unsubscribe from the FIGlet mailing list, please
        send   email   to   figlet-subscribe@figlet.org or figlet-unsub-
        scribe@figlet.org   or   visit   the   following   web   page:
        http://www.figlet.org/mailman/listinfo/figlet

AUTHORS

Glenn Chappell did most of the work. You can e-mail him but he is
not an e-mail fanatic; people who e-mail Glenn will probably get
answers, but if you e-mail his best friend:

Ian Chai, who is an e-mail fanatic, you'll get answers, endless
conversation about the mysteries of life, invitations to join some
473 mailing lists and a free toaster. (Well, ok, maybe not the
free toaster.)

Frank inspired this whole project with his .sig, but don't e-mail
him; he's decidedly an un-e-mail-fanatic.

Gilbert "The Mad Programmer" Healton added the -A option for ver-
sion 2.1.1. This option specified input from the command line; it
is still allowed, but has no effect.

John Cowan added the -o, -s, -k, -S, and -W options, and the sup-
port for Unicode mapping tables, ISO 2022/HZ/Shift-JIS/UTF-8
input, and compressed fonts and control files. He also revised
this documentation, with a lot of input from Paul Burton.

Claudio Matsuoka added the support for .tlf files for version
2.2.4 and performs random hacks and bugfixes.

As a fan of FIGlet, Christiaan Keet revised the official FIGlet
documentation and set up the new FIGlet website at
http://www.figlet.org/   (and   the   corresponding
ftp://ftp.figlet.org/pub/figlet/)

SEE ALSO
        figlist(6), chkfont(6), showfigfonts(6), toilet(1)