

```
-----  
/  This is  \  
\  ducksay! /  
-----
```

```
  \  
  \  
    --  
    >( ' )  
      )/  
      /(  
      /  '-----/  
      \   ~=-  /  
~~~~~
```

```
-----  
( But which Version? )  
-----
```

```
  \  
  \  
    >()_  
    (__)__ _  
  
  \  
  \  
    ^__^  
    (oo)\_____  
    (__)\\    )\\/  
        ||----w |  
        ||     ||
```

```
-----  
( by Jonathan P. Spratte )  
-----
```

```
  \  
  \  
    ,-----.  
    , '_/_/_\_'  
    /<<::8[0]::>\  
    |-----|  
    | |====--| |  
    | |--=====| |  
    \ |:::|(O)|| /  
    | |....|(O)||  
    | |-----| |  
    | | \_____/ | |  
    /  \  /  \  \  
    '---' '---' '---'
```

```
-----  
( Today is 2017/09/25 )  
-----
```

```
  \  
  \  
    .\|//|\\|  
    \  |/\|/|/|/|  
    /  '|\|/|/|/|  
    o__ _|//|/|\\|'
```

# Contents

1	Macros	1
2	Options	2
2.1	Package Options	2
3	Defects	2
4	Dependencies	3
5	Available Animals	3
6	Miscellaneous	6

## 1 Macros

The following macros are available:

`\ducksay[<options>]{<message>}`

options might include any of the options described in section 2. Prints an `<animal>` saying `<message>`. `<message>` is not read in verbatim. Multi-line `<message>`s are possible using `\\`. `\\` should not be inside a macro but at toplevel. Else use the option `ht`.

`\duckthink[<options>]{<message>}`

options might include any of the options described in section 2. Prints an `<animal>` thinking `<message>`. `<message>` is not read in verbatim. It is implemented using regular expressions replacing a `\` which is only preceded by `\s*` in the first three lines with `0` and `o`. It is therefore slower than `\ducksay`. Multi-line `<message>`s are possible using `\\`. `\\` should not be inside a macro but at toplevel. Else use the option `ht`.

`\DefaultAnimal{<animal>}`

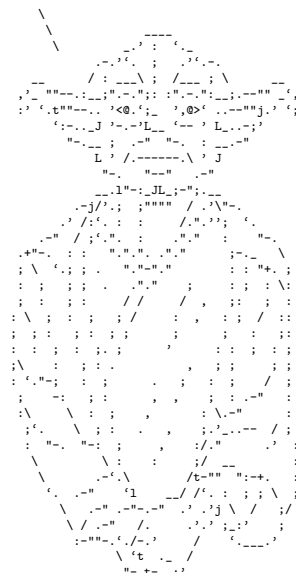
use the `<animal>` if none is given in the optional argument to `\ducksay` or `\duckthink`. Package default is `duck`. You might specify more options than `<animal>` with this, but it should contain an `<animal>`, otherwise you have to specify an `<animal>` in each call of `\ducksay`.

`\AddAnimal(*){<animal>}<ascii-art>`

adds `<animal>` to the known animals. `<ascii-art>` is multi-line verbatim and therefore should be delimited either by matching braces or by anything that works for `\verb`. If the star is given `<animal>` is the new default. One space is added to the begin of `<animal>` (compensating the opening symbol). For example, snowman is added with:

```
\AddAnimal{snowman}
{ \
  \[_]_
  (")
  >-( : )-<
  ( : : )}
```

( Use those, you might )



## 2 Options

<animal>

bubble=#1

body=#1

align=#1

msg-align=#1

wd=#1

ht=#1

## 2.1 Package Options

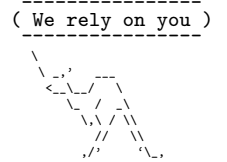
animal=#1

### 3 Defects

- no automatic line wrapping

## 4 Dependencies

The package depends on the two packages `xparse` and `l3keys2e` and all of their dependencies.



## 5 Available Animals

The following animals are provided by this package. I did not create them (but altered some), they belong to their original creators.

```
-----
( duck )
  \
  \ >(')
    )/
    /(
    / '----/
    \ ~=- /
-----
```

```
-----
( small-duck )
  \
  \ >()_
    ( ) _ _
-----
```

```
-----
( duck-family )
  \
  \ >(')
    )/
    /(
    / '----/ -()_ >()_
    \ ~=- / _ _ ( ) _ _ _
-----
```

```
-----
( cow )
  \ ^ _ ^
  \ (oo)\-----
    ( ) \ ) \ \
        ||----w |
        ||
-----
```

```
-----
( tux )
  \
  \ .-.
    |o_o |
    |__| |
    // \ / \
    (| | | )
    /'\_ _/ '\
    \___)=(___/
-----
```

```
-----
( pig )
  \ _//| .-~~-.
  \ _/oo } }-@
    ('')_ } |
    '---| { }--{ }
          // _ / _ /
-----
```

```
-----
( frog )
  \
  \ (.)_(.)
  \ ( _ _ ) _
  / \/'-----'\ \
  _ \ ( ( ) ) / _
  ) \ / \ _ . / \ (
  ) _ / / \ \ / \ \ _ (
-----
```

```
-----
( snowman )
  \
  \ _[ ]_
  \ (")
  >-( : )-<
  ( _ : _ )
-----
```

```
-----
( head-in )
  \ ^ _ ^
  \ (oo)\-----/
    ( ) \ )=( _ _ | _ \
        ||----w | \ \ \ _ _ _ |
        ||      ||
-----
```

```
-----
( sodomized )
  \
  \ ^ _ ^
  \ (oo)\-----/ \ \
    ( ) \ ) / \
        ||----w ( (
        ||      ||>>
-----
```

```

\      .\|//| \| |.
\    |/\|/|/|//|/|
/.  ' |/\|/|/|/|
o_--_|//|/| \| |'

```

\\\_//  
(')---.  
\_/\_-( )o

\ .-\' \ / \  
 " \ , -----  
 \_\_\_\_ / ( , ' \_\_\_\_  
 , ----- , uuuu , ----- , uuuuu ,

\ , ; ; ; ;  
 ; ; ; ;  
 . = ' , ; ; ; ;  
 / \_ ' " = . ' ; ; ; ;  
 @ = : \_ , \ ; ; ; ;  
 \_ ( \ . = ; ; ; ;  
 ' " ( \_ / = " '  
 € " ; ; €

The drawing is a dense, abstract composition of various line styles and symbols. It features a mix of solid, dashed, dotted, and wavy lines, along with numerous slashes, parentheses, braces, and other punctuation marks. The lines are arranged in a complex, overlapping pattern that suggests a sense of depth and movement. The overall effect is that of a highly stylized, perhaps digital or algorithmic, sketch. The drawing is primarily black on a white background, with some lines extending towards the right edge.

$\backslash$   
 $\backslash$   
 $\backslash$   $\wedge$   
 $( )$   
 $. ( o ) .$

( r2d2 )

```
\
 \,-----.
 \,/_/_/_\_.
 /<<::8[0]:>\
 |-----|
 | |=====| |
 | |-----| |
 \ |:::|O|| /
 | |...|O|| |
 | |-----| |
 | | \-----/ |
 / \ / \ / \
 '---' '---' '---
```

( yoda-head )

```
\
 \,-----.
 \,/_/_/_\_.
 /<<::8[0]:>\
 |-----|
 | |=====| |
 | |-----| |
 \ |:::|O|| /
 | |...|O|| |
 | |-----| |
 | | \-----/ |
 / \ / \ / \
 '---' '---' '---
```

( yoda )

```
\
 \,-----.
 \,/_/_/_\_.
 /<<::8[0]:>\
 |-----|
 | |=====| |
 | |-----| |
 \ |:::|O|| /
 | |...|O|| |
 | |-----| |
 | | \-----/ |
 / \ / \ / \
 '---' '---' '---
```

( vader )

```
\
 \,-----.
 \,/_/_/_\_.
 /<<::8[0]:>\
 |-----|
 | |=====| |
 | |-----| |
 \ |:::|O|| /
 | |...|O|| |
 | |-----| |
 | | \-----/ |
 / \ / \ / \
 '---' '---' '---
```

( small-yoda )

```
\
 \,-----.
 \,/_/_/_\_.
 /<<::8[0]:>\
 |-----|
 | |=====| |
 | |-----| |
 \ |:::|O|| /
 | |...|O|| |
 | |-----| |
 | | \-----/ |
 / \ / \ / \
 '---' '---' '---
```

## 6 Miscellaneous

This package is distributed under the terms of the GPLv3 or later, or the LPPL 1.3c or later, choose which ever license fits your needs the best.

The package is hosted on [https://github.com/Skillmon/ltx\\_ducksay](https://github.com/Skillmon/ltx_ducksay), you might report bugs there.

```
-----
( I'd choose WTFPL )
-----
\          ,:::,
  \          ::::,
   \="', "=: ::::,
    /_', "=: ::::,
     @=:_=: \, ::::,
      _(\,= ::::,
       _=(/_="
        e::e
```

-----  
( Who's gonna use it anyway? )  
-----

0  
o  
> ( ' )  
)/  
/(  
/ '-----/  
\\ ~=- /  
~~~~~

-----  
( hosted at [https://github.com/Skillmon/ltx\\_ducksay](https://github.com/Skillmon/ltx_ducksay) )  
-----

\\  
--.-.-  
'-\_- "7'  
/'.-c  
| /T  
\_)\_/LI