# The duckuments package

Jonathan P. Spratte*

version 0.4b, released 2018/09/13

## Contents



## 1 Introducktion

This package was inspired by the question getting ducks in example images. It began on the idea to patch `\includegraphics` to automatically change its behaviour if `example-image-duck` is used, but then it turned out to be a simple alternative to the blindtext package.

It is written as a docstrip file: executing `latex duckuments.dtx` generates the `duckuments.sty`, `example-image-duck.tex`, and `example-image-duck-portrait.tex` file and typesets this duckumentation; execute `tex duckuments.dtx` to only generate the files `duckuments.sty`, `example-image-duck.tex`, and `example-image-duck-portrait.tex`.

For its functionality `example-image-duck.tex` and `example-image-duck-portrait.tex` must be compiled at least once. The sources are hosted on github.

**The package does currently only work on pdfTeX, LuaTeX, and XeTeX.**

*E-mail: jspratte@yahoo.de

# 2 Duckumentation

## 2.1 Ducky content

`\duckument` `\duckument[⟨key=value⟩]`

Produces a duckument with one sectioning entry of each level starting at `\chapter` (if available) and two variants of the list environment `itemize`, `enumerate`, and `description`, one only at top level and one with 4 environments nested. The ⟨*key=value*⟩s accept every key as explained in subsection 2.4, but not every key has an effect.

`\blindduck` `\blindduck[⟨key=value⟩]`

Produces one paragraph of dummy content. The ⟨*key=value*⟩s accept every key as explained in subsection 2.4, but not every key has an effect.

`\ducklist` `\ducklist⟨*⟩{⟨environment⟩}`

Sets a list of the specified ⟨*environment*⟩, if ⟨*⟩ is given `\item[⟨dummy⟩]` is used instead of only `\item`. For `description` the starred version is used automatically.

`\ducklistlist` `\ducklistlist⟨*⟩{⟨list⟩}`

Sets 4 levels of a nested list of the specified ⟨*environment*⟩, if ⟨*⟩ is given `\item[⟨dummy⟩]` is used instead of only `\item`. For `description` the starred version is used automatically.

`\duckitemize` Abbreviation for `\ducklist{itemize}`.

`\duckenumerate` Abbreviation for `\ducklist{enumerate}`.

`\duckdescription` Abbreviation for `\ducklist{description}`.

## 2.2 Other Macros

`\duckumentsCreateExampleFiles`

Creates the files `example-image-duck.tex` and `example-image-duck-portrait.tex` in the current working directory.

`\duckumentsDrawRandomDucks` `\duckumentsDrawRandomDucks⟨*⟩[⟨count⟩]`

Draws ⟨*count*⟩ random tikzducks using Ti*k*Z. ⟨*count*⟩ defaults to `\duckuments@randoms`. Note that duckuments doesn't load Ti*k*Z, this macro is for the use in `example-image-duck.tex` and `example-image-duck-portrait.tex`. If the optional star is given, the images will be drawn in portrait orientation else in landscape.

## 2.3 Patches

The package patches `\includegraphics` if graphicx is loaded at the time the patch is applied (see subsection 2.4, `immediate`). The patch changes the behaviour if the used image file is `example-image-duck`. If that is the case, a random page of that document is used. There shouldn't be any change in behaviour if other files are used.

The random page is chosen with `\int_random:nn` in pdfTeX and LuaTeX. If XeTeX is used, the package implements a RC4 pseudo-random generating algorithm which is seeded using the current time and jobname. The generator can produce only numbers between 1 and 256 and is biased if 256 is not a multiple of the page count of `example-image-duck.pdf`.

The patch is done so that one can use tikzducks ducks without the need of loading tikz in a minimal working duckument as example images.

## 2.4 Options

The package and commands which take a `[`⟨*key=value*⟩`]` accept the following options. Some of which only make sense as package options. The **`bold`** printed value is the one used if you don't specify a value. The *`italic`* printed value is the initial one.

`toc=`**`true`**`|`*`false`*
: If `true` the `\duckument` contains a ToC. During package load time the key doesn't throw errors but only warnings for values other than `true` and `false`. This is done for compatibility reasons with classes which take a non-boolean `toc` option (like those of KOMA-script).

`maths=`**`both`**`|inline|display|`*`none`*
: If `both` the `\blindduck` (which is also used by `\duckument`) contains both inline and displayed math. With `inline` and `display` the respective maths is activated. `none` disables both.

`full`
: This typesets the full range of `\blindduck`. Don't use this as a package option.

`all` synonym for `full`.

`immediate=`**`true`**`|`*`false`*
: If `true` `\includegraphics` is patched during package load time, else the patching is done `\AtBeginDocument`.

Additionally `\blindduck` and `\duckument` accept another key which must match one of the following patterns and doesn't get any value. Patterns:

⟨*num1*⟩
: The paragraph ⟨*num1*⟩ is printed.

⟨*num1-*⟩
: Like the above

⟨*-num2*⟩
: The paragraphs up to ⟨*num2*⟩ are printed.

⟨*num1-num2*⟩
: The paragraphs from ⟨*num1*⟩ up to ⟨*num2*⟩ are printed.

⟨-⟩ The next paragraph is printed.

If the used pattern includes a "-" a `\par` is used after each paragraph. If any pattern is used the next pattern that doesn't include a ⟨*num1*⟩ starts with the paragraph after the last used paragraph, cycling back to the first after the last available paragraph.

# 3 Implementation

⟨*pkg⟩

⟨@@=duckuments⟩

## 3.1 Check for possible problems

Check which engine is used.

```
^^A>>>
\bool_if:nF
  {
    \sys_if_engine_luatex_p:
    || \sys_if_engine_pdftex_p:
    || \sys_if_engine_xetex_p:
  }
  {%>>>
    \msg_new:nnnn { duckuments } { incompatible }
      {
        The~duckuments~package~is~currently~only~compatible~with~pdfTeX,~
        LuaTeX,~and~XeTeX!
      }
      {
        \ ~_____\\
        \ (Quack!~Sorry~for~that.~Quack!)\\
        \ ~"""""""""""""""""""""""""""""\\
        \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ \ ~\ ~\ ~\ ~\string\ \\
        \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ \ ~\ ~\ ~\ ~\ ~>()_\\
        \ \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ \ ~\ ~\ ~\ ~\ ~\ ~(__)__
      }
    \msg_error:nn { duckuments } { incompatible }
    \endinput
  }%<<<
```

Check whether `example-image-duck.pdf` exists.

```
^^A>>>
\file_if_exist:nF { example-image-duck.pdf }
  {%>>>
```

If the current `\jobname` doesn't match `example-image-duck` throw an error.

```
    \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
      {
        \msg_new:nnnn { duckuments } { missing~file }
          {
            The~file~`#1`~can't~be~found.~Make~sure~to~create~it
            \tl_if_empty:nF{#2}{~#2}.
          }
          { Sorry~for~the~inconvenience.~#3 }
        \msg_error:nnnnn { duckuments } { missing~file }
```

```
          { example-image-duck.pdf }
          {
            by~compiling~example-image-duck.tex~at~least~once
          }
          {
            If~you~don't~find~the~file~on~your~machine~you~can~use~
            `\duckumentsCreateExampleFiles`~in~your~document~to~produce~a~copy~
            in~the~current~working~directory.
          }
      }
  }%<<<
```

And do the same for `example-image-duck-portrait.pdf`.

```
^^A>>>
\file_if_exist:nF { example-image-duck-portrait.pdf }
  {%>>>
    \str_if_eq:VnF \c_sys_jobname_str { example-image-duck-portrait }
      {
        \msg_if_exist:nnF { duckuments } { missing~file }
          {
            \msg_new:nnnn { duckuments } { missing~file }
              {
                The~file~`#1`~can't~be~found.~Make~sure~to~create~it
                \tl_if_empty:nF{#2}{~#2}.
              }
              { Sorry~for~the~inconvenience.~#3 }
          }
        \msg_error:nnnnn { duckuments } { missing~file }
          { example-image-duck-portrait.pdf }
          {
            by~compiling~example-image-duck-portrait.tex~at~least~once
          }
          {
            If~you~don't~find~the~file~on~your~machine~you~can~use~
            `\duckumentsCreateExampleFiles`~in~your~document~to~produce~a~copy~
            in~the~current~working~directory.
          }
      }
  }%<<<
```

## 3.2 Variables

`\duckuments@randoms` Stores the number of random ducks in `example-image-duck.pdf`.

```
\newcommand*\duckuments@randoms{128}
```

(*End definition for* `\duckuments@randoms`. *This variable is documented on page* **??**.)

`\l_duckuments_immediate_bool` Stores whether the patch is to be done during package load time.

```
\bool_new:N \l_duckuments_immediate_bool
```

(*End definition for* `\l_duckuments_immediate_bool`. *This variable is documented on page* **??**.)

`\l_duckuments_toc_bool` Stores whether to display a ToC in `\duckument`.

```
\bool_new:N \l_duckuments_toc_bool
```

(*End definition for* `\l_duckuments_toc_bool`. *This variable is documented on page* **??**.)

`\l_duckuments_math_inline_bool` Stores whether to display inline math in `\blindduck`.

```
\bool_new:N \l_duckuments_math_inline_bool
```

(*End definition for* `\l_duckuments_math_inline_bool`. *This variable is documented on page* **??**.)

`\l_duckuments_math_display_bool` Stores whether to display displayed math in `\blindduck`.

```
\bool_new:N \l_duckuments_math_display_bool
```

(*End definition for* `\l_duckuments_math_display_bool`. *This variable is documented on page* **??**.)

`\l_duckuments_blindduck_pars_bool` Stores whether each paragraph of `\blindduck` should end with a `\par`.

```
\bool_new:N \l_duckuments_blindduck_pars_bool
```

(*End definition for* `\l_duckuments_blindduck_pars_bool`. *This variable is documented on page* **??**.)

`\l_duckuments_range_seq` Stores the paragraphs range for `\blindduck`.

```
\seq_new:N \l_duckuments_range_seq
```

(*End definition for* `\l_duckuments_range_seq`. *This variable is documented on page* **??**.)

`\g_duckuments_blindduck_start_int` Stores the paragraph with which `\blindduck` should start.

```
\int_new:N \g_duckuments_blindduck_start_int
\int_gset:Nn \g_duckuments_blindduck_start_int { \c_one }
```

(*End definition for* `\g_duckuments_blindduck_start_int`. *This variable is documented on page* **??**.)

`\g_duckuments_blindduck_end_int` Stores the paragraph with which `\blindduck` should end.

```
\int_new:N \g_duckuments_blindduck_end_int
```

(*End definition for* `\g_duckuments_blindduck_end_int`. *This variable is documented on page* **??**.)

## 3.3 Constants

`\c_duckuments_example_regex` Regex against which the patch of `\includegraphics` is testing.

```
\regex_const:Nn \c_duckuments_example_regex
  {
    example-image-duck
    |example-image-duck.pdf
    |example-image-duck-portrait
    |example-image-duck-portrait.pdf
  }
```

(*End definition for* `\c_duckuments_example_regex`. *This variable is documented on page* **??**.)

`\c_duckuments_range_regex` Regex against which the optional range in `\blindduck` is checked.

```
\regex_const:Nn \c_duckuments_range_regex { (\d+|\d+-|-\d+|\d+-\d+|-) }
```

(*End definition for* `\c_duckuments_range_regex`. *This variable is documented on page* **??**.)

`\c_duckuments_blindduck_pars_int`

```
\int_const:Nn \c_duckuments_blindduck_pars_int { \c_five }
```

(*End definition for* `\c_duckuments_blindduck_pars_int`. *This variable is documented on page* **??**.)

\c_duckuments_example_pages_int

```
\bool_if:nTF
  {
    \str_if_eq_p:Vn \c_sys_jobname_str { example-image-duck }
    || \str_if_eq_p:Vn \c_sys_jobname_str { example-image-duck-portrait }
  }
  { \int_const:Nn \c_duckuments_example_pages_int { 1 } }
  {
    \group_begin:
      \sys_if_engine_pdftex:T
        {
          \pdfximage{example-image-duck.pdf}
          \int_set:Nn \l_tmpa_int { \the\pdflastximagepages }
          \pdfximage{example-image-duck-portrait.pdf}
          \int_set:Nn \l_tmpb_int { \the\pdflastximagepages }
        }
      \sys_if_engine_luatex:T
        {
          \saveimageresource{example-image-duck.pdf}
          \int_set:Nn \l_tmpa_int { \lastsavedimageresourcepages }
          \saveimageresource{example-image-duck-portrait.pdf}
          \int_set:Nn \l_tmpb_int { \lastsavedimageresourcepages }
        }
      \sys_if_engine_xetex:T
        {
          \int_set:Nn \l_tmpa_int
            { \XeTeXpdfpagecount "example-image-duck.pdf" }
          \int_set:Nn \l_tmpb_int
            { \XeTeXpdfpagecount "example-image-duck-portrait.pdf" }
        }
      \int_const:Nn \c_duckuments_example_pages_int
        { \int_min:nn { \l_tmpa_int } { \l_tmpb_int } }
    \group_end:
  }
```

(*End definition for* `\c_duckuments_example_pages_int`. *This variable is documented on page* **??**.)

## 3.4 Messages

```
^^A duckuments option~unknown >>>
\msg_new:nnnn { duckuments } { option~unknown }
  {
    Unknown~option~'#1'~for~package~duckuments.
  }
  {
    \ ~_____________________________________\\
    \ (Quack!~Nothing~here,~sorry.~Quack!)\\
    \ ~"""""""""""""""""""""""""""""""""""\\
    \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\string\ \\
    \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~>()_\\
    \ \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~(__)__
  }
^^A duckuments non-boolean~toc >>>
\msg_new:nnnn { duckuments } { non-boolean~toc }
```

```
  {
    A~non-boolean~value~`#1`~was~received~for~the~`toc`~option~during~package~
    load~time.~It~is~assumed~this~was~given~to~\string\documentclass\ and~
    therefore~ignored.
  }
  {
    \ ~_____________________________________\\
    \ (Quack!~Nothing~here,~sorry.~Quack!)\\
    \ ~"""""""""""""""""""""""""""""""""""""\\
    \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\string\ \\
    \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~>()_\\
    \ \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~(__)__
  }
^^A duckuments out~of~range >>>
\msg_new:nnnn { duckuments } { out~of~range }
  {
    You~requested~element~#3~out~of~the~range~#1~to~#2~of~array~'#4'.\\
    I'll~just~use~element~#1~for~you.
  }
  {
    \ ~_____________________________________\\
    \ (Quack!~Nothing~here,~sorry.~Quack!)\\
    \ ~"""""""""""""""""""""""""""""""""""""\\
    \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\string\ \\
    \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~>()_\\
    \ \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~(__)__
  }
```

\duckuments_patch_see_duckumentation:

```
\cs_new:Npn \duckuments_patch_see_duckumentation:
  {%>>>
    \cs_set:Nn \msg_see_documentation_text:n
      {
        \\\\
        See~the~\str_if_eq:nnTF { ##1 } { duckuments } { duckumentation }
          {
            \str_if_eq:nnTF { ##1 } { LaTeX } { LaTeX3~ } { ##1~ }
            documentation
          }~for~further~information.
      }
  }%<<<
\duckuments_patch_see_duckumentation:
```

(*End definition for* `\duckuments_patch_see_duckumentation:`. *This function is documented on page* **??**.)

## 3.5 Options and Configurations

```
\keys_define:nn { duckuments }
  {%>>>
    ,immediate .bool_set:N = \l_duckuments_immediate_bool
    ,immediate .default:n = true
    ,full  .code:n =
      \duckuments_blindduck_range_test:n { 1-\c_duckuments_blindduck_pars_int }
    ,all   .meta:n = { full }
```

```
    ,maths .choice:
    ,maths / both    .code:n =
      {
        \bool_set_true:N \l_duckuments_math_inline_bool
        \bool_set_true:N \l_duckuments_math_display_bool
      }
    ,maths / display .code:n = \bool_set_true:N \l_duckuments_math_display_bool
    ,maths / inline  .code:n = \bool_set_true:N \l_duckuments_math_inline_bool
    ,maths / none    .code:n =
      {
        \bool_set_false:N \l_duckuments_math_inline_bool
        \bool_set_false:N \l_duckuments_math_display_bool
      }
    ,maths .default:n = both
    ,toc   .code:n =
      {
        \str_case:nnF { #1 }
          {
            { true  } { \bool_set_true:N  \l_duckuments_toc_bool }
            { false } { \bool_set_false:N \l_duckuments_toc_bool }
          }
          {
            \msg_warning:nnn { duckuments } { non-boolean~toc } { #1 }
          }
      }
    ,toc     .default:n = true
    ,unknown .code:n =
      { \msg_error:nnx { duckuments } { option~unknown } { \l_keys_key_tl } }
  }%<<<
\ProcessKeysOptions { duckuments }
\keys_define:nn { duckuments }
  {%>>>
    ,toc     .undefine:
    ,toc     .bool_set:N = \l_duckuments_toc_bool
    ,toc     .default:n = true
    ,unknown .code:n = \duckuments_blindduck_range_test:V \l_keys_key_tl
  }%<<<
\bool_if:NTF \l_duckuments_immediate_bool
  { \AtEndOfPackage { \duckuments_patch_includegraphics: } }
  { \AtBeginDocument { \duckuments_patch_includegraphics: } }
```

## 3.6 Functions

### 3.6.1 Duckument Level

\duckument

```
\NewDocumentCommand \duckument { O{} }
  {%>>>
    \group_begin:
    \keys_set:nn { duckuments } { #1 }
    \bool_if:NT \l_duckuments_toc_bool { \tableofcontents }
    \cs_if_exist_use:NT \chapter
      { {\duckuments@headings@text{0}} \blindduck }
    \duckuments@headings{1} \blindduck
    \duckuments@headings{2} \blindduck
```

```
    \duckuments@headings{3} \blindduck
    \duckuments@headings{4} \blindduck
    \section {Lists}
    \duckuments_list_example:n { itemize }
    \duckuments_list_example:n { enumerate }
    \duckuments_list_example:n { description }
    \group_end:
  }%<<<
```

(*End definition for* `\duckument`. *This function is documented on page* 2.)

`\blindduck`

```
\NewDocumentCommand \blindduck { O{} }
  {%>>>
    \group_begin:
    \keys_set:nn { duckuments } { #1 }
    \duckuments@blindduck@text
    \bool_if:NT \l_duckuments_blindduck_pars_bool { \par }
    \group_end:
  }%<<<
```

(*End definition for* `\blindduck`. *This function is documented on page* 2.)

`\ducklist`

```
\NewDocumentCommand \ducklist {s m}
  {%>>>
    \begin{#2}
      \IfBooleanTF { #1 }
        {\ducklists@content@starred}
        {
          \str_if_eq:nnTF { #2 } { description }
            \ducklists@content@starred
            \ducklists@content
        }
    \end{#2}
  }%<<<
```

(*End definition for* `\ducklist`. *This function is documented on page* 2.)

`\ducklistlist`

```
\NewDocumentCommand \ducklistlist { s m }
  {%>>>
    \IfBooleanTF { #1 }
      { \duckuments@listlist@starred { #2 } }
      {
        \str_if_eq:nnTF { #2 } { description }
          { \duckuments@listlist@starred { description } }
          { \duckuments@listlist{#2} }
      }
  }%<<<
```

(*End definition for* `\ducklistlist`. *This function is documented on page* 2.)

`\duckenumerate`

```
\newcommand*\duckenumerate{\ducklist{enumerate}}
```

(*End definition for* `\duckenumerate`. *This function is documented on page 2.*)

`\duckitemize`

```
\newcommand*\duckitemize{\ducklist{itemize}}
```

(*End definition for* `\duckitemize`. *This function is documented on page 2.*)

`\duckdescription`

```
\newcommand*\duckdescription{\ducklist{description}}
```

(*End definition for* `\duckdescription`. *This function is documented on page 2.*)

`\duckumentsCreateExampleFiles` Create example-image-duck.tex

```
\newcommand*\duckumentsCreateExampleFiles
  {%>>>
    \iow_new:N \duckuments_example_file_iow
    \iow_open:Nn \duckuments_example_file_iow { example-image-duck.tex }
    \iow_now:Nn \duckuments_example_file_iow
      { \documentclass[tikz,multi]{standalone} }
    \iow_now:Nn \duckuments_example_file_iow
      { \usepackage{tikzducks} }
    \iow_now:Nn \duckuments_example_file_iow
      { \usepackage{duckuments} }
    \iow_now:Nn \duckuments_example_file_iow
      { \begin{document} }
    \iow_now:Nn \duckuments_example_file_iow
      { \duckumentsDrawRandomDucks }
    \iow_now:Nn \duckuments_example_file_iow
      { \end{document} }
    \iow_close:N \duckuments_example_file_iow
```

Create example-image-duck-portrait.tex

```
    \iow_open:Nn \duckuments_example_file_iow
      { example-image-duck-portrait.tex }
    \iow_now:Nn \duckuments_example_file_iow
      { \documentclass[tikz,multi]{standalone} }
    \iow_now:Nn \duckuments_example_file_iow
      { \usepackage{tikzducks} }
    \iow_now:Nn \duckuments_example_file_iow
      { \usepackage{duckuments} }
    \iow_now:Nn \duckuments_example_file_iow
      { \begin{document} }
    \iow_now:Nn \duckuments_example_file_iow
      { \duckumentsDrawRandomDucks* }
    \iow_now:Nn \duckuments_example_file_iow
      { \end{document} }
    \iow_close:N \duckuments_example_file_iow
  }%<<<
```

(*End definition for* `\duckumentsCreateExampleFiles`. *This function is documented on page 2.*)

`\duckumentsDrawRandomDucks`

```
\newcommand*\duckumentsDrawRandomDucks
  {%>>>
    \@ifstar
```

```
      {\duckumentsDrawRandomDucks@portrait}
      {\duckumentsDrawRandomDucks@landscape}%
  }%<<<
```

(*End definition for* `\duckumentsDrawRandomDucks`. *This function is documented on page* 2.)

### 3.6.2 Intern

`\duckuments@headings`

```
\newcommand*\duckuments@headings[1]
  {%>>>
    \ifcase#1\relax
      \expandafter\chapter
    \or \expandafter\section
    \or \expandafter\subsection
    \or \expandafter\subsubsection
    \or \expandafter\paragraph
    \else \expandafter\@gobble
    \fi
    {\duckuments@headings@text{#1}}
  }%<<<
```

(*End definition for* `\duckuments@headings`. *This function is documented on page* ??.)

`\duckuments@headings@level`

```
\newcommand*\duckuments@headings@level[1]
  {%>>>
    (
    \ifcase#1
      chapter
    \or section
    \or subsection
    \or subsubsection
    \or paragraph
    \fi
    )
  }%<<<
```

(*End definition for* `\duckuments@headings@level`. *This function is documented on page* ??.)

`\duckuments@ifinline`

```
\newcommand*\duckuments@ifinline[2][]
  { \bool_if:NTF \l_duckuments_math_inline_bool { #2 } { #1 } }
```

(*End definition for* `\duckuments@ifinline`. *This function is documented on page* ??.)

`\duckuments@ifdisplay`

```
\newcommand*\duckuments@ifdisplay[2][]
  { \bool_if:NTF \l_duckuments_math_display_bool { #2 } { #1 } }
```

(*End definition for* `\duckuments@ifdisplay`. *This function is documented on page* ??.)

\duckuments_list_example:n

```
\cs_new_protected_nopar:Npn \duckuments_list_example:n #1
  {%>>>
    \subsection{Example\ for\ ducks\ (#1)}
    \ducklist { #1 }
    \subsubsection{Nested\ ducks}
    \ducklistlist { #1 }
  }%<<<
```

(*End definition for* `\duckuments_list_example:n`. *This function is documented on page* **??**.)

\duckuments@enquote

```
\NewDocumentCommand \duckuments@enquote { s +m }
  {%>>>
    \IfBooleanTF { #1 }
      {
        \cs_if_exist_use:NTF
          \enquote { * { #2 } }
          { ‘#2’ }
      }
      {
        \cs_if_exist_use:NTF
          \enquote { { #2 } }
          {‘‘#2’’}
      }
  }%<<<
```

(*End definition for* `\duckuments@enquote`. *This function is documented on page* **??**.)

\duckumentsDrawRandomDucks@landscape

```
\newcommand*\duckumentsDrawRandomDucks@landscape[1][\duckuments@randoms]
  {%>>>
    \foreach\x in {1,2,...,#1}
      {\duckumentsDrawRandomDucks@draw{6.47214}{4}{3.23607}{2}}
  }%<<<
```

(*End definition for* `\duckumentsDrawRandomDucks@landscape`. *This function is documented on page* **??**.)

\duckumentsDrawRandomDucks@portrait

```
\newcommand*\duckumentsDrawRandomDucks@portrait[1][\duckuments@randoms]
  {%>>>
    \foreach\x in {1,2,...,#1}
      {\duckumentsDrawRandomDucks@draw{1.8541}{3}{.92705}{1.5}}
  }%<<<
```

(*End definition for* `\duckumentsDrawRandomDucks@portrait`. *This function is documented on page* **??**.)

\duckumentsDrawRandomDucks@draw

```
\newcommand*\duckumentsDrawRandomDucks@draw[4]
  {%>>>
    \begin{tikzpicture}
      \draw[fill=gray!50,use~as~bounding~box] (0,0) rectangle (#1,#2);
      \draw[gray,thin] (0,0) -- (#1,#2);
      \draw[gray,thin] (0,#2) -- (#1,0);
      \draw[gray,thin] (#3,0) -- (#3,#2);
```

```
      \draw[gray,thin] (0,#4) -- (#1,#4);
      \node at (#3,#4) {\tikz\randuck;};
      \draw[black] (0,0) rectangle (#1,#2);
    \end{tikzpicture}
  }%<<<
```

(*End definition for* `\duckumentsDrawRandomDucks@draw`. *This function is documented on page* **??**.)

`\duckuments_patch_includegraphics:`

```
\cs_new_protected_nopar:Npn \duckuments_patch_includegraphics:
  {%>>>
    \@ifpackageloaded { graphicx }
      {
        \LetLtxMacro\duckuments@includegraphicsBAK\includegraphics
        \RenewDocumentCommand \includegraphics
          { >{\duckuments_starred:n}s O{} o m }
          {
            \IfValueTF { ##3 }
              { \duckuments@includegraphicsBAK##1[##2][##3]{##4} }
              {
                \regex_match:NnTF \c_duckuments_example_regex { ##4 }
                  {
                    \duckuments_get_random_page:
                    \duckuments@includegraphicsBAK##1
                      [page=\duckuments_random_page:,##2]
                      { ##4 }
                  }
                  {
                    \duckuments@includegraphicsBAK##1[##2]{##4}
                  }
              }
          }
      }
      {}
  }%<<<
```

(*End definition for* `\duckuments_patch_includegraphics:`. *This function is documented on page* **??**.)

`\duckuments_blindduck_range_test:n`

```
\cs_new_protected:Npn \duckuments_blindduck_range_test:n #1
  {%>>>
    \regex_match:NnTF \c_duckuments_range_regex { #1 }
      {
        \seq_set_split:Nnn \l_duckuments_range_seq { - } { #1 }
        \int_compare:nNnTF { 1 } = { \seq_count:N \l_duckuments_range_seq }
          {
            \cs_set:Npn \duckuments@blindduck@text
              {
                \duckuments_blindduck_single_par:n { #1 }
                \duckuments_blindduck_set_next_start:n { #1 }
              }
          }
          {
            \bool_set_true:N \l_duckuments_blindduck_pars_bool
            \exp_args:Nx
```

```
            \tl_if_empty:nF { \seq_item:Nn \l_duckuments_range_seq { \c_one } }
              {
                \int_gset:Nn \g_duckuments_blindduck_start_int
                  { \seq_item:Nn \l_duckuments_range_seq { \c_one } }
              }
            \exp_args:Nx
            \tl_if_empty:nTF { \seq_item:Nn \l_duckuments_range_seq { \c_two } }
              {
                \int_gset_eq:NN
                  \g_duckuments_blindduck_end_int
                  \g_duckuments_blindduck_start_int
              }
              {
                \int_set:Nn \g_duckuments_blindduck_end_int
                  { \seq_item:Nn \l_duckuments_range_seq { \c_two } }
              }
            \duckuments_blindduck_set_text:xx
              { \int_use:N \g_duckuments_blindduck_start_int }
              { \int_use:N \g_duckuments_blindduck_end_int }
          }
      }
      {
        \exp_args:NnnV
          \msg_error:nnn { duckuments } { option~unknown } \l_keys_key_tl
      }
  }%<<<
\cs_generate_variant:Nn \duckuments_blindduck_range_test:n { V }
```

(*End definition for* `\duckuments_blindduck_range_test:n`*. This function is documented on page* **??**.)

`\duckuments_blindduck_set_text:nn`

```
\cs_new:Npn \duckuments_blindduck_set_text:nn #1 #2
  {%>>>
    \def \duckuments@blindduck@text
      {
        \int_step_function:nnnN { #1 } { \c_one } { #2 }
          \duckuments_blindduck_par_loop:n
        \duckuments_blindduck_set_next_start:n { #2 }
      }
  }%<<<
\cs_generate_variant:Nn \duckuments_blindduck_set_text:nn { xx }
```

(*End definition for* `\duckuments_blindduck_set_text:nn`*. This function is documented on page* **??**.)

`\duckuments_blindduck_set_next_start:n`

```
\cs_new:Npn \duckuments_blindduck_set_next_start:n #1
  {%>>>
    \int_gset:Nn \g_duckuments_blindduck_start_int
      { \int_mod:nn { #1 } { \c_duckuments_blindduck_pars_int } + \c_one }
  }%<<<
```

(*End definition for* `\duckuments_blindduck_set_next_start:n`*. This function is documented on page* **??**.)

\duckuments_blindduck_single_par:n

```
\cs_new:Npn \duckuments_blindduck_single_par:n #1
  {%>>>
    \bool_if:nTF
      {
        \int_compare_p:nNn { #1 } > { \c_duckuments_blindduck_pars_int }
        || \int_compare_p:nNn { #1 } < { \c_one }
      }
      {
        \msg_error:nnxxxx { duckuments } { out~of~range }
          { 1 } { \int_use:N \c_duckuments_blindduck_pars_int } { #1 }
          { blindduck~paragraphs }
        \duckuments@blindduck@text@i
      }
      {
        \use:c { duckuments@blindduck@text@ \int_to_roman:n { #1 } }
      }
  }%<<<
```

(*End definition for* `\duckuments_blindduck_single_par:n`. *This function is documented on page* **??**.)

\duckuments_blindduck_par_loop:n

```
\cs_new:Npn \duckuments_blindduck_par_loop:n #1
  {%>>>
    \duckuments_blindduck_single_par:n { #1 }
    \par
  }%<<<
```

(*End definition for* `\duckuments_blindduck_par_loop:n`. *This function is documented on page* **??**.)

\duckuments_starred:n

```
\cs_new_protected:Npn \duckuments_starred:n #1
  {%>>>
    \IfBooleanTF { #1 }
      { \def\ProcessedArgument{*} }
      { \def\ProcessedArgument{} }
  }%<<<
```

(*End definition for* `\duckuments_starred:n`. *This function is documented on page* **??**.)

\duckuments_get_random_page:
\duckuments_random_page:

```
\sys_if_engine_xetex:TF
  {
```

For XeTeX we need a bit more code in order to get random numbers. The following is an implementation of RC4. First declare some variables:

```
    \int_new:N \g_duckuments_RCiv_i_int
    \int_new:N \g_duckuments_RCiv_j_int
    \int_new:N \g_duckuments_RCiv_keylength_int
    \int_new:N \g_duckuments_tmpa_int
    \int_const:Nn \c_duckuments_RCiv_Slength_int { 256 }
    \tl_new:N \l_duckuments_tmpa_tl
    \tl_new:N \l_duckuments_tmpb_tl
```

Initialize the S array:

```
\cs_new_protected_nopar:Npn \duckuments_RCiv_S_new:n #1
  { \int_new:c { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int } }
\cs_new_protected_nopar:Npn \duckuments_RCiv_S_set:nn #1 #2
  { \int_gset:cn { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int } { #2 } }
\cs_new_nopar:Npn \duckuments_RCiv_S_get:n #1
  { \int_use:c { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int } }
\cs_new_protected_nopar:Npn \duckuments_RCiv_key_new:n #1
  { \int_new:c { g_duckuments_RCiv_key_ \int_eval:n { #1 } _int } }
\cs_new_protected_nopar:Npn \duckuments_RCiv_key_set:nn #1 #2
  { \int_gset:cn { g_duckuments_RCiv_key_ \int_eval:n { #1 } _int } { #2 } }
\cs_new_nopar:Npn \duckuments_RCiv_key_get:n #1
  { \int_use:c { g_duckuments_RCiv_key_ \int_eval:n { #1 } _int } }
\int_step_inline:nnnn { 0 } { 1 } { 255 }
  {
    \duckuments_RCiv_S_new:n { #1 }
    \duckuments_RCiv_S_set:nn { #1 } { #1 }
  }
\int_step_inline:nnnn { 0 } { 1 } { 4 }
  { \duckuments_RCiv_key_new:n { #1 } }
\duckuments_RCiv_key_set:nn { 0 } { \c_sys_minute_int }
\duckuments_RCiv_key_set:nn { 1 } { \c_sys_hour_int }
\duckuments_RCiv_key_set:nn { 2 } { \c_sys_day_int }
\duckuments_RCiv_key_set:nn { 3 } { \c_sys_month_int }
\duckuments_RCiv_key_set:nn { 4 }
  { \int_mod:nn { \c_sys_year_int } { \c_duckuments_RCiv_Slength_int } }
\int_gset:Nn \g_duckuments_RCiv_keylength_int { 5 }
\str_map_inline:Nn \c_sys_jobname_str
  {
    \duckuments_RCiv_key_new:n { \g_duckuments_RCiv_keylength_int }
    \duckuments_RCiv_key_set:nn
      { \g_duckuments_RCiv_keylength_int }
      { \int_from_alph:n { #1 } }
    \int_gincr:N \g_duckuments_RCiv_keylength_int
  }
\cs_new_protected_nopar:Npn \duckuments_swap_S_entries:nn #1 #2
  {
    \int_set_eq:Nc
      \g_duckuments_tmpa_int
      { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int }
    \int_set_eq:cc
      { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int }
      { g_duckuments_RCiv_S_ \int_eval:n { #2 } _int }
    \int_set_eq:cN
      { g_duckuments_RCiv_S_ \int_eval:n { #2 } _int }
      \g_duckuments_tmpa_int
  }
\int_gset:Nn \g_duckuments_RCiv_keylength_int { 5 }
\cs_new:Npn \duckuments_gadd_mod:Nnn #1 #2 #3
  { \int_gset:Nn #1 { \int_mod:nn { #1 + ( #2 ) } { #3 } } }
\cs_new:Npn \duckuments_gadd_mod_Slength:Nn #1 #2
  {
    \duckuments_gadd_mod:Nnn #1
      { #2 } { \c_duckuments_RCiv_Slength_int }
```

```
    }
  \int_step_inline:nnnn { 0 } { 1 } { 255 }
    {
      \int_gset:Nn \g_duckuments_tmpa_int
        { \int_mod:nn { #1 } { \g_duckuments_RCiv_keylength_int } }
      \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_j_int
        {
          \duckuments_RCiv_S_get:n { #1 }
          + \duckuments_RCiv_key_get:n { \g_duckuments_tmpa_int }
        }
      \duckuments_swap_S_entries:nn { #1 } { \g_duckuments_RCiv_j_int }
    }
  \int_gzero:N \g_duckuments_RCiv_i_int
  \int_gzero:N \g_duckuments_RCiv_j_int
```

Provide a function which gets the next random number and sets `\duckuments_random_page:` to it.

```
  \cs_new_protected_nopar:Npn \duckuments_get_random_page:
    {
      \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_i_int { \c_one }
      \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_j_int
        { \duckuments_RCiv_S_get:n { \g_duckuments_RCiv_i_int } }
      \duckuments_swap_S_entries:nn
        { \g_duckuments_RCiv_i_int }
        { \g_duckuments_RCiv_j_int }
      \int_gset:Nn \g_duckuments_tmpa_int
        { \duckuments_RCiv_S_get:n { \g_duckuments_RCiv_i_int } }
      \duckuments_gadd_mod_Slength:Nn \g_duckuments_tmpa_int
        { \duckuments_RCiv_S_get:n { \g_duckuments_RCiv_j_int } }
      \cs_set:Nx \duckuments_random_page:
        {
          \int_eval:n
            {
              \int_mod:nn
                { \duckuments_RCiv_S_get:n { \g_duckuments_tmpa_int } }
                { \c_duckuments_example_pages_int }
              + \c_one
            }
        }
    }
  \cs_new:Npn \duckuments_random_page: { 1 }
}
```

Both pdfTEX and LuaTEX don't need the RC4 as there `\int_random:nn` is available.

```
{
  \cs_new:Npn \duckuments_get_random_page: {}
  \cs_new:Npn \duckuments_random_page:
    { \int_rand:nn { 1 } { \c_duckuments_example_pages_int } }
}
```

(*End definition for* `\duckuments_get_random_page:` *and* `\duckuments_random_page:`*. These functions are documented on page* **??**.)

```
\ExplSyntaxOff
```

`\duckuments@blindduck@text`

```
\newcommand*\duckuments@blindduck@text{\duckuments@blindduck@text@i}
\newcommand*\duckuments@blindduck@text@i
  {%>>>
    There once was a very smart but sadly blind duck. When it was still a small
    duckling it was renowned for its good vision. But sadly as the duck grew
    older it caught a sickness which caused its eyesight to worsen. It became so
    bad, that the duck couldn't read the notes it once took containing much of
    inline math\duckuments@ifinline{ just like its favoured equation: $d = u_c
    \cdot k$}. Only displayed equations remained legible%
    \duckuments@ifdisplay[.]{ so it could still read \begin{equation}d = r a^k
    e\hbox{.}\end{equation}} That annoyed the smart duck, as it wasn't able to
    do its research any longer. It called for its underduckling and said:
    \duckuments@enquote{Go, find me the best eye ducktor there is. He shall
    heal me from my disease!}%
  }%<<<
\newcommand*\duckuments@blindduck@text@ii
  {%>>>
    \duckuments@enquote{But my duck, how are you supposed to manage your daily
    routines without my visual guidance}, replied the underduckling. The smart
    duck's face turned grim in anger. \duckuments@enquote{You dare to talk
    back?} The underduckling blushed ashamed. How could he have objections
    after his duck gave strict orders? The underduckling was so embarrassed
    about his own behaviour he had to solve an equation.%
  }%<<<
\newcommand*\duckuments@blindduck@text@iii
  {%>>>
    After the equation was solved and the underduckling prepared his leave for
    the next day it fell asleep in a shaky mood. It did not know what the
    journey had prepared for him and if he was prepared enough for it. His sleep
    was restless. The dreams he had that night were not calm and bright as they
    used to be for an innocent underduckling.%
  }%<<<
\newcommand*\duckuments@blindduck@text@iv
  {%>>>
    Before dawn the underduckling woke. He didn't have the feeling of being well
    rested. But nonetheless he knew that this was the day he should leave.
    Except saying goodbye to his beloved ones there was nothing holding him
    back. His duck had sent him on the most important mission a five weeks old
    inexperienced underduckling was ever sent on. He bid farewell to his mother,
    all his brothers and sisters, and finally from his duck. The bag was
    shouldered, the boots were tied, the underduckling left.%
  }%<<<
\newcommand*\duckuments@blindduck@text@v
  {%>>>
    As the underduckling went along he saw a tree. It was so tall he couldn't
    even approximate its height. Oh, how small he felt. What should he, a small
    duckling whose plumage was still fluffy, achieve in this vast, gigantic
    world? He stuck his trembling wings under the straps of his backpack to calm
    his shaking limbs. Where should he start his search? How should he survive
    in the hostile nature surrounding the sheltered pond of his community?
  }%<<<
```

(*End definition for* `\duckuments@blindduck@text`. *This function is documented on page* ??.)

\duckuments@headings@text

```
\newcommand*\duckuments@headings@text[1]
  {A friendly duck at level #1 \duckuments@headings@level{#1}}
```

(*End definition for* `\duckuments@headings@text`. *This function is documented on page* **??**.)

\ducklists@content

```
\newcommand*\ducklists@content
  {%>>>
    \item First swims father drake
    \item Then floats mother duck
    \item After her paddles baby duckling
    \item And over there bathes uncle canard
  }%<<<
```

(*End definition for* `\ducklists@content`. *This function is documented on page* **??**.)

\ducklists@content@starred

```
\newcommand*\ducklists@content@starred
  {%>>>
    \item[drake] is the swimming father
    \item[duck] is the floating mother
    \item[duckling] is the paddling baby
    \item[canard] is the bathing uncle
  }%<<<
```

(*End definition for* `\ducklists@content@starred`. *This function is documented on page* **??**.)

\duckuments@listlist

```
\newcommand*\duckuments@listlist[1]
  {%>>>
    \begin{#1}
      \item swimming father drake
        \begin{#1}
          \item swimming father drake
            \begin{#1}
              \item swimming father drake
                \begin{#1}
                  \item swimming father drake
                  \item floating mother duck
                \end{#1}
              \item floating mother duck
            \end{#1}
          \item floating mother duck
        \end{#1}
      \item floating mother duck
    \end{#1}%
  }%<<<
```

(*End definition for* `\duckuments@listlist`. *This function is documented on page* **??**.)

\duckuments@listlist@starred

```
\newcommand*\duckuments@listlist@starred[1]
  {%>>>
    \begin{#1}
      \item[drake] is the swimming father
        \begin{#1}
          \item[drake] is the swimming father
            \begin{#1}
              \item[drake] is the swimming father
                \begin{#1}
                  \item[drake] is the swimming father
                  \item[duck] is the floating mother
                \end{#1}
              \item[duck] is the floating mother
            \end{#1}
          \item[duck] is the floating mother
        \end{#1}
      \item[duck] is the floating mother
    \end{#1}%
  }%<<<
```

(*End definition for* `\duckuments@listlist@starred`. *This function is documented on page* **??**.)

```
\endinput
```

⟨/pkg⟩

# 4 The story of the duck

paragraph 1 There once was a very smart but sadly blind duck. When it was still a small duckling it was renowned for its good vision. But sadly as the duck grew older it caught a sickness which caused its eyesight to worsen. It became so bad, that the duck couldn't read the notes it once took containing much of inline math. Only displayed equations remained legible. That annoyed the smart duck, as it wasn't able to do its research any longer. It called for its underduckling and said: "Go, find me the best eye ducktor there is. He shall heal me from my disease!"

paragraph 2 "But my duck, how are you supposed to manage your daily routines without my visual guidance", replied the underduckling. The smart duck's face turned grim in anger. "You dare to talk back?" The underduckling blushed ashamed. How could he have objections after his duck gave strict orders? The underduckling was so embarrassed about his own behaviour he had to solve an equation.

paragraph 3 After the equation was solved and the underduckling prepared his leave for the next day it fell asleep in a shaky mood. It did not know what the journey had prepared for him and if he was prepared enough for it. His sleep was restless. The dreams he had that night were not calm and bright as they used to be for an innocent underduckling.

paragraph 4 Before dawn the underduckling woke. He didn't have the feeling of being well rested. But nonetheless he knew that this was the day he should leave. Except saying goodbye to his beloved ones there was nothing holding him back. His duck had sent him on the most important mission a five weeks old inexperienced underduckling was ever sent on. He bid farewell to his mother, all his brothers and sisters, and finally from his duck. The bag was shouldered, the boots were tied, the underduckling left.

paragraph 5 As the underduckling went along he saw a tree. It was so tall he couldn't even approximate its height. Oh, how small he felt. What should he, a small duckling whose plumage was still fluffy, achieve in this vast, gigantic world? He stuck his trembling wings under the straps of his backpack to calm his shaking limbs. Where should he start his search? How should he survive in the hostile nature surrounding the sheltered pond of his community?