

The duckuments package

Jonathan P. Spratte*

version 0.5, released 2019-10-03

Contents

1	Introductkion	1
2	Duckumentation	2
2.1	Ducky content	2
2.2	Other Macros	2
2.3	Patches	3
2.4	Options	3
3	Implementation	4
3.1	Check for possible problems	4
3.2	Variables	5
3.3	Constants	6
3.4	Messages	8
3.5	Options and Configurations	9
3.6	Functions	10
3.6.1	Duckument Level	10
3.6.2	Intern	12
4	The story of the duck	20

1 Introductkion

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](#).

*E-mail: jspratte@yahoo.de

The package should support any engine. However for engines other than pdfTeX, LuaTeX, and XeTeX it requires `extractbb` to be legal in a shell-escape (which should normally be the case).

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 `TikZ`. `⟨count⟩` defaults to `\duckuments@randoms`. Note that `duckuments` doesn't load `TikZ`, 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` or `example-image-duck-portrait`. 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 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

```
1 <*pkg>
2 <@@=duckuments>
```

3.1 Check for possible problems

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

```
3 ^^A>>=
4 \file_if_exist:nF { example-image-duck.pdf }
5 {%>>=
```

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

```
6   \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
7   {
8     \msg_new:nnnn { duckuments } { missing-file }
9     {
10      The~file~'#1'~can't~be~found.~Make~sure~to~create~it
11      \tl_if_empty:nF{#2}{~#2}.
12    }
13    { Sorry~for~the~inconvenience.~#3 }
14    \msg_error:nnnn { duckuments } { missing-file }
15    { example-image-duck.pdf }
16    {
17      by~compiling~example-image-duck.tex~at~least~once
18    }
19    {
20      If~you~don't~find~the~file~on~your~machine~you~can~use~
21      'duckumentsCreateExampleFiles'~in~your~document~to~produce~a~copy~
22      in~the~current~working~directory.
23    }
24  }
25 }%=<<
```

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

```
26 ^^A>>=
27 \file_if_exist:nF { example-image-duck-portrait.pdf }
28 {%>>=
29   \str_if_eq:VnF \c_sys_jobname_str { example-image-duck-portrait }
30   {
31     \msg_if_exist:nnF { duckuments } { missing-file }
32     {
33       \msg_new:nnnn { duckuments } { missing-file }
34       {
35         The~file~'#1'~can't~be~found.~Make~sure~to~create~it
36         \tl_if_empty:nF{#2}{~#2}.
37       }
38       { Sorry~for~the~inconvenience.~#3 }
39     }
40     \msg_error:nnnn { duckuments } { missing-file }
41     { example-image-duck-portrait.pdf }
42     {
43       by~compiling~example-image-duck-portrait.tex~at~least~once
44     }
45   }
```

```

46         If~you~don't~find~the~file~on~your~machine~you~can~use~
47         '\duckumentsCreateExampleFiles'~in~your~document~to~produce~a~copy~
48         in~the~current~working~directory.
49     }
50 }
51 }%=<<

```

3.2 Variables

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

```
52 \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.

```
53 \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`.

```
54 \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`.

```
55 \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`.

```
56 \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`.

```
57 \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`.

```
58 \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.

```
59 \int_new:N \g_duckuments_blindduck_start_int
```

```
60 \int_gset:Nn \g_duckuments_blindduck_start_int { \c_one_int }
```

(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.

```
61 \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.

```
62 \regex_const:Nn \c_duckuments_example_regex
63 {
64   example-image-duck
65   |example-image-duck.pdf
66   |example-image-duck-portrait
67   |example-image-duck-portrait.pdf
68 }
```

(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.

```
69 \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`

```
70 \int_const:Nn \c_duckuments_blindduck_pars_int { 5 }
```

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

`\c_duckuments_example_pages_int` Define a macro that gets the page count of a PDF (this is engine dependent):

```
71 \sys_if_engine_pdftex:T
72 {
73   \cs_new_protected:Npn \duckuments_set_to_pdf_page_count:Nn #1 #2
74   {
75     \group_begin:
76     \pdfximage { #2 }
77     \exp_args:NNNo
78     \group_end:
79     \int_set:Nn #1 { \the \pdfastximagepages }
80   }
81 }
82 \sys_if_engine_luatex:T
83 {
84   \cs_new_protected:Npn \duckuments_set_to_pdf_page_count:Nn #1 #2
85   {
86     \group_begin:
87     \saveimageresource { #2 }
88     \exp_args:NNNo
89     \group_end:
90     \int_set:Nn #1 { \lastsavedimageresourcepages }
91   }
92 }
93 \sys_if_engine_xetex:T
94 {
95   \cs_new_protected:Npn \duckuments_set_to_pdf_page_count:Nn #1 #2
96   {
97     \int_set:Nn #1 { \XeTeXpdfpagecount " #2 " }
98   }
99 }
100 \bool_lazy_any:nTF
101 {
```

```

102 { \sys_if_engine luatex_p: }
103 { \sys_if_engine pdftex_p: }
104 { \sys_if_engine xetex_p: }
105 }
106 { \if_false: }
107 { \if_true: }
108 \group_begin:
109 \char_set_catcode_other:N \%
110 \cs_new_protected:Npn \duckuments_set_to_pdf_page_count:Nn #1 #2
111 {
112   \group_begin:
113   \char_set_catcode_other:N \%
114   \char_set_catcode_letter:N \:
115   \char_set_catcode_ignore:n { 32 }
116   \ior_shell_open:Nn \g_tmpa_ior { extractbb ~ -0 ~ #2 }
117   \ior_map_inline:Nn \g_tmpa_ior
118   {
119     \__duckuments_set_to_pdf_page_count_aux_a:w ##1 %%Pages: \q_stop
120   }
121   \ior_close:N \g_tmpa_ior
122   \exp_args:NNNo
123   \group_end:
124   \int_set:Nn #1 { \int_use:N \l_tmpa_int }
125 }
126 \cs_new_protected:Npn \__duckuments_set_to_pdf_page_count_aux_a:w
127 #1 %%Pages: #2 \q_stop
128 {
129   \tl_if_empty:nT { #1 }
130   { \__duckuments_set_to_pdf_page_count_aux_b:w #2 \q_stop }
131 }
132 \cs_new_protected:Npn \__duckuments_set_to_pdf_page_count_aux_b:w
133 #1 %%Pages: \q_stop
134 {
135   \int_set:Nn \l_tmpa_int { #1 }
136   \ior_map_break:
137 }
138 \group_end:
139 \fi:

```

Setting the constant:

```

140 \bool_if:nTF
141 {
142   \str_if_eq_p:Vn \c_sys_jobname_str { example-image-duck }
143   || \str_if_eq_p:Vn \c_sys_jobname_str { example-image-duck-portrait }
144 }
145 { \int_const:Nn \c_duckuments_example_pages_int { \c_one_int } }
146 {
147   \group_begin:
148   \duckuments_set_to_pdf_page_count:Nn \l_tmpa_int
149   { example-image-duck.pdf }
150   \duckuments_set_to_pdf_page_count:Nn \l_tmpb_int
151   { example-image-duck-portrait.pdf }
152   \int_const:Nn \c_duckuments_example_pages_int
153   { \int_min:nn { \l_tmpa_int } { \l_tmpb_int } }
154   \group_end:

```



```

202     \\\
203     See~the~\str_if_eq:nnTF { ##1 } { duckuments } { duckumentation }
204     {
205         \str_if_eq:nnTF { ##1 } { LaTeX } { LaTeX3~ } { ##1~ }
206         documentation
207     }~for~further~information.
208 }
209 }%=<<
210 \duckuments_patch_see_duckumentation:

```

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

3.5 Options and Configurations

```

211 \keys_define:nn { duckuments }
212 {%>>=
213     ,immediate .bool_set:N = \l_duckuments_immediate_bool
214     ,immediate .default:n = true
215     ,full .code:n =
216         \duckuments_blindduck_range_test:n { 1-\c_duckuments_blindduck_pars_int }
217     ,all .meta:n = { full }
218     ,maths .choice:
219     ,maths / both .code:n =
220     {
221         \bool_set_true:N \l_duckuments_math_inline_bool
222         \bool_set_true:N \l_duckuments_math_display_bool
223     }
224     ,maths / display .code:n = \bool_set_true:N \l_duckuments_math_display_bool
225     ,maths / inline .code:n = \bool_set_true:N \l_duckuments_math_inline_bool
226     ,maths / none .code:n =
227     {
228         \bool_set_false:N \l_duckuments_math_inline_bool
229         \bool_set_false:N \l_duckuments_math_display_bool
230     }
231     ,maths .default:n = both
232     ,toc .code:n =
233     {
234         \str_case:nnF { #1 }
235         {
236             { true } { \bool_set_true:N \l_duckuments_toc_bool }
237             { false } { \bool_set_false:N \l_duckuments_toc_bool }
238         }
239         {
240             \msg_warning:nnn { duckuments } { non-boolean~toc } { #1 }
241         }
242     }
243     ,toc .default:n = true
244     ,unknown .code:n =
245     { \msg_error:nnx { duckuments } { option~unknown } { \l_keys_key_tl } }
246 }%=<<
247 \ProcessKeysOptions { duckuments }
248 \keys_define:nn { duckuments }
249 {%>>=
250     ,toc .undefine:

```

```

251     ,toc      .bool_set:N = \l_duckuments_toc_bool
252     ,toc      .default:n = true
253     ,unknown .code:n = \duckuments_blindduck_range_test:V \l_keys_key_tl
254   }%=<<
255 \bool_if:NTF \l_duckuments_immediate_bool
256 { \AtEndOfPackage { \duckuments_patch_includegraphics: } }
257 { \AtBeginDocument { \duckuments_patch_includegraphics: } }

```

3.6 Functions

3.6.1 Duckument Level

`\duckument`

```

258 \NewDocumentCommand \duckument { 0{ } }
259   {>>=
260     \group_begin:
261     \keys_set:nn { duckuments } { #1 }
262     \bool_if:NT \l_duckuments_toc_bool { \tableofcontents }
263     \cs_if_exist_use:NT \chapter
264       { {\duckuments@headings@text{0}} \blindduck }
265     \duckuments@headings{1} \blindduck
266     \duckuments@headings{2} \blindduck
267     \duckuments@headings{3} \blindduck
268     \duckuments@headings{4} \blindduck
269     \section {Lists}
270     \duckuments_list_example:n { itemize }
271     \duckuments_list_example:n { enumerate }
272     \duckuments_list_example:n { description }
273     \group_end:
274   }%=<<

```

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

`\blindduck`

```

275 \NewDocumentCommand \blindduck { 0{ } }
276   {>>=
277     \group_begin:
278     \keys_set:nn { duckuments } { #1 }
279     \duckuments@blindduck@text
280     \bool_if:NT \l_duckuments_blindduck_pars_bool { \par }
281     \group_end:
282   }%=<<

```

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

`\ducklist`

```

283 \NewDocumentCommand \ducklist {s m}
284   {>>=
285     \begin{#2}
286       \IfBooleanTF { #1 }
287         {\ducklists@content@starred}
288         {
289           \str_if_eq:nnTF { #2 } { description }
290             \ducklists@content@starred
291             \ducklists@content

```

```

292     }
293     \end{#2}
294 }%=<<

```

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

`\ducklistlist`

```

295 \NewDocumentCommand \ducklistlist { s m }
296   {%>>=
297   \IfBooleanTF { #1 }
298     { \duckuments@listlist@starred { #2 } }
299     {
300       \str_if_eq:nnTF { #2 } { description }
301         { \duckuments@listlist@starred { description } }
302         { \duckuments@listlist{#2} }
303     }
304   }%=<<

```

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

`\duckenumerate`

```

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

```

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

`\duckitemize`

```

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

```

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

`\duckdescription`

```

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

```

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

`\duckumentsCreateExampleFiles`

Create `example-image-duck.tex`

```

308 \newcommand*\duckumentsCreateExampleFiles
309   {%>>=
310   \iow_new:N \duckuments_example_file_iow
311   \iow_open:Nn \duckuments_example_file_iow { example-image-duck.tex }
312   \iow_now:Nn \duckuments_example_file_iow
313     { \documentclass[tikz,multi]{standalone} }
314   \iow_now:Nn \duckuments_example_file_iow
315     { \usepackage{tikzducks} }
316   \iow_now:Nn \duckuments_example_file_iow
317     { \usepackage{duckuments} }
318   \iow_now:Nn \duckuments_example_file_iow
319     { \begin{document} }
320   \iow_now:Nn \duckuments_example_file_iow
321     { \duckumentsDrawRandomDucks }
322   \iow_now:Nn \duckuments_example_file_iow
323     { \end{document} }
324   \iow_close:N \duckuments_example_file_iow

```

Create example-image-duck-portrait.tex

```
325 \iow_open:Nn \duckuments_example_file_iow
326 { example-image-duck-portrait.tex }
327 \iow_now:Nn \duckuments_example_file_iow
328 { \documentclass[tikz,multi]{standalone} }
329 \iow_now:Nn \duckuments_example_file_iow
330 { \usepackage{tikzducks} }
331 \iow_now:Nn \duckuments_example_file_iow
332 { \usepackage{duckuments} }
333 \iow_now:Nn \duckuments_example_file_iow
334 { \begin{document} }
335 \iow_now:Nn \duckuments_example_file_iow
336 { \duckumentsDrawRandomDucks* }
337 \iow_now:Nn \duckuments_example_file_iow
338 { \end{document} }
339 \iow_close:N \duckuments_example_file_iow
340 }%=<<
```

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

`\duckumentsDrawRandomDucks`

```
341 \newcommand*\duckumentsDrawRandomDucks
342 {>>=
343 \ifstar
344 {\duckumentsDrawRandomDucks@portrait}
345 {\duckumentsDrawRandomDucks@landscape}%
346 }%=<<
```

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

3.6.2 Intern

`\duckuments@headings`

```
347 \newcommand*\duckuments@headings[1]
348 {>>=
349 \ifcase#1\relax
350 \expandafter\chapter
351 \or \expandafter\section
352 \or \expandafter\subsection
353 \or \expandafter\subsubsection
354 \or \expandafter\paragraph
355 \else \expandafter\@gobble
356 \fi
357 {\duckuments@headings@text{#1}}
358 }%=<<
```

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

`\duckuments@headings@level`

```
359 \newcommand*\duckuments@headings@level[1]
360 {>>=
361 (
362 \ifcase#1
363 chapter
```

```

364     \or section
365     \or subsection
366     \or subsubsection
367     \or paragraph
368     \fi
369   )
370 }%=<<

```

(End definition for `\duckuments@headings@level1`. This function is documented on page ??.)

`\duckuments@ifinline`

```

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

```

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

`\duckuments@ifdisplay`

```

373 \newcommand*\duckuments@ifdisplay[2] []
374 { \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`

```

375 \cs_new_protected_nopar:Npn \duckuments_list_example:n #1
376 { %>>=
377   \subsection{Example\ for\ ducks\ (#1)}
378   \ducklist { #1 }
379   \subsubsection{Nested\ ducks}
380   \ducklistlist { #1 }
381 }%=<<

```

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

`\duckuments@enquote`

```

382 \NewDocumentCommand \duckuments@enquote { s +m }
383 { %>>=
384   \IfBooleanTF { #1 }
385   {
386     \cs_if_exist_use:NTF
387     \enquote { * { #2 } }
388     { '#2' }
389   }
390   {
391     \cs_if_exist_use:NTF
392     \enquote { { #2 } }
393     { {'#2'} }
394   }
395 }%=<<

```

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

`\duckumentsDrawRandomDucks@landscape`

```

396 \newcommand*\duckumentsDrawRandomDucks@landscape[1] [\duckuments@randoms]
397 { %>>=
398   \foreach\x in {1,2,...,#1}
399     {\duckumentsDrawRandomDucks@draw{6.47214}{4}{3.23607}{2}}
400 }%=<<

```

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

`\duckumentsDrawRandomDucks@portrait`

```
401 \newcommand*\duckumentsDrawRandomDucks@portrait[1][\duckuments@randoms]
402   {>>=
403   \foreach\x in {1,2,...,#1}
404     {\duckumentsDrawRandomDucks@draw{1.8541}{3}{.92705}{1.5}}
405   }<<=
```

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

`\duckumentsDrawRandomDucks@draw`

```
406 \newcommand*\duckumentsDrawRandomDucks@draw[4]
407   {>>=
408   \begin{tikzpicture}
409     \draw[fill=gray!50,use-as-bounding-box] (0,0) rectangle (#1,#2);
410     \draw[gray,thin] (0,0) -- (#1,#2);
411     \draw[gray,thin] (0,#2) -- (#1,0);
412     \draw[gray,thin] (#3,0) -- (#3,#2);
413     \draw[gray,thin] (0,#4) -- (#1,#4);
414     \node at (#3,#4) {\tikz\randuck};
415     \draw[black] (0,0) rectangle (#1,#2);
416   \end{tikzpicture}
417   }<<=
```

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

`\duckuments_patch_includegraphics:`

```
418 \cs_new_protected_nopar:Npn \duckuments_patch_includegraphics:
419   {>>=
420   \ifpackageloaded { graphicx }
421     {
422       \LetLtxMacro\duckuments@includegraphicsBAK\includegraphics
423       \RenewDocumentCommand \includegraphics
424         { >{\duckuments_starred:n}s 0{ } o m }
425         {
426           \IfValueTF { ##3 }
427             { \duckuments@includegraphicsBAK##1[##2][##3]{##4} }
428             {
429               \regex_match:NnTF \c_duckuments_example_regex { ##4 }
430                 {
431                   \duckuments@includegraphicsBAK##1
432                     [{page=\duckuments_random_page:,##2}]
433                     { ##4 }
434                 }
435                 {
436                   \duckuments@includegraphicsBAK##1[##2]{##4}
437                 }
438             }
439         }
440     }
441   }<<=
```

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

\duckuments_blindduck_range_test:n

```
443 \cs_new_protected:Npn \duckuments_blindduck_range_test:n #1
444   {%>>=
445     \regex_match:NnTF \c_duckuments_range_regex { #1 }
446     {
447       \seq_set_split:Nnn \l_duckuments_range_seq { - } { #1 }
448       \int_compare:nNnTF
449         { \c_one_int } = { \seq_count:N \l_duckuments_range_seq }
450         {
451           \cs_set:Npn \duckuments@blindduck@text
452             {
453               \duckuments_blindduck_single_par:n { #1 }
454               \duckuments_blindduck_set_next_start:n { #1 }
455             }
456           }
457         {
458           \bool_set_true:N \l_duckuments_blindduck_pars_bool
459           \exp_args:Nx
460           \tl_if_empty:nF
461             { \seq_item:Nn \l_duckuments_range_seq { \c_one_int } }
462             {
463               \int_gset:Nn \g_duckuments_blindduck_start_int
464                 { \seq_item:Nn \l_duckuments_range_seq { \c_one_int } }
465             }
466           \exp_args:Nx
467           \tl_if_empty:nTF { \seq_item:Nn \l_duckuments_range_seq { 2 } }
468             {
469               \int_gset_eq:NN
470                 \g_duckuments_blindduck_end_int
471                 \g_duckuments_blindduck_start_int
472             }
473             {
474               \int_set:Nn \g_duckuments_blindduck_end_int
475                 { \seq_item:Nn \l_duckuments_range_seq { 2 } }
476             }
477           \duckuments_blindduck_set_text:xx
478             { \int_use:N \g_duckuments_blindduck_start_int }
479             { \int_use:N \g_duckuments_blindduck_end_int }
480         }
481       }
482     {
483       \exp_args:NnnV
484       \msg_error:nnn { duckuments } { option-unknown } \l_keys_key_tl
485     }
486   }%<<
487 \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

```
488 \cs_new:Npn \duckuments_blindduck_set_text:nn #1 #2
489   {%>>=
490     \def \duckuments@blindduck@text
491     {
```

```

492     \int_step_function:nnnN { #1 } { \c_one_int } { #2 }
493     \duckuments_blindduck_par_loop:n
494     \duckuments_blindduck_set_next_start:n { #2 }
495   }
496   }%<<
497 \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`

```

498 \cs_new:Npn \duckuments_blindduck_set_next_start:n #1
499   {%>>=
500   \int_gset:Nn \g_duckuments_blindduck_start_int
501     { \int_mod:nn { #1 } { \c_duckuments_blindduck_pars_int } + \c_one_int }
502   }%<<

```

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

`\duckuments_blindduck_single_par:n`

```

503 \cs_new:Npn \duckuments_blindduck_single_par:n #1
504   {%>>=
505   \bool_if:nTF
506     {
507     \int_compare_p:nNn { #1 } > { \c_duckuments_blindduck_pars_int }
508     || \int_compare_p:nNn { #1 } < { \c_one_int }
509     }
510     {
511     \msg_error:nxxxx { duckuments } { out-of-range }
512     { 1 } { \int_use:N \c_duckuments_blindduck_int } { #1 }
513     { blindduck-paragraphs }
514     \duckuments@blindduck@text@i
515     }
516     {
517     \use:c { duckuments@blindduck@text@ \int_to_roman:n { #1 } }
518     }
519   }%<<

```

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

`\duckuments_blindduck_par_loop:n`

```

520 \cs_new:Npn \duckuments_blindduck_par_loop:n #1
521   {%>>=
522   \duckuments_blindduck_single_par:n { #1 }
523   \par
524   }%<<

```

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

`\duckuments_starred:n`

```

525 \cs_new_protected:Npn \duckuments_starred:n #1
526   {%>>=
527   \IfBooleanTF { #1 }
528     { \def\ProcessedArgument{*} }
529     { \def\ProcessedArgument{} }
530   }%<<

```


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

\duckuments_random_page:

```
531 \cs_new:Npn \duckuments_random_page:
532   { \int_rand:n { \c_duckuments_example_pages_int } }
```

(End definition for \duckuments_random_page:. This function is documented on page ??.)

```
533 \ExplSyntaxOff
```

\duckuments@blindduck@text

```
534 \newcommand*\duckuments@blindduck@text{\duckuments@blindduck@text@i}
535 \newcommand*\duckuments@blindduck@text@i
536   {%>>=
537   There once was a very smart but sadly blind duck. When it was still a small
538   duckling it was renowned for its good vision. But sadly as the duck grew
539   older it caught a sickness which caused its eyesight to worsen. It became so
540   bad, that the duck couldn't read the notes it once took containing much of
541   inline math\duckuments@ifinline{ just like its favoured equation:  $d = r \cdot k$ . Only displayed equations remained legible%
542   \duckuments@ifdisplay[.]{ so it could still read \begin{equation}d = r \cdot k
543   e\hbox{.}\end{equation}} That annoyed the smart duck, as it wasn't able to
544   do its research any longer. It called for its underduckling and said:
545   \duckuments@enquote{Go, find me the best eye ducktor there is. He shall
546   heal me from my disease!}%
547   }%=<<
548 }%=<<
549 \newcommand*\duckuments@blindduck@text@ii
550   {%>>=
551   \duckuments@enquote{But my duck, how are you supposed to manage your daily
552   routines without my visual guidance}, replied the underduckling. The smart
553   duck's face turned grim in anger. \duckuments@enquote{You dare to talk
554   back?} The underduckling blushed ashamed. How could he have objections
555   after his duck gave strict orders? The underduckling was so embarrassed
556   about his own behaviour he had to solve an equation.%
557   }%=<<
558 }%=<<
559 \newcommand*\duckuments@blindduck@text@iii
560   {%>>=
561   After the equation was solved and the underduckling prepared his leave for
562   the next day it fell asleep in a shaky mood. It did not know what the
563   journey had prepared for him and if he was prepared enough for it. His sleep
564   was restless. The dreams he had that night were not calm and bright as they
565   used to be for an innocent underduckling.%
566   }%=<<
567 }%=<<
568 \newcommand*\duckuments@blindduck@text@iv
569   {%>>=
570   Before dawn the underduckling woke. He didn't have the feeling of being well
571   rested. But nonetheless he knew that this was the day he should leave.
572   Except saying goodbye to his beloved ones there was nothing holding him
573   back. His duck had sent him on the most important mission a five weeks old
574   inexperienced underduckling was ever sent on. He bid farewell to his mother,
575   all his brothers and sisters, and finally from his duck. The bag was
576   shouldered, the boots were tied, the underduckling left.%
577   }%=<<
578 }%=<<
579 \newcommand*\duckuments@blindduck@text@v
580   {%>>=
```

```

578     As the underduckling went along he saw a tree. It was so tall he couldn't
579     even approximate its height. Oh, how small he felt. What should he, a small
580     duckling whose plumage was still fluffy, achieve in this vast, gigantic
581     world? He stuck his trembling wings under the straps of his backpack to calm
582     his shaking limbs. Where should he start his search? How should he survive
583     in the hostile nature surrounding the sheltered pond of his community?
584     }%=<<

```

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

\duckuments@headings@text

```

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

```

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

\ducklists@content

```

587 \newcommand*\ducklists@content
588   {%>>=
589     \item First swims father drake
590     \item Then floats mother duck
591     \item After her paddles baby duckling
592     \item And over there bathes uncle canard
593   }%=<<

```

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

\ducklists@content@starred

```

594 \newcommand*\ducklists@content@starred
595   {%>>=
596     \item[drake] is the swimming father
597     \item[duck] is the floating mother
598     \item[duckling] is the paddling baby
599     \item[canard] is the bathing uncle
600   }%=<<

```

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

\duckuments@listlist

```

601 \newcommand*\duckuments@listlist[1]
602   {%>>=
603     \begin{#1}
604       \item swimming father drake
605       \begin{#1}
606         \item swimming father drake
607         \begin{#1}
608           \item swimming father drake
609           \begin{#1}
610             \item swimming father drake
611             \item floating mother duck
612           \end{#1}
613         \item floating mother duck
614       \end{#1}
615     \item floating mother duck
616   \end{#1}

```

```

617     \item floating mother duck
618     \end{#1}%
619 }%=<<

```

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

`\duckuments@listlist@starred`

```

620 \newcommand*\duckuments@listlist@starred[1]
621 {%>>=
622   \begin{#1}
623     \item[drake] is the swimming father
624     \begin{#1}
625       \item[drake] is the swimming father
626       \begin{#1}
627         \item[drake] is the swimming father
628         \begin{#1}
629           \item[drake] is the swimming father
630           \item[duck] is the floating mother
631         \end{#1}
632         \item[duck] is the floating mother
633       \end{#1}
634       \item[duck] is the floating mother
635     \end{#1}
636     \item[duck] is the floating mother
637   \end{#1}%
638 }%=<<

```

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

```

639 \endinput

```

```

640 </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?