

The duckuments package

Jonathan P. Spratte*

version 0.4e, released 2019-09-28

Contents

1	Introducttion	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	7
3.5	Options and Configurations	8
3.6	Functions	9
3.6.1	Duckument Level	9
3.6.2	Intern	12
4	The story of the duck	20

1 Introducttion

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

<code>\duckument</code>	<code>\duckument[⟨key=value⟩]</code>
-------------------------	--------------------------------------

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.

<code>\blindduck</code>	<code>\blindduck[⟨key=value⟩]</code>
-------------------------	--------------------------------------

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.

<code>\ducklist</code>	<code>\ducklist⟨*⟩{⟨environment⟩}</code>
------------------------	--

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.

<code>\ducklistlist</code>	<code>\ducklistlist⟨*⟩{⟨list⟩}</code>
----------------------------	---------------------------------------

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.

<code>\duckitemize</code>	Abbreviation for <code>\ducklist{itemize}</code> .
---------------------------	--

<code>\duckenumerate</code>	Abbreviation for <code>\ducklist{enumerate}</code> .
-----------------------------	--

<code>\duckdescription</code>	Abbreviation for <code>\ducklist{description}</code> .
-------------------------------	--

2.2 Other Macros

<code>\duckumentsCreateExampleFiles</code>
--

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

<code>\duckumentsDrawRandomDucks</code>	<code>\duckumentsDrawRandomDucks⟨*⟩[⟨count⟩]</code>
---	---

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 `[\langle key=value \rangle]` 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:

`\langle num1 \rangle`

The paragraph `\langle num1 \rangle` is printed.

`\langle num1- \rangle`

Like the above

`\langle -num2 \rangle`

The paragraphs up to `\langle num2 \rangle` are printed.

`\langle num1-num2 \rangle`

The paragraphs from `\langle num1 \rangle` up to `\langle num2 \rangle` are printed.

`\langle - \rangle`

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 `\langle num1 \rangle` 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 which engine is used.

```

3 ^~A>>>
4 \bool_if:nF
5 {
6   \sys_if_engine luatex_p:
7   || \sys_if_engine pdftex_p:
8   || \sys_if_engine xetex_p:
9 }
10 {%>>>
11   \msg_new:nnnn { duckuments } { incompatible }
12   {
13     The~duckuments~package~is~currently~only~compatible~with~pdfTeX,~
14     LuaTeX,~and~XeTeX!
15   }
16   {
17     \ ~-----\\
18     \ (Quack!~Sorry~for~that.~Quack!)\\
19     \ ~~~~~~\\
20     | ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ | ~\ ~\ ~\ ~\string\ \\
21     | ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ | ~\ ~\ ~\ ~\ ~\ >()_\\
22     | | ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ | ~\ ~\ ~\ ~\ ~\ ~(__)__
23   }
24   \msg_error:nn { duckuments } { incompatible }
25   \endinput
26 }%<<<

```

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

```
27 ^^A>>>
28 \file_if_exist:nF { example-image-duck.pdf }
29 {%>>>
```

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

```

30 \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
31 {
32   \msg_new:nnnn { duckuments } { missing-file }
33   {
34     The~file~‘#1’~can’t~be~found.~Make~sure~to~create~it
35     \tl_if_empty:nF{#2}{~#2}.
36   }
37   { Sorry~for~the~inconvenience.~#3 }
38   \msg_error:nnnn { duckuments } { missing-file }
39   { example-image-duck.pdf }
40   {
41     by~compiling~example-image-duck.tex~at~least~once
42   }
43   {
44     If~you~don’t~find~the~file~on~your~machine~you~can~use~
45     ‘\duckumentsCreateExampleFiles’~in~your~document~to~produce~a~copy~

```

```

46         in~the~current~working~directory.
47     }
48 }
49 }%<<<

```

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

```

50 ^^A>>>
51 \file_if_exist:nF { example-image-duck-portrait.pdf }
52 {%>>>
53     \str_if_eq:VnF \c_sys_jobname_str { example-image-duck-portrait }
54     {
55         \msg_if_exist:nnF { duckuments } { missing~file }
56         {
57             \msg_new:nnnn { duckuments } { missing~file }
58             {
59                 The~file~‘#1’~can’t~be~found.~Make~sure~to~create~it
60                 \tl_if_empty:nF{#2}{~#2}.
61             }
62             { Sorry~for~the~inconvenience.~#3 }
63         }
64         \msg_error:nnnnn { duckuments } { missing~file }
65         { example-image-duck-portrait.pdf }
66         {
67             by~compiling~example-image-duck-portrait.tex~at~least~once
68         }
69         {
70             If~you~don’t~find~the~file~on~your~machine~you~can~use~
71             ‘\duckumentsCreateExampleFiles’~in~your~document~to~produce~a~copy~
72             in~the~current~working~directory.
73         }
74     }
75 }%<<<

```

3.2 Variables

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

```
76 \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.

```
77 \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`.

```
78 \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`.

```
79 \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`.
`80 \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`.
`81 \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`.
`82 \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.
`83 \int_new:N \g_duckuments_blindduck_start_int`
`84 \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.
`85 \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.
`86 \regex_const:Nn \c_duckuments_example_regex`
`87 {`
`88 example-image-duck`
`89 |example-image-duck.pdf`
`90 |example-image-duck-portrait`
`91 |example-image-duck-portrait.pdf`
`92 }`
(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.
`93 \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`
`94 \int_const:Nn \c_duckuments_blindduck_pars_int { 5 }`
(End definition for `\c_duckuments_blindduck_pars_int`. This variable is documented on page ??.)

```

95 \bool_if:nTF
96 {
97   \str_if_eq_p:Vn \c_sys_jobname_str { example-image-duck }
98   || \str_if_eq_p:Vn \c_sys_jobname_str { example-image-duck-portrait }
99 }
100 { \int_const:Nn \c_duckuments_example_pages_int { \c_one_int } }
101 {
102   \group_begin:
103     \sys_if_engine_pdftex:T
104     {
105       \pdfximage{example-image-duck.pdf}
106       \int_set:Nn \l_tmpa_int { \the\pdflastximagepages }
107       \pdfximage{example-image-duck-portrait.pdf}
108       \int_set:Nn \l_tmpb_int { \the\pdflastximagepages }
109     }
110     \sys_if_engine luatex:T
111     {
112       \saveimageresource{example-image-duck.pdf}
113       \int_set:Nn \l_tmpa_int { \lastsavedimageresourcepages }
114       \saveimageresource{example-image-duck-portrait.pdf}
115       \int_set:Nn \l_tmpb_int { \lastsavedimageresourcepages }
116     }
117     \sys_if_engine_xetex:T
118     {
119       \int_set:Nn \l_tmpa_int
120       { \XeTeXpdfpagecount "example-image-duck.pdf" }
121       \int_set:Nn \l_tmpb_int
122       { \XeTeXpdfpagecount "example-image-duck-portrait.pdf" }
123     }
124     \int_const:Nn \c_duckuments_example_pages_int
125     { \int_min:nn { \l_tmpa_int } { \l_tmpb_int } }
126   \group_end:
127 }

```

3.4 Messages

7

```
\duckuments_patch_see_duckumentation:
```

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

```

183 \keys_define:nn { duckuments }
184   {%>>>
185     ,immediate .bool_set:N = \l_duckuments_immediate_bool
186     ,immediate .default:n = true
187     ,full .code:n =
188       \duckuments_blindduck_range_test:n { 1-\c_duckuments_blindduck_pars_int }
189     ,all .meta:n = { full }

```



```

190 ,maths .choice:
191 ,maths / both .code:n =
192 {
193   \bool_set_true:N \l_duckuments_math_inline_bool
194   \bool_set_true:N \l_duckuments_math_display_bool
195 }
196 ,maths / display .code:n = \bool_set_true:N \l_duckuments_math_display_bool
197 ,maths / inline .code:n = \bool_set_true:N \l_duckuments_math_inline_bool
198 ,maths / none .code:n =
199 {
200   \bool_set_false:N \l_duckuments_math_inline_bool
201   \bool_set_false:N \l_duckuments_math_display_bool
202 }
203 ,maths .default:n = both
204 ,toc .code:n =
205 {
206   \str_case:nnF { #1 }
207   {
208     { true } { \bool_set_true:N \l_duckuments_toc_bool }
209     { false } { \bool_set_false:N \l_duckuments_toc_bool }
210   }
211   {
212     \msg_warning:nnn { duckuments } { non-boolean-toc } { #1 }
213   }
214 }
215 ,toc .default:n = true
216 ,unknown .code:n =
217 { \msg_error:nnx { duckuments } { option-unknown } { \l_keys_key_tl } }
218 }%<<<
219 \ProcessKeysOptions { duckuments }
220 \keys_define:nn { duckuments }
221 {%>>>
222 ,toc .undefine:
223 ,toc .bool_set:N = \l_duckuments_toc_bool
224 ,toc .default:n = true
225 ,unknown .code:n = \duckuments_blindduck_range_test:V \l_keys_key_tl
226 }%<<<
227 \bool_if:NTF \l_duckuments_immediate_bool
228 { \AtEndOfPackage { \duckuments_patch_includegraphics: } }
229 { \AtBeginDocument { \duckuments_patch_includegraphics: } }

```

3.6 Functions

3.6.1 Duckument Level

\duckument

```

230 \NewDocumentCommand \duckument { 0{ } }
231 {%>>>
232   \group_begin:
233   \keys_set:nn { duckuments } { #1 }
234   \bool_if:NT \l_duckuments_toc_bool { \tableofcontents }
235   \cs_if_exist_use:NT \chapter
236   { {\duckuments@headings@text{0}} \blindduck }
237   \duckuments@headings{1} \blindduck
238   \duckuments@headings{2} \blindduck

```

```

239 \duckuments@headings{3} \blindduck
240 \duckuments@headings{4} \blindduck
241 \section {Lists}
242 \duckuments_list_example:n { itemize }
243 \duckuments_list_example:n { enumerate }
244 \duckuments_list_example:n { description }
245 \group_end:
246 }%<<<

```

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

\blindduck

```

247 \NewDocumentCommand \blindduck { 0{ } }
248 {%>>>
249 \group_begin:
250 \keys_set:nn { duckuments } { #1 }
251 \duckuments@blindduck@text
252 \bool_if:NT \l_duckuments_blindduck_pars_bool { \par }
253 \group_end:
254 }%<<<

```

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

\ducklist

```

255 \NewDocumentCommand \ducklist { s m }
256 {%>>>
257 \begin{#2}
258 \IfBooleanTF { #1 }
259 { \ducklists@content@starred }
260 {
261 \str_if_eq:nnTF { #2 } { description }
262 \ducklists@content@starred
263 \ducklists@content
264 }
265 \end{#2}
266 }%<<<

```

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

\ducklistlist

```

267 \NewDocumentCommand \ducklistlist { s m }
268 {%>>>
269 \IfBooleanTF { #1 }
270 { \duckuments@listlist@starred { #2 } }
271 {
272 \str_if_eq:nnTF { #2 } { description }
273 { \duckuments@listlist@starred { description } }
274 { \duckuments@listlist{#2} }
275 }
276 }%<<<

```

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

\duckenumerate

```

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

```

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

`\duckitemize`

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

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

`\duckdescription`

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

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

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

```
280 \newcommand*\duckumentsCreateExampleFiles
281   {%>>>
282     \iow_new:N \duckuments_example_file_iow
283     \iow_open:Nn \duckuments_example_file_iow { example-image-duck.tex }
284     \iow_now:Nn \duckuments_example_file_iow
285       { \documentclass[tikz, multi]{standalone} }
286     \iow_now:Nn \duckuments_example_file_iow
287       { \usepackage{tikzducks} }
288     \iow_now:Nn \duckuments_example_file_iow
289       { \usepackage{duckuments} }
290     \iow_now:Nn \duckuments_example_file_iow
291       { \begin{document} }
292     \iow_now:Nn \duckuments_example_file_iow
293       { \duckumentsDrawRandomDucks }
294     \iow_now:Nn \duckuments_example_file_iow
295       { \end{document} }
296     \iow_close:N \duckuments_example_file_iow
```

Create example-image-duck-portrait.tex

```
297     \iow_open:Nn \duckuments_example_file_iow
298       { example-image-duck-portrait.tex }
299     \iow_now:Nn \duckuments_example_file_iow
300       { \documentclass[tikz, multi]{standalone} }
301     \iow_now:Nn \duckuments_example_file_iow
302       { \usepackage{tikzducks} }
303     \iow_now:Nn \duckuments_example_file_iow
304       { \usepackage{duckuments} }
305     \iow_now:Nn \duckuments_example_file_iow
306       { \begin{document} }
307     \iow_now:Nn \duckuments_example_file_iow
308       { \duckumentsDrawRandomDucks* }
309     \iow_now:Nn \duckuments_example_file_iow
310       { \end{document} }
311     \iow_close:N \duckuments_example_file_iow
312   }%<<<
```

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

`\duckumentsDrawRandomDucks`

```
313 \newcommand*\duckumentsDrawRandomDucks
314   {%>>>
315     \@ifstar
```

```

316         {\duckumentsDrawRandomDucks@portrait}
317         {\duckumentsDrawRandomDucks@landscape}%
318     }%<<<

```

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

3.6.2 Intern

`\duckuments@headings`

```

319 \newcommand*\duckuments@headings[1]
320 {%>>>
321     \ifcase#1\relax
322     \expandafter\chapter
323     \or \expandafter\section
324     \or \expandafter\subsection
325     \or \expandafter\subsubsection
326     \or \expandafter\paragraph
327     \else \expandafter\@gobble
328     \fi
329     {\duckuments@headings@text{#1}}
330 }%<<<

```

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

`\duckuments@headings@level`

```

331 \newcommand*\duckuments@headings@level[1]
332 {%>>>
333     (
334     \ifcase#1
335     chapter
336     \or section
337     \or subsection
338     \or subsubsection
339     \or paragraph
340     \fi
341     )
342 }%<<<

```

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

`\duckuments@ifinline`

```

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

```

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

`\duckuments@ifdisplay`

```

345 \newcommand*\duckuments@ifdisplay[2] []
346 { \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

```

347 \cs_new_protected_nopar:Npn \duckuments_list_example:n #1
348 {%>>>
349   \subsection{Example\ for\ ducks\ (#1)}
350   \ducklist { #1 }
351   \subsubsection{Nested\ ducks}
352   \ducklistlist { #1 }
353   }%<<<

```

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

\duckuments@enquote

```

354 \NewDocumentCommand \duckuments@enquote { s +m }
355 {%>>>
356   \IfBooleanTF { #1 }
357   {
358     \cs_if_exist_use:NTF
359     \enquote { * { #2 } }
360     { '#2' }
361   }
362   {
363     \cs_if_exist_use:NTF
364     \enquote { { #2 } }
365     { '{ #2}' }
366   }
367   }%<<<

```

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

\duckumentsDrawRandomDucks@landscape

```

368 \newcommand*\duckumentsDrawRandomDucks@landscape[1][\duckuments@randoms]
369 {%>>>
370   \foreach\x in {1,2,...,#1}
371   {\duckumentsDrawRandomDucks@draw{6.47214}{4}{3.23607}{2}}
372   }%<<<

```

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

\duckumentsDrawRandomDucks@portrait

```

373 \newcommand*\duckumentsDrawRandomDucks@portrait[1][\duckuments@randoms]
374 {%>>>
375   \foreach\x in {1,2,...,#1}
376   {\duckumentsDrawRandomDucks@draw{1.8541}{3}{.92705}{1.5}}
377   }%<<<

```

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

\duckumentsDrawRandomDucks@draw

```

378 \newcommand*\duckumentsDrawRandomDucks@draw[4]
379 {%>>>
380   \begin{tikzpicture}
381     \draw[fill=gray!50,use-as-bounding-box] (0,0) rectangle (#1,#2);
382     \draw[gray,thin] (0,0) -- (#1,#2);
383     \draw[gray,thin] (0,#2) -- (#1,0);
384     \draw[gray,thin] (#3,0) -- (#3,#2);

```

```

385     \draw[gray,thin] (0,#4) -- (#1,#4);
386     \node at (#3,#4) {\tikz\randuck;};
387     \draw[black] (0,0) rectangle (#1,#2);
388     \end{tikzpicture}
389 }%<<<

```

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

\duckuments_patch_includegraphics:

```

390 \cs_new_protected_nopar:Npn \duckuments_patch_includegraphics:
391   {%>>>
392     \ifpackageloaded { graphicx }
393     {
394       \LetLtxMacro\duckuments@includegraphicsBAK\includegraphics
395       \RenewDocumentCommand \includegraphics
396       { >{\duckuments_starred:n}s O{ } o m }
397       {
398         \IfValueTF { ##3 }
399         { \duckuments@includegraphicsBAK##1[{\##2}][{\##3}]{\##4} }
400         {
401           \regex_match:NnTF \c_duckuments_example_regex { ##4 }
402           {
403             \duckuments@includegraphicsBAK##1
404             [{page=\duckuments_random_page:,##2}]
405             { ##4 }
406           }
407           {
408             \duckuments@includegraphicsBAK##1[{\##2}]{\##4}
409           }
410         }
411       }
412     }
413   }%<<<
414

```

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

\duckuments_blindduck_range_test:n

```

415 \cs_new_protected:Npn \duckuments_blindduck_range_test:n #1
416   {%>>>
417     \regex_match:NnTF \c_duckuments_range_regex { #1 }
418     {
419       \seq_set_split:Nnn \l_duckuments_range_seq { - } { #1 }
420       \int_compare:nNnTF
421       { \c_one_int } = { \seq_count:N \l_duckuments_range_seq }
422       {
423         \cs_set:Npn \duckuments@blindduck@text
424         {
425           \duckuments_blindduck_single_par:n { #1 }
426           \duckuments_blindduck_set_next_start:n { #1 }
427         }
428       }
429       {
430         \bool_set_true:N \l_duckuments_blindduck_pars_bool
431         \exp_args:Nx

```

```

432 \tl_if_empty:nF
433 { \seq_item:Nn \l_duckuments_range_seq { \c_one_int } }
434 {
435   \int_gset:Nn \g_duckuments_blindduck_start_int
436   { \seq_item:Nn \l_duckuments_range_seq { \c_one_int } }
437 }
438 \exp_args:Nx
439 \tl_if_empty:nTF { \seq_item:Nn \l_duckuments_range_seq { 2 } }
440 {
441   \int_gset_eq:NN
442   \g_duckuments_blindduck_end_int
443   \g_duckuments_blindduck_start_int
444 }
445 {
446   \int_set:Nn \g_duckuments_blindduck_end_int
447   { \seq_item:Nn \l_duckuments_range_seq { 2 } }
448 }
449 \duckuments_blindduck_set_text:xx
450 { \int_use:N \g_duckuments_blindduck_start_int }
451 { \int_use:N \g_duckuments_blindduck_end_int }
452 }
453 }
454 {
455   \exp_args:NnnV
456   \msg_error:nnn { duckuments } { option~unknown } \l_keys_key_tl
457 }
458 }%<<<
459 \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

```

460 \cs_new:Npn \duckuments_blindduck_set_text:nn #1 #2
461 { %>>>
462   \def \duckuments@blindduck@text
463   {
464     \int_step_function:nnnN { #1 } { \c_one_int } { #2 }
465     \duckuments_blindduck_par_loop:n
466     \duckuments_blindduck_set_next_start:n { #2 }
467   }
468   }%<<<
469 \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

```

470 \cs_new:Npn \duckuments_blindduck_set_next_start:n #1
471 { %>>>
472   \int_gset:Nn \g_duckuments_blindduck_start_int
473   { \int_mod:nn { #1 } { \c_duckuments_blindduck_pars_int } + \c_one_int }
474   }%<<<

```

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

\duckuments_blindduck_single_par:n

```

475 \cs_new:Npn \duckuments_blindduck_single_par:n #1
476   {%>>>
477   \bool_if:nTF
478     {
479       \int_compare_p:nNn { #1 } > { \c_duckuments_blindduck_pars_int }
480       || \int_compare_p:nNn { #1 } < { \c_one_int }
481     }
482     {
483       \msg_error:nnxxxx { duckuments } { out-of-range }
484       { 1 } { \int_use:N \c_duckuments_blindduck_pars_int } { #1 }
485       { blindduck~paragraphs }
486       \duckuments@blindduck@text@i
487     }
488     {
489       \use:c { duckuments@blindduck@text@ \int_to_roman:n { #1 } }
490     }
491   }%<<<

```

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

\duckuments_blindduck_par_loop:n

```

492 \cs_new:Npn \duckuments_blindduck_par_loop:n #1
493   {%>>>
494   \duckuments_blindduck_single_par:n { #1 }
495   \par
496   }%<<<

```

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

\duckuments_starred:n

```

497 \cs_new_protected:Npn \duckuments_starred:n #1
498   {%>>>
499   \IfBooleanTF { #1 }
500     { \def\ProcessedArgument{*} }
501     { \def\ProcessedArgument{} }
502   }%<<<

```

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

\duckuments_random_page:

```

503 \cs_new:Npn \duckuments_random_page:
504   { \int_rand:n { \c_duckuments_example_pages_int } }

```

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

```

505 \ExplSyntaxOff

```

\duckuments@blindduck@text

```

506 \newcommand*\duckuments@blindduck@text{\duckuments@blindduck@text@i}
507 \newcommand*\duckuments@blindduck@text@i
508   {%>>>
509   There once was a very smart but sadly blind duck. When it was still a small
510   duckling it was renowned for its good vision. But sadly as the duck grew
511   older it caught a sickness which caused its eyesight to worsen. It became so

```



```

512 bad, that the duck couldn't read the notes it once took containing much of
513 inline math\duckuments@ifinline{ just like its favoured equation:  $d = u_c$ 
514  $\cdot k$ }. Only displayed equations remained legible%
515 \duckuments@ifdisplay[.]{ so it could still read \begin{equation}d = r a^k
516 e\hbox{.}\end{equation}} That annoyed the smart duck, as it wasn't able to
517 do its research any longer. It called for its underduckling and said:
518 \duckuments@enquote{Go, find me the best eye ducktor there is. He shall
519 heal me from my disease!}%
520 }%<<<
521 \newcommand*\duckuments@blindduck@text@ii
522 {%>>>
523 \duckuments@enquote{But my duck, how are you supposed to manage your daily
524 routines without my visual guidance}, replied the underduckling. The smart
525 duck's face turned grim in anger. \duckuments@enquote{You dare to talk
526 back?} The underduckling blushed ashamed. How could he have objections
527 after his duck gave strict orders? The underduckling was so embarrassed
528 about his own behaviour he had to solve an equation.%
529 }%<<<
530 \newcommand*\duckuments@blindduck@text@iii
531 {%>>>
532 After the equation was solved and the underduckling prepared his leave for
533 the next day it fell asleep in a shaky mood. It did not know what the
534 journey had prepared for him and if he was prepared enough for it. His sleep
535 was restless. The dreams he had that night were not calm and bright as they
536 used to be for an innocent underduckling.%
537 }%<<<
538 \newcommand*\duckuments@blindduck@text@iv
539 {%>>>
540 Before dawn the underduckling woke. He didn't have the feeling of being well
541 rested. But nonetheless he knew that this was the day he should leave.
542 Except saying goodbye to his beloved ones there was nothing holding him
543 back. His duck had sent him on the most important mission a five weeks old
544 inexperienced underduckling was ever sent on. He bid farewell to his mother,
545 all his brothers and sisters, and finally from his duck. The bag was
546 shouldered, the boots were tied, the underduckling left.%
547 }%<<<
548 \newcommand*\duckuments@blindduck@text@v
549 {%>>>
550 As the underduckling went along he saw a tree. It was so tall he couldn't
551 even approximate its height. Oh, how small he felt. What should he, a small
552 duckling whose plumage was still fluffy, achieve in this vast, gigantic
553 world? He stuck his trembling wings under the straps of his backpack to calm
554 his shaking limbs. Where should he start his search? How should he survive
555 in the hostile nature surrounding the sheltered pond of his community?
556 }%<<<

```

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

\duckuments@headings@text

```

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

```

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

\ducklists@content

```
559 \newcommand*\ducklists@content
560 {%>>>
561   \item First swims father drake
562   \item Then floats mother duck
563   \item After her paddles baby duckling
564   \item And over there bathes uncle canard
565 }%<<<
```

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

\ducklists@content@starred

```
566 \newcommand*\ducklists@content@starred
567 {%>>>
568   \item[drake] is the swimming father
569   \item[duck] is the floating mother
570   \item[duckling] is the paddling baby
571   \item[canard] is the bathing uncle
572 }%<<<
```

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

\duckuments@listlist

```
573 \newcommand*\duckuments@listlist[1]
574 {%>>>
575   \begin{#1}
576     \item swimming father drake
577     \begin{#1}
578       \item swimming father drake
579       \begin{#1}
580         \item swimming father drake
581         \begin{#1}
582           \item swimming father drake
583           \item floating mother duck
584         \end{#1}
585       \item floating mother duck
586     \end{#1}
587     \item floating mother duck
588   \end{#1}
589   \item floating mother duck
590 \end{#1}%
591 }%<<<
```

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

\duckuments@listlist@starred

```
592 \newcommand*\duckuments@listlist@starred[1]
593 {%>>>
594   \begin{#1}
595     \item[drake] is the swimming father
596     \begin{#1}
597       \item[drake] is the swimming father
598       \begin{#1}
599         \item[drake] is the swimming father
```

```

600         \begin{#1}
601             \item[drake] is the swimming father
602             \item[duck] is the floating mother
603         \end{#1}
604         \item[duck] is the floating mother
605     \end{#1}
606     \item[duck] is the floating mother
607 \end{#1}
608     \item[duck] is the floating mother
609 \end{#1}%
610 }%<<<

```

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

```

611 \endinput
612 </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?