

# The duckuments package

Jonathan P. Spratte\*

version v0.2, released 2018/03/14

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Duckumentation</b>	<b>2</b>
2.1	Dummy content . . . . .	2
2.2	Other Macros . . . . .	2
2.3	Patches . . . . .	3
2.4	Options . . . . .	3
<b>3</b>	<b>Implementation</b>	<b>3</b>
3.1	Check for possible problems . . . . .	3
3.2	Variables . . . . .	4
3.3	Constants . . . . .	5
3.4	Options and Configurations . . . . .	7
3.5	Functions . . . . .	8
3.5.1	Duckument Level . . . . .	8
3.5.2	Intern . . . . .	10

## 1 Introduction

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` and `example-image-duck.tex` file and typesets this duckumentation; execute `tex duckuments.dtx` to only generate the files `duckuments.sty` and `example-image-duck.tex`.

For its functionality `example-image-duck.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](mailto:jspratte@yahoo.de)

## 2 Duckumentation

### 2.1 Dummy 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>\duckumentsCreateExampleFile</code>
---

Creates the file `example-image-duck.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`.

## 2.3 Patches

The package patches `\includegraphics` if it is defined at the time the patch is applied (see [subsection 2.4](#), `immediate`). The patch changes the behaviour if `example-image-duck` is used. 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.

`toc=true|false`

If **true** the `\duckument` contains a ToC.

`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.

`immediate=true|false`

If **true** `\includegraphics` is patched during package load time, else the patching is done `\AtBeginDocument`.

## 3 Implementation

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

### 3.1 Check for possible problems

Check which engine is used.

```
3 \bool_if:nF
4 {
5   \sys_if_engine luatex_p:
6   || \sys_if_engine pdftex_p:
7   || \sys_if_engine xetex_p:
8 }
9 {%>>>
10 \msg_new:nnnn { duckuments } { incompatible }
11 {
12   The~duckuments~package~is~currently~only~compatible~with~pdfTeX,~
13   LuaTeX,~and~XeTeX!
14 }
```

```

15      {
16      Sorry~for~that.
17      }
18      \msg_error:nn { duckuments } { incompatible }
19      \endinput
20  }%<<<

```

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

```

21 \file_if_exist:nF { example-image-duck.pdf }
22 {%>>>

```

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

```

23 \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
24 {
25     \msg_new:nnnn { duckuments } { missing-file }
26     {
27         The~file~‘#1’~can’t~be~found.~Make~sure~to~create~it
28         \tl_if_empty:nF{#2}{~#2}.
29     }
30     { Sorry~for~the~inconvenience.~#3 }
31     \msg_error:nnnn { duckuments } { missing-file }
32     { example-image-duck.pdf }
33     {
34         by~compiling~example-image-duck.tex~at~least~once
35     }
36     {
37         If~you~don’t~find~the~file~on~your~machine~you~can~use~
38         ‘\duckumentsCreateExampleFile’~in~your~document~to~produce~a~copy~
39         in~the~current~working~directory.
40     }
41 }
42 }%<<<

```

## 3.2 Variables

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

```

43 \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.

```

44 \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`.

```

45 \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`.

```

46 \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`.  
`47 \bool_new:N \l_duckuments_math_display_bool`  
*(End definition for `\l_duckuments_math_display_bool`. This variable is documented on page ??.)*

### 3.3 Constants

`\c_duckuments_regex` Regex against which the patch of `\includegraphics` is testing.  
`48 \regex_const:Nn \c_duckuments_regex`  
`49 { example-image-duck|example-image-duck.pdf }`  
*(End definition for `\c_duckuments_regex`. This variable is documented on page ??.)*

`\c_duckuments_example_pages_int`

```

50 \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
51 {\file_if_exist:nTF { example-image-duck.pdf }
52 {
53   \sys_if_engine_pdftex:T
54   {
55     \pdfximage{example-image-duck.pdf}
56     \int_const:Nn \c_duckuments_example_pages_int
57     { \the\pdflastximagepages }
58   }
59   \sys_if_engine luatex:T
60   {
61     \saveimageresource{example-image-duck.pdf}
62     \int_const:Nn \c_duckuments_example_pages_int
63     { \lastsavedimageresourcepages }
64   }
65   \sys_if_engine_xetex:TF
66   {
67     \int_set:Nn \l_tmpa_int { \XeTeXpdfpagecount "example-image-duck.pdf" }
68     \int_const:Nn \c_duckuments_example_pages_int { \l_tmpa_int }

```

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

```

69 \int_new:N \g_duckuments_RCiv_i_int
70 \int_new:N \g_duckuments_RCiv_j_int
71 \int_new:N \g_duckuments_RCiv_keylength_int
72 \int_new:N \g_duckuments_tmpa_int
73 \int_const:Nn \c_duckuments_RCiv_Slength_int { 256 }
74 \tl_new:N \l_duckuments_tmpa_tl
75 \tl_new:N \l_duckuments_tmpb_tl
76 \cs_generate_variant:Nn \prop_gput:Nnn { Nxx }
77 \cs_generate_variant:Nn \prop_get:NnN { NxN }

```

Initialize the S array:

```

78 \prop_new:N \g_duckuments_RCiv_S_prop
79 \int_step_inline:nnnn { 0 } { 1 } { 255 }
80 {
81   \prop_gput:Nnn \g_duckuments_RCiv_S_prop { #1 } { #1 }
82 }
83 \prop_new:N \g_duckuments_RCiv_key_prop
84 \prop_gput:Nnx \g_duckuments_RCiv_key_prop
85 { 0 } { \int_use:N \c_sys_minute_int }

```

```

86     \prop_gput:Nnx \g_duckuments_RCiv_key_prop
87     { 1 } { \c_sys_hour_int }
88     \prop_gput:Nnx \g_duckuments_RCiv_key_prop
89     { 2 } { \c_sys_day_int }
90     \prop_gput:Nnx \g_duckuments_RCiv_key_prop
91     { 3 } { \c_sys_month_int }
92     \prop_gput:Nnx \g_duckuments_RCiv_key_prop
93     { 4 } { \int_eval:n { \int_mod:nn { \c_sys_year_int } { 256 } } }
94     \int_gset:Nn \g_duckuments_RCiv_keylength_int { 5 }
95     \str_map_inline:Nn \c_sys_jobname_str
96     {
97         \prop_gput:Nxx \g_duckuments_RCiv_key_prop
98         { \int_use:N \g_duckuments_RCiv_keylength_int }
99         { \int_from_alph:n { #1 } }
100        \int_gadd:Nn \g_duckuments_RCiv_keylength_int { 1 }
101    }
102    \cs_new:Nn \duckuments_swap_prop_entries:Nnn
103    {
104        \prop_get:NnN #1 { #2 } \l_duckuments_tmpa_tl
105        \prop_get:NnN #1 { #3 } \l_duckuments_tmpb_tl
106        \prop_gput:Nnx #1 { #2 } { \l_duckuments_tmpb_tl }
107        \prop_gput:Nnx #1 { #3 } { \l_duckuments_tmpa_tl }
108    }
109    \cs_generate_variant:Nn \duckuments_swap_prop_entries:Nnn { Nxx }
110    \cs_new:Nn \duckuments_gadd_mod:Nnn
111    {
112        \int_gset:Nn #1 { \int_mod:nn { #1 + #2 } { #3 } }
113    }
114    \cs_generate_variant:Nn \duckuments_gadd_mod:Nnn { Nxn }
115    \cs_new:Nn \duckuments_gadd_mod_Slength:Nn
116    {
117        \duckuments_gadd_mod:Nnn #1
118        { #2 } { \c_duckuments_RCiv_Slength_int }
119    }
120    \int_step_inline:nnnn { 0 } { 1 } { 255 }
121    {
122        \prop_get:NnN \g_duckuments_RCiv_S_prop { #1 } \l_duckuments_tmpa_tl
123        \int_gset:Nn \g_duckuments_tmpa_int
124        {
125            \int_mod:nn
126            { #1 }
127            { \g_duckuments_RCiv_keylength_int }
128        }
129        \prop_get:NxN \g_duckuments_RCiv_key_prop
130        { \int_use:N \g_duckuments_tmpa_int } \l_duckuments_tmpb_tl
131        \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_j_int
132        { \l_duckuments_tmpa_tl + \l_duckuments_tmpb_tl }
133        \duckuments_swap_prop_entries:Nxx \g_duckuments_RCiv_S_prop
134        { #1 } { \int_use:N \g_duckuments_RCiv_j_int }
135    }
136    \int_gzero:N \g_duckuments_RCiv_i_int
137    \int_gzero:N \g_duckuments_RCiv_j_int

```

Provide a function which gets the next random number and sets \duckuments\_random\_page:

to it.

```

138 \cs_new:Nn \duckuments_get_random_page:
139 {
140   \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_i_int { \c_one }
141   \prop_get:NxN \g_duckuments_RCiv_S_prop
142   { \int_use:N \g_duckuments_RCiv_i_int } \l_duckuments_tmpa_tl
143   \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_j_int
144   { \l_duckuments_tmpa_tl }
145   \duckuments_swap_prop_entries:Nxx \g_duckuments_RCiv_S_prop
146   { \int_use:N \g_duckuments_RCiv_i_int }
147   { \int_use:N \g_duckuments_RCiv_j_int }
148   \int_gset:Nn \g_duckuments_tmpa_int { \l_duckuments_tmpa_tl }
149   \duckuments_gadd_mod_Slength:Nn \g_duckuments_tmpa_int
150   { \l_duckuments_tmpb_tl }
151   \prop_get:NxN \g_duckuments_RCiv_S_prop
152   { \int_use:N \g_duckuments_tmpa_int } \l_duckuments_tmpa_tl
153   \int_gset:Nn \g_duckuments_tmpa_int
154   {
155     \int_mod:nn
156     { \l_duckuments_tmpa_tl } { \c_duckuments_example_pages_int }
157     + \c_one
158   }
159   \cs_set:Nx \duckuments_random_page:
160   { \int_use:N \g_duckuments_tmpa_int }
161 }
162 \cs_new:Nn \duckuments_random_page: { 1 }
163 }
164 {
165   \cs_new:Nn \duckuments_get_random_page: {}
166   \cs_new:Nn \duckuments_random_page:
167   { \int_rand:nn { 1 } { \c_duckuments_example_pages_int } }
168 }
169 }}
170 {
171   \int_const:Nn \c_duckuments_example_pages_int { 1 }
172 }

```

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

### 3.4 Options and Configurations

```

173 \keys_define:nn { duckuments }
174 {%>>>
175   ,toc .bool_set:N = \l_duckuments_toc_bool
176   ,toc .default:n = true
177   ,maths .choice:
178   ,maths / both .code:n =
179   {
180     \bool_set_true:N \l_duckuments_math_inline_bool
181     \bool_set_true:N \l_duckuments_math_display_bool
182   }
183   ,maths / none .code:n =
184   {
185     \bool_set_false:N \l_duckuments_math_inline_bool

```

```

186         \bool_set_false:N \l_duckuments_math_display_bool
187     }
188     ,maths / inline .code:n = \bool_set_true:N \l_duckuments_math_inline_bool
189     ,maths / display .code:n = \bool_set_true:N \l_duckuments_math_display_bool
190     ,maths .default:n = both
191     ,immediate .bool_set:N = \l_duckuments_immediate_bool
192     ,immediate .default:n = true
193 }%<<<
194 \ProcessKeysOptions { duckuments }
195 \bool_if:NTF \l_duckuments_immediate_bool
196 { \AtEndOfPackage { \duckuments_patch_includegraphics: } }
197 { \AtBeginDocument { \duckuments_patch_includegraphics: } }

```

## 3.5 Functions

### 3.5.1 Duckument Level

#### `\duckument`

```

198 \NewDocumentCommand \duckument { 0{ } }
199 {%>>>
200     \group_begin:
201     \keys_set:nn { duckuments } { #1 }
202     \bool_if:NT \l_duckuments_toc_bool { \tableofcontents }
203     \cs_if_exist_use:NT \chapter
204     { {\duckuments@headings@text{0}} \blindduck }
205     \duckuments@headings{1} \blindduck
206     \duckuments@headings{2} \blindduck
207     \duckuments@headings{3} \blindduck
208     \duckuments@headings{4} \blindduck
209     \section {Lists}
210     \duckuments_list_example:n { itemize }
211     \duckuments_list_example:n { enumerate }
212     \duckuments_list_example:n { description }
213     \group_end:
214 }%<<<

```

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

#### `\blindduck`

```

215 \NewDocumentCommand \blindduck { 0{ } }
216 {%>>>
217     \group_begin:
218     \keys_set:nn { duckuments } { #1 }
219     \duckuments@blindduck@text
220     \group_end:
221 }%<<<

```

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

#### `\ducklist`

```

222 \NewDocumentCommand \ducklist { s m }
223 {%>>>
224     \begin{#2}
225     \IfBooleanTF { #1 }
226     {\ducklists@content@starred}

```



```

227     {
228       \str_if_eq:nnTF { #2 } { description }
229       \ducklists@content@starred
230       \ducklists@content
231     }
232   \end{#2}
233 }%<<<

```

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

#### \ducklistlist

```

234 \NewDocumentCommand \ducklistlist { s m }
235 {%>>>
236   \IfBooleanTF { #1 }
237   { \duckuments@listlist@starred { #2 } }
238   {
239     \str_if_eq:nnTF { #2 } { description }
240     { \duckuments@listlist@starred { description } }
241     { \duckuments@listlist{#2} }
242   }
243 }%<<<

```

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

#### \duckenumerate

```

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

```

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

#### \duckitemize

```

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

```

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

#### \duckdescription

```

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

```

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

#### \duckumentsCreateExampleFile

```

247 \newcommand*\duckumentsCreateExampleFile
248 {%>>>
249   \iow_new:N \duckuments_example_file_iow
250   \iow_open:Nn \duckuments_example_file_iow { example-image-duck.tex }
251   \iow_now:Nn \duckuments_example_file_iow
252   { \documentclass[tikz,multi]{standalone} }
253   \iow_now:Nn \duckuments_example_file_iow
254   { \usepackage{tikzducks} }
255   \iow_now:Nn \duckuments_example_file_iow
256   { \usepackage{duckuments} }
257   \iow_now:Nn \duckuments_example_file_iow
258   { \begin{document} }
259   \iow_now:Nn \duckuments_example_file_iow
260   { \duckumentsDrawRandomDucks }
261   \iow_now:Nn \duckuments_example_file_iow

```

```

262     { \end{document} }
263     \iow_close:N \duckuments_example_file_iow
264 }%<<<

```

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

### `\duckumentsDrawRandomDucks`

```

265 \newcommand*\duckumentsDrawRandomDucks[1][\duckuments@randoms]
266 {%>>>
267   \foreach\x in {1,2,...,#1}
268   {\begin{tikzpicture}
269     \draw[black,fill=gray!50] (0,0) rectangle (6,4);
270     \draw[gray,thin] (0,0) -- (6,4);
271     \draw[gray,thin] (0,4) -- (6,0);
272     \draw[gray,thin] (3,0) -- (3,4);
273     \draw[gray,thin] (0,2) -- (6,2);
274     \node at (3,2) {\tikz\randuck;};
275   \end{tikzpicture}}
276 }%<<<

```

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

## 3.5.2 Intern

### `\duckuments@headings`

```

277 \newcommand*\duckuments@headings[1]
278 {%>>>
279   \ifcase#1\relax
280     \expandafter\chapter
281   \or \expandafter\section
282   \or \expandafter\subsection
283   \or \expandafter\subsubsection
284   \or \expandafter\paragraph
285   \else \expandafter\@gobble
286   \fi
287   {\duckuments@headings@text{#1}}
288 }%<<<

```

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

### `\duckuments@headings@level`

```

289 \newcommand*\duckuments@headings@level[1]
290 {%>>>
291   (
292     \ifcase#1
293       chapter
294     \or section
295     \or subsection
296     \or subsubsection
297     \or paragraph
298     \fi
299   )
300 }%<<<

```

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

`\duckuments@ifinline`

```

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

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

```

`\duckuments@ifdisplay`

```

303 \newcommand*\duckuments@ifdisplay[2] []
304 { \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`

```

305 \cs_new_protected_nopar:Nn \duckuments_list_example:n
306 { %>>>
307   \subsection{Example\ for\ ducks\ (#1)}
308   \ducklist { #1 }
309   \subsubsection{Nested\ ducks}
310   \ducklistlist { #1 }
311 } %<<<

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

```

`\duckuments@enquote`

```

312 \newcommand*\duckuments@enquote [1]
313 { %>>>
314   \cs_if_exist_use:NTF
315     \enquote { #1 }
316     { ‘ ‘ #1 ’ ’ }
317 } %<<<

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

```

`\duckuments_patch_includegraphics:`

```

318 \cs_new_protected_nopar:Nn \duckuments_patch_includegraphics:
319 { %>>>
320   \cs_if_exist:NT \includegraphics
321   {
322     \LetLtxMacro\duckuments@includegraphicsBAK\includegraphics
323     \RenewDocumentCommand \includegraphics
324       { >{\duckuments_starred:n}s O{ } m }
325       {
326         \regex_match:NnTF \c_duckuments_regex { ##3 }
327         {
328           \duckuments_get_random_page:
329           \duckuments@includegraphicsBAK##1
330             [page=\duckuments_random_page:,##2]
331             { ##3 }
332         }
333         {
334           \duckuments@includegraphicsBAK##1[##2]{##3}
335         }
336       }
337   }
338 } %<<<

```

(End definition for \duckuments\_patch\_includegraphics:. This function is documented on page ??.)

\duckuments\_starred:n

```
339 \cs_new_protected:Nn \duckuments_starred:n
340   {%>>>
341   \IfBooleanTF { #1 }
342     { \def\ProcessedArgument{*} }
343     { \def\ProcessedArgument{} }
344   }%<<<
```

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

```
345 \ExplSyntaxOff
```

\duckuments@blindduck@text

```
346 \newcommand*\duckuments@blindduck@text
347   {%>>>
348   There once was a very smart but sadly blind duck. When it was still a small
349   duckling it was renowned for its good vision. But sadly as the duck grew
350   older it caught a sickness which caused its eyesight to worsen. It became so
351   bad, that the duck couldn't read the notes it once took containing much of
352   inline math\duckuments@ifinline{ just like its favoured equation:  $d = u_c$ 
353   \cdot k$}. Only displayed equations remained legible%
354   \duckuments@ifdisplay[.]{ so it could still read \begin{equation}d = r a^k
355   e\hbox{.}\end{equation}} That annoyed the smart duck, as it wasn't able to
356   do its research any longer. It called for its underduckling and said:
357   \duckuments@enquote{Go, find me the best eye ducktor there is. He shall
358   heal me from my disease!}%
359   }%<<<
```

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

\duckuments@headings@text

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

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

\ducklists@content

```
362 \newcommand*\ducklists@content
363   {%>>>
364   \item First swims father drake
365   \item Then floats mother duck
366   \item After her paddles baby duckling
367   \item And over there bathes uncle canard
368   }%<<<
```

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

\ducklists@content@starred

```
369 \newcommand*\ducklists@content@starred
370   {%>>>
371   \item[drake] is the swimming father
372   \item[duck] is the floating mother
373   \item[duckling] is the paddling baby
374   \item[canard] is the bathing uncle
375   }%<<<
```

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

\duckuments@listlist

```

376 \newcommand*\duckuments@listlist[1]
377   {%>>>
378     \begin{#1}
379       \item swimming father drake
380       \begin{#1}
381         \item swimming father drake
382         \begin{#1}
383           \item swimming father drake
384           \begin{#1}
385             \item swimming father drake
386             \item floating mother duck
387           \end{#1}
388           \item floating mother duck
389         \end{#1}
390         \item floating mother duck
391       \end{#1}
392       \item floating mother duck
393     \end{#1}%
394   }%<<<

```

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

\duckuments@listlist@starred

```

395 \newcommand*\duckuments@listlist@starred[1]
396   {%>>>
397     \begin{#1}
398       \item[drake] is the swimming father
399       \begin{#1}
400         \item[drake] is the swimming father
401         \begin{#1}
402           \item[drake] is the swimming father
403           \begin{#1}
404             \item[drake] is the swimming father
405             \item[duck] is the floating mother
406           \end{#1}
407           \item[duck] is the floating mother
408         \end{#1}
409         \item[duck] is the floating mother
410       \end{#1}
411       \item[duck] is the floating mother
412     \end{#1}%
413   }%<<<

```

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

414 \endinput

415 \</pkg>