

The duckuments package

Jonathan P. Spratte*

version v0.3, released 2018/03/19

Contents

| | | |
|----------|---------------------------------------|-----------|
| 1 | Introduction | 1 |
| 2 | Duckumentation | 2 |
| 2.1 | Dummy 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 | 7 |
| 3.6 | Functions | 8 |
| 3.6.1 | Duckument Level | 8 |
| 3.6.2 | Intern | 10 |
| 4 | The story of the duck | 18 |

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

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. The *italic* printed value is the default.

`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.

`full`

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

`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 used instead of the first.

`<num1->`

Like the above but after the paragraph a `\par` is inserted

`<-num2>`

The paragraphs from one after the last time `<num1>` or `<num2>` was used as an argument (the latter if both) up to `<num2>` are printed. A `\par` is inserted after it.

`<num1-num2>`

The paragraphs from `<num1>` up to `<num2>` are printed. A `\par` is inserted after it.

`<->`

The next paragraph after the last time `<num1>` or `<num2>` was used (the latter if both) is printed. A `\par` is inserted after it.

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     Sorry~for~that.
18   }
19   \msg_error:nn { duckuments } { incompatible }
20   \endinput
21 }%<<<
```

Check whether example-image-duck.pdf exists.

```
22 ^^A>>>
23 \file_if_exist:nF { example-image-duck.pdf }
24 {%>>>
```

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

```
25   \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
26   {
27     \msg_new:nnnn { duckuments } { missing-file }
28     {
29       The~file~'#1'~can't~be~found.~Make~sure~to~create~it
30       \tl_if_empty:nF{#2}{~#2}.
31     }
32     { Sorry~for~the~inconvenience.~#3 }
33     \msg_error:nnnn { duckuments } { missing-file }
34     { example-image-duck.pdf }
35     {
36       by~compiling~example-image-duck.tex~at~least~once
37     }
38     {
39       If~you~don't~find~the~file~on~your~machine~you~can~use~
40       '\duckumentsCreateExampleFile'~in~your~document~to~produce~a~copy~
41       in~the~current~working~directory.
42     }
43   }
44 }%<<<
```

3.2 Variables

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

45 `\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.

46 `\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`.

47 `\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`.

48 `\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`.

49 `\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`.

50 `\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`.

51 `\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.

52 `\int_new:N \g_duckuments_blindduck_start_int`

53 `\int_gset:Nn \g_duckuments_blindduck_start_int { \c_one }`

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

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

54 `\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.

```

55 \regex_const:Nn \c_duckuments_example_regex
56   { example-image-duck|example-image-duck.pdf }

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

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

```

57 \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`

```

58 \int_const:Nn \c_duckuments_blindduck_pars_int { \c_five }

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

`\c_duckuments_example_pages_int`

```

59 \str_if_eq:VnF \c_sys_jobname_str { example-image-duck }
60   {
61     \file_if_exist:nTF { example-image-duck.pdf }
62     {
63       \sys_if_engine_pdftex:T
64       {
65         \pdfximage{example-image-duck.pdf}
66         \int_const:Nn \c_duckuments_example_pages_int
67           { \the\pdflastximagepages }
68       }
69       \sys_if_engine luatex:T
70       {
71         \saveimageresource{example-image-duck.pdf}
72         \int_const:Nn \c_duckuments_example_pages_int
73           { \lastsavedimageresourcepages }
74       }
75       \sys_if_engine_xetex:T
76       {
77         \group_begin:
78         \int_gset:Nn \l_tmpa_int
79           { \XeTeXpdfpagecount "example-image-duck.pdf" }
80         \int_const:Nn \c_duckuments_example_pages_int { \l_tmpa_int }
81         \group_end:
82       }
83     }
84   }
85   { \int_const:Nn \c_duckuments_example_pages_int { 1 } }
```

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

3.4 Messages

```

86 ~^A duckuments option-unknown >>>
87 \msg_new:nnnn { duckuments } { option-unknown }
88 {
89     Unknown~option~'#1'~for~package~duckuments.
90 }
91 {
92     \ ~-----\\
93     \ (Quack!~Nothing~here,~sorry.~Quack!)\\
94     \ ~~~~~\\
95     \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\string\ \\
96     \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\~>()_\\
97     \ | ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~(__)__
98 }
99 ^^A duckuments out-of-range >>>
100 \msg_new:nnnn { duckuments } { out-of-range }
101 {
102     You-requested-element~#3-out-of-the-range~#1-to~#2-of-array~'#4'.\\
103     I'll-just-use-element~#1-for-you.
104 }
105 {
106     \ ~-----\\
107     \ (Quack!~Nothing~here,~sorry.~Quack!)\\
108     \ ~~~~~\\
109     \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\string\ \\
110     \ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\~>()_\\
111     \ | ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~\ ~(__)__
112 }

```

```
\duckuments_patch_see_duckumentation:
```

```

113 \cs_new:Nn \duckuments_patch_see_duckumentation:
114     {%>>>
115         \cs_set:Nn \msg_see_documentation_text:n
116             {
117                 \\\
118                 See~the~\str_if_eq:nnTF { ##1 } { LaTeX } { LaTeX3~documentation }
119                 {
120                     \str_if_eq:nnTF { ##1 } { duckuments } { duckumentation }
121                     { ##1~documentation }
122                 }~for~further~information.
123             }
124     }%<<<
125 \duckuments_patch_see_duckumentation:

```

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

3.5 Options and Configurations

```

126 \keys_define:nn { duckuments }
127   {%>>>
128     ,immediate .bool_set:N = \l_duckuments_immediate_bool
129     ,immediate .default:n = true
130     ,full .code:n =
131       \duckuments_blindduck_range_test:n { 1-\c_duckuments_blindduck_pars_int }

```

```

132 ,maths .choice:
133 ,maths / both .code:n =
134 {
135   \bool_set_true:N \l_duckuments_math_inline_bool
136   \bool_set_true:N \l_duckuments_math_display_bool
137 }
138 ,maths / display .code:n = \bool_set_true:N \l_duckuments_math_display_bool
139 ,maths / inline .code:n = \bool_set_true:N \l_duckuments_math_inline_bool
140 ,maths / none .code:n =
141 {
142   \bool_set_false:N \l_duckuments_math_inline_bool
143   \bool_set_false:N \l_duckuments_math_display_bool
144 }
145 ,maths .default:n = both
146 ,toc .bool_set:N = \l_duckuments_toc_bool
147 ,toc .default:n = true
148 ,unknown .code:n =
149 { \msg_error:nnx { duckuments } { option~unknown } { \l_keys_key_tl } }
150 }%<<<
151 \ProcessKeysOptions { duckuments }
152 \keys_define:nn { duckuments }
153 {%>>>
154   unknown .code:n = \duckuments_blindduck_range_test:V \l_keys_key_tl
155 }%<<<
156 \bool_if:NTF \l_duckuments_immediate_bool
157 { \AtEndOfPackage { \duckuments_patch_includegraphics: } }
158 { \AtBeginDocument { \duckuments_patch_includegraphics: } }

```

3.6 Functions

3.6.1 Duckument Level

\duckument

```

159 \NewDocumentCommand \duckument { 0{ } }
160 {%>>>
161   \group_begin:
162   \keys_set:nn { duckuments } { #1 }
163   \bool_if:NT \l_duckuments_toc_bool { \tableofcontents }
164   \cs_if_exist_use:NT \chapter
165   { {\duckuments@headings@text{0}} \blindduck }
166   \duckuments@headings{1} \blindduck
167   \duckuments@headings{2} \blindduck
168   \duckuments@headings{3} \blindduck
169   \duckuments@headings{4} \blindduck
170   \section {Lists}
171   \duckuments_list_example:n { itemize }
172   \duckuments_list_example:n { enumerate }
173   \duckuments_list_example:n { description }
174   \group_end:
175 }%<<<

```

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

\blindduck

```

176 \NewDocumentCommand \blindduck { 0{ } }

```



```

177 {%>>>
178 \group_begin:
179 \keys_set:nn { duckuments } { #1 }
180 \duckuments@blindduck@text
181 \bool_if:NT \l_duckuments_blindduck_pars_bool { \par }
182 \group_end:
183 }%<<<

```

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

\ducklist

```

184 \NewDocumentCommand \ducklist {s m}
185 {%>>>
186 \begin{#2}
187 \IfBooleanTF { #1 }
188 { \ducklists@content@starred }
189 {
190 \str_if_eq:nnTF { #2 } { description }
191 \ducklists@content@starred
192 \ducklists@content
193 }
194 \end{#2}
195 }%<<<

```

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

\ducklistlist

```

196 \NewDocumentCommand \ducklistlist { s m }
197 {%>>>
198 \IfBooleanTF { #1 }
199 { \duckuments@listlist@starred { #2 } }
200 {
201 \str_if_eq:nnTF { #2 } { description }
202 { \duckuments@listlist@starred { description } }
203 { \duckuments@listlist{#2} }
204 }
205 }%<<<

```

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

\duckenumerate

```

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

```

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

\duckitemize

```

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

```

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

\duckdescription

```

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

```

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

\duckumentsCreateExampleFile

```
209 \newcommand*\duckumentsCreateExampleFile
210 {%>>>
211   \iow_new:N \duckuments_example_file_iow
212   \iow_open:Nn \duckuments_example_file_iow { example-image-duck.tex }
213   \iow_now:Nn \duckuments_example_file_iow
214     { \documentclass[tikz,multi]{standalone} }
215   \iow_now:Nn \duckuments_example_file_iow
216     { \usepackage{tikzducks} }
217   \iow_now:Nn \duckuments_example_file_iow
218     { \usepackage{duckuments} }
219   \iow_now:Nn \duckuments_example_file_iow
220     { \begin{document} }
221   \iow_now:Nn \duckuments_example_file_iow
222     { \duckumentsDrawRandomDucks }
223   \iow_now:Nn \duckuments_example_file_iow
224     { \end{document} }
225   \iow_close:N \duckuments_example_file_iow
226 }%<<<
```

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

\duckumentsDrawRandomDucks

```
227 \newcommand*\duckumentsDrawRandomDucks[1][\duckuments@randoms]
228 {%>>>
229   \foreach\x in {1,2,...,#1}
230     {\begin{tikzpicture}
231       \draw[black,fill=gray!50] (0,0) rectangle (6,4);
232       \draw[gray,thin] (0,0) -- (6,4);
233       \draw[gray,thin] (0,4) -- (6,0);
234       \draw[gray,thin] (3,0) -- (3,4);
235       \draw[gray,thin] (0,2) -- (6,2);
236       \node at (3,2) {\tikz\randuck;};
237     \end{tikzpicture}}
238 }%<<<
```

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

3.6.2 Intern

\duckuments@headings

```
239 \newcommand*\duckuments@headings[1]
240 {%>>>
241   \ifcase#1\relax
242     \expandafter\chapter
243   \or \expandafter\section
244   \or \expandafter\subsection
245   \or \expandafter\subsubsection
246   \or \expandafter\paragraph
247   \else \expandafter\@gobble
248   \fi
249   {\duckuments@headings@text{#1}}
250 }%<<<
```

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

`\duckuments@headings@level`

```
251 \newcommand*\duckuments@headings@level[1]
252 {%>>>
253 (
254 \ifcase#1
255 chapter
256 \or section
257 \or subsection
258 \or subsubsection
259 \or paragraph
260 \fi
261 )
262 }%<<<
```

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

`\duckuments@ifinline`

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

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

`\duckuments@ifdisplay`

```
265 \newcommand*\duckuments@ifdisplay[2] []
266 { \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`

```
267 \cs_new_protected_nopar:Nn \duckuments_list_example:n
268 {%>>>
269 \subsection{Example\ for\ ducks\ (#1)}
270 \ducklist { #1 }
271 \subsubsection{Nested\ ducks}
272 \ducklistlist { #1 }
273 }%<<<
```

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

`\duckuments@enquote`

```
274 \newcommand*\duckuments@enquote [1]
275 {%>>>
276 \cs_if_exist_use:NTF
277 \enquote { #1 }
278 { ‘ ‘ #1 ’ ’ }
279 }%<<<
```

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

`\duckuments_patch_includegraphics:`

```
280 \cs_new_protected_nopar:Nn \duckuments_patch_includegraphics:
281 {%>>>
282 \cs_if_exist:NT \includegraphics
283 {
284 \LetLtxMacro\duckuments@includegraphicsBAK\includegraphics
```

```

285 \RenewDocumentCommand \includegraphics
286 { >{\duckuments_starred:n}s 0{ } m }
287 {
288 \regex_match:NnTF \c_duckuments_example_regex { ##3 }
289 {
290 \duckuments_get_random_page:
291 \duckuments@includegraphicsBAK##1
292 [page=\duckuments_random_page:,##2]
293 { ##3 }
294 }
295 {
296 \duckuments@includegraphicsBAK##1[##2]{##3}
297 }
298 }
299 }
300 }%<<<

```

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

\duckuments_blindduck_range_test:n

```

301 \cs_new_protected:Nn \duckuments_blindduck_range_test:n
302 {%>>>
303 \regex_match:NnTF \c_duckuments_range_regex { #1 }
304 {
305 \seq_set_split:Nnn \l_duckuments_range_seq { - } { #1 }
306 \int_compare:nNnTF { 1 } = { \seq_count:N \l_duckuments_range_seq }
307 {
308 \cs_set:Npn \duckuments@blindduck@text
309 {
310 \duckuments_blindduck_single_par:n { #1 }
311 \int_gset:Nn \g_duckuments_blindduck_start_int { #1 + \c_one }
312 }
313 }
314 {
315 \bool_set_true:N \l_duckuments_blindduck_pars_bool
316 \exp_args:Nx
317 \tl_if_empty:nF { \seq_item:Nn \l_duckuments_range_seq { \c_one } }
318 {
319 \int_gset:Nn \g_duckuments_blindduck_start_int
320 { \seq_item:Nn \l_duckuments_range_seq { \c_one } }
321 }
322 \exp_args:Nx
323 \tl_if_empty:nTF { \seq_item:Nn \l_duckuments_range_seq { \c_two } }
324 {
325 \int_gset_eq:NN
326 \g_duckuments_blindduck_end_int
327 \g_duckuments_blindduck_start_int
328 }
329 {
330 \int_set:Nn \g_duckuments_blindduck_end_int
331 { \seq_item:Nn \l_duckuments_range_seq { \c_two } }
332 }
333 \duckuments_blindduck_set_text:xx
334 { \int_use:N \g_duckuments_blindduck_start_int }

```

```

335         { \int_use:N \g_duckuments_blindduck_end_int }
336     }
337 }
338 {
339     \group_begin:
340     \exp_args:NnnV
341     \msg_error:nnn { duckuments } { option-unknown } \l_keys_key_tl
342     \group_end:
343 }
344 }%<<<
345 \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

```

346 \cs_new:Nn \duckuments_blindduck_set_text:nn
347 { %>>>
348     \def \duckuments@blindduck@text
349     {
350         \int_step_function:nnnN { #1 } { \c_one } { #2 }
351         \duckuments_blindduck_par_loop:n
352         \int_gset:Nn \g_duckuments_blindduck_start_int { #2 + \c_one }
353     }
354 }%<<<
355 \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_single_par:n

```

356 \cs_new:Nn \duckuments_blindduck_single_par:n
357 { %>>>
358     \bool_if:nTF
359     {
360         \int_compare_p:nNn { #1 } > { \c_duckuments_blindduck_pars_int }
361         || \int_compare_p:nNn { #1 } < { \c_one }
362     }
363     {
364         \msg_error:nnxxxx { duckuments } { out-of-range }
365         { 1 } { \int_use:N \c_duckuments_blindduck_pars_int } { #1 }
366         { blindduck-paragraphs }
367         \duckuments@blindduck@text@i
368     }
369     {
370         \use:c { duckuments@blindduck@text@ \int_to_roman:n { #1 } }
371     }
372 }%<<<

```

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

\duckuments_blindduck_par_loop:n

```

373 \cs_new:Nn \duckuments_blindduck_par_loop:n
374 { %>>>
375     \duckuments_blindduck_single_par:n { #1 }
376     \par
377 }%<<<

```

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

\duckuments_starred:n

```

378 \cs_new_protected:Nn \duckuments_starred:n
379   {%>>>
380   \IfBooleanTF { #1 }
381     { \def\ProcessedArgument{*} }
382     { \def\ProcessedArgument{} }
383   }%<<<

```

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

\duckuments_get_random_page:

\duckuments_random_page:

```

384 \sys_if_engine_xetex:TF
385   {

```

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

```

386   \int_new:N \g_duckuments_RCiv_i_int
387   \int_new:N \g_duckuments_RCiv_j_int
388   \int_new:N \g_duckuments_RCiv_keylength_int
389   \int_new:N \g_duckuments_tmpa_int
390   \int_const:Nn \c_duckuments_RCiv_Slength_int { 256 }
391   \tl_new:N \l_duckuments_tmpa_tl
392   \tl_new:N \l_duckuments_tmpb_tl

```

Initialize the S array:

```

393   \cs_new_protected_nopar:Nn \duckuments_RCiv_S_new:n
394     { \int_new:c { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int } }
395   \cs_new_protected_nopar:Nn \duckuments_RCiv_S_set:nn
396     { \int_gset:cn { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int } { #2 } }
397   \cs_new_nopar:Nn \duckuments_RCiv_S_get:n
398     { \int_use:c { g_duckuments_RCiv_S_ \int_eval:n { #1 } _int } }
399   \cs_new_protected_nopar:Nn \duckuments_RCiv_key_new:n
400     { \int_new:c { g_duckuments_RCiv_key_ \int_eval:n { #1 } _int } }
401   \cs_new_protected_nopar:Nn \duckuments_RCiv_key_set:nn
402     { \int_gset:cn { g_duckuments_RCiv_key_ \int_eval:n { #1 } _int } { #2 } }
403   \cs_new_nopar:Nn \duckuments_RCiv_key_get:n
404     { \int_use:c { g_duckuments_RCiv_key_ \int_eval:n { #1 } _int } }
405   \int_step_inline:nnnn { 0 } { 1 } { 255 }
406     {
407       \duckuments_RCiv_S_new:n { #1 }
408       \duckuments_RCiv_S_set:nn { #1 } { #1 }
409     }
410   \int_step_inline:nnnn { 0 } { 1 } { 4 }
411     { \duckuments_RCiv_key_new:n { #1 } }
412   \duckuments_RCiv_key_set:nn { 0 } { \c_sys_minute_int }
413   \duckuments_RCiv_key_set:nn { 1 } { \c_sys_hour_int }
414   \duckuments_RCiv_key_set:nn { 2 } { \c_sys_day_int }
415   \duckuments_RCiv_key_set:nn { 3 } { \c_sys_month_int }
416   \duckuments_RCiv_key_set:nn { 4 }
417     { \int_mod:nn { \c_sys_year_int } { \c_duckuments_RCiv_Slength_int } }
418   \int_gset:Nn \g_duckuments_RCiv_keylength_int { 5 }
419   \str_map_inline:Nn \c_sys_jobname_str
420     {

```

```

421     \duckuments_RCiv_key_new:n { \g_duckuments_RCiv_keylength_int }
422     \duckuments_RCiv_key_set:nn
423     { \g_duckuments_RCiv_keylength_int }
424     { \int_from_alph:n { #1 } }
425     \int_gincr:N \g_duckuments_RCiv_keylength_int
426 }
427 \cs_new_protected_nopar:Nn \duckuments_swap_S_entries:nn
428 {
429     \int_set_eq:Nc
430     \g_duckuments_tmpa_int
431     { \g_duckuments_RCiv_S_ \int_eval:n { #1 } _int }
432     \int_set_eq:cc
433     { \g_duckuments_RCiv_S_ \int_eval:n { #1 } _int }
434     { \g_duckuments_RCiv_S_ \int_eval:n { #2 } _int }
435     \int_set_eq:cN
436     { \g_duckuments_RCiv_S_ \int_eval:n { #2 } _int }
437     \g_duckuments_tmpa_int
438 }
439 \int_gset:Nn \g_duckuments_RCiv_keylength_int { 5 }
440 \cs_new:Nn \duckuments_gadd_mod:Nnn
441 { \int_gset:Nn #1 { \int_mod:nn { #1 + ( #2 ) } { #3 } } }
442 \cs_new:Nn \duckuments_gadd_mod_Slength:Nn
443 {
444     \duckuments_gadd_mod:Nnn #1
445     { #2 } { \c_duckuments_RCiv_Slength_int }
446 }
447 \int_step_inline:nnnn { 0 } { 1 } { 255 }
448 {
449     \int_gset:Nn \g_duckuments_tmpa_int
450     { \int_mod:nn { #1 } { \g_duckuments_RCiv_keylength_int } }
451     \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_j_int
452     {
453         \duckuments_RCiv_S_get:n { #1 }
454         + \duckuments_RCiv_key_get:n { \g_duckuments_tmpa_int }
455     }
456     \duckuments_swap_S_entries:nn { #1 } { \g_duckuments_RCiv_j_int }
457 }
458 \int_gzero:N \g_duckuments_RCiv_i_int
459 \int_gzero:N \g_duckuments_RCiv_j_int

```

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

```

460 \cs_new_protected_nopar:Nn \duckuments_get_random_page:
461 {
462     \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_i_int { \c_one }
463     \duckuments_gadd_mod_Slength:Nn \g_duckuments_RCiv_j_int
464     { \duckuments_RCiv_S_get:n { \g_duckuments_RCiv_i_int } }
465     \duckuments_swap_S_entries:nn
466     { \g_duckuments_RCiv_i_int }
467     { \g_duckuments_RCiv_j_int }
468     \int_gset:Nn \g_duckuments_tmpa_int
469     { \duckuments_RCiv_S_get:n { \g_duckuments_RCiv_i_int } }
470     \duckuments_gadd_mod_Slength:Nn \g_duckuments_tmpa_int
471     { \duckuments_RCiv_S_get:n { \g_duckuments_RCiv_j_int } }

```

```

472     \cs_set:Nx \duckuments_random_page:
473     {
474         \int_eval:n
475         {
476             \int_mod:nn
477             { \duckuments_RCiv_S_get:n { \g_duckuments_tmpa_int } }
478             { \c_duckuments_example_pages_int }
479             + \c_one
480         }
481     }
482 }
483 \cs_new:Nn \duckuments_random_page: { 1 }
484 }

```

Both pdfTeX and LuaTeX don't need the RC4 as there \int_random:nn is available.

```

485 {
486     \cs_new:Nn \duckuments_get_random_page: {}
487     \cs_new:Nn \duckuments_random_page:
488     { \int_rand:nn { 1 } { \c_duckuments_example_pages_int } }
489 }

```

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

```

490 \ExplSyntaxOff

```

\duckuments@blindduck@text

```

491 \newcommand*\duckuments@blindduck@text{\duckuments@blindduck@text@i}
492 \newcommand*\duckuments@blindduck@text@i
493 {%>>>
494     There once was a very smart but sadly blind duck. When it was still a small
495     duckling it was renowned for its good vision. But sadly as the duck grew
496     older it caught a sickness which caused its eyesight to worsen. It became so
497     bad, that the duck couldn't read the notes it once took containing much of
498     inline math\duckuments@ifinline{ just like its favoured equation:  $d = u_c$ 
499     \cdot k$}. Only displayed equations remained legible%
500     \duckuments@ifdisplay[.] { so it could still read \begin{equation}d = r a^k
501     e\hbox{.}\end{equation}} That annoyed the smart duck, as it wasn't able to
502     do its research any longer. It called for its underduckling and said:
503     \duckuments@enquote{Go, find me the best eye ducktor there is. He shall
504     heal me from my disease!}%
505 }%<<<
506 \newcommand*\duckuments@blindduck@text@ii
507 {%>>>
508     \duckuments@enquote{But my duck, how are you supposed to manage your daily
509     routines without my visual guidance}, replied the underduckling. The smart
510     duck's face turned grim in anger. \duckuments@enquote{You dare to talk
511     back?} The underduckling blushed ashamed. How could he have objections
512     after his duck gave strict orders? The underduckling was so embarrassed
513     about his own behaviour he had to solve an equation.%
514 }%<<<
515 \newcommand*\duckuments@blindduck@text@iii
516 {%>>>
517     After the equation was solved and the underduckling prepared his leave for
518     the next day it fell asleep in a shaky mood. It did not know what the

```



```

519     journey had prepared for him and if he was prepared enough for it. His sleep
520     was restless. The dreams he had that night were not calm and bright as they
521     used to be for an innocent underduckling.%
522     }%<<<
523 \newcommand*\duckuments@blindduck@text@iv
524     {%>>>
525     Before the first sunlight the underduckling woke. He didn't have the feeling
526     of being well rested. But nonetheless he knew that this was the day he
527     should leave. Except saying goodbye to his beloved ones there was nothing
528     holding him back. His duck had sent him on the most important mission a five
529     weeks old inexperienced underduckling was ever sent. He bid farewell to his
530     mother, all his brothers and sisters, and finally from his duck. The bag was
531     shouldered, the boots were tied, the underduckling left.%
532     }%<<<

```

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

`\duckuments@headings@text`

```

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

```

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

`\ducklists@content`

```

535 \newcommand*\ducklists@content
536     {%>>>
537     \item First swims father drake
538     \item Then floats mother duck
539     \item After her paddles baby duckling
540     \item And over there bathes uncle canard
541     }%<<<

```

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

`\ducklists@content@starred`

```

542 \newcommand*\ducklists@content@starred
543     {%>>>
544     \item[drake] is the swimming father
545     \item[duck] is the floating mother
546     \item[duckling] is the paddling baby
547     \item[canard] is the bathing uncle
548     }%<<<

```

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

`\duckuments@listlist`

```

549 \newcommand*\duckuments@listlist[1]
550     {%>>>
551     \begin{#1}
552     \item swimming father drake
553     \begin{#1}
554     \item swimming father drake
555     \begin{#1}
556     \item swimming father drake
557     \begin{#1}

```

```

558         \item swimming father drake
559         \item floating mother duck
560     \end{#1}
561     \item floating mother duck
562 \end{#1}
563     \item floating mother duck
564 \end{#1}
565     \item floating mother duck
566 \end{#1}%
567 }%<<<

```

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

\duckuments@listlist@starred

```

568 \newcommand*\duckuments@listlist@starred[1]
569 {%>>>
570     \begin{#1}
571         \item[drake] is the swimming father
572         \begin{#1}
573             \item[drake] is the swimming father
574             \begin{#1}
575                 \item[drake] is the swimming father
576                 \begin{#1}
577                     \item[drake] is the swimming father
578                     \item[duck] is the floating mother
579                 \end{#1}
580                 \item[duck] is the floating mother
581             \end{#1}
582             \item[duck] is the floating mother
583         \end{#1}
584         \item[duck] is the floating mother
585     \end{#1}%
586 }%<<<

```

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

```

587 \endinput
588 \</pkg>

```

4 The story of the duck

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: "G"o, find me the best eye ducktor there is. He shall heal me from my disease!

"B"ut 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. "Y"ou 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.

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.

Before the first sunlight 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. 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.