

# Le paquet impnattypo

Raphaël Pinson  
raphink@gmail.com

1.6 en date du 2026/05/27

## 1 Introduction

En matière de typographie française, le *Lexique des règles typographiques en usage à l’Imprimerie Nationale* est une référence incontournable.

Si la majorité des recommandations de cet ouvrage est implémentée dans le module frenchb pour babel, certaines autres recommandations méritent encore d’être automatisées pour être implémentées en L<sup>A</sup>T<sub>E</sub>X.

C’est le but original de ce paquet, initié par une question sur le site tex.stackexchange.com<sup>1</sup>, et qui implémente plusieurs règles édictées dans ce lexique afin de les rendre plus facilement applicables aux textes édités avec L<sup>A</sup>T<sub>E</sub>X.

Au fur et à mesure que ce paquet a grandi, des fonctionnalités sont venues s’ajouter, dont certaines ne sont pas directement liées au lexique, mais améliorent la qualité typographique des documents.

## 2 Utilisation

Pour utiliser le paquet impnattypo, entrez la ligne :

```
\usepackage[<options>]{impnattypo}
```

Les options du paquet sont décrites dans les sections suivantes.

### 2.1 Césures

**hyphenation** En dehors des règles générales de coupure des mots, le lexique indique qu’il faut « [éviter] les coupures de mots sur plus de trois lignes consécutives. »

Afin de simplifier le code, l’implémentation proposée décourage fortement les césures en fin de page, ainsi que les césures sur deux lignes consécutives.

Pour activer cette fonctionnalité, utilisez l’option **hyphenation** :

```
\usepackage[hyphenation]{impnattypo}
```

## 2.2 Formatage des paragraphes

**parindent** Le lexique conseille une indentation des paragraphes de 1em. Ce réglage de `\parindent` peut être obtenu par l'utilisation de l'option `parindent`:

```
\usepackage[parindent]{impnattyto}
```

**lastparline** De plus, il est indiqué dans la section « Coupure des mots » que la dernière ligne d'un alinéa doit comporter un mot ou une fin de mot de longueur au moins égale au double du renforcement de l'alinéa suivant. À défaut d'implémenter exactement cette solution, l'option `lastparline` s'assure que la dernière ligne d'un alinéa est au moins aussi longue que le double de la valeur de `\parindent`.<sup>2</sup>

Lorsque LuaTeX est utilisé, la solution de Patrick Gundlach<sup>3</sup> est utilisée. Avec les autres moteurs de rendu, c'est la solution native de Enrico Gregorio<sup>4</sup> qui fait office d'implémentation. L'option `lastparlineminlength` permet de choisir, pour la dernière ligne de l'alinéa, une longueur minimale supérieure (mais pas inférieure) à sa valeur par défaut (`2\parindent`):

```
\usepackage[lastparline, lastparlineminlength=3em]{impnattyto}
```

Avec LuaTeX, la valeur passée à l'option `lastparlineminlength` est stockée dans la longueur `\lastparlineminlength`, qui peut être modifiée ultérieurement, dans le document.

Lorsque l'option `draft` est activée et que LuaTeX est utilisé, les espaces insécables insérés sont colorés en teal. La couleur utilisée peut être ajustée par l'option `lastparlinecolor`.

**nosingleletter** Il est également recommandé d'éviter les coupures isolant une lettre. La solution proposée par Patrick Gundlach<sup>5</sup> permet de remédier à cela en utilisant LuaTeX. Pour activer cette fonctionnalité, il faut utiliser l'option `nosingleletter`:

```
\usepackage[nosingleletter]{impnattyto}
```

Lorsque cette option est activée, seul LuaTeX (via la commande `lualatex`) pourra effectuer le rendu du document.

Lorsque l'option `draft` est activée, les espaces insécables insérés sont colorés en brown. La couleur utilisée peut être ajustée par l'option `nosinglelettercolor`.

**homeoarchy** Lorsque deux lignes consécutives commencent ou finissent par le même mot ou la même série de lettres, cela peut induire le lecteur en erreur et cela est donc à éviter.

La correction automatique de ce phénomène est très complexe et en général non souhaitable.<sup>6</sup> C'est pourquoi l'option `homeoarchy` de ce paquet se contente de les détecter

---

1. <http://tex.stackexchange.com/questions/20493/french-typography-recommendations>

2. <http://tex.stackexchange.com/questions/28357/ensure-minimal-length-of-last-line>

3. <http://tex.stackexchange.com/questions/28357/ensure-minimal-length-of-last-line/28361#28361>

4. <http://tex.stackexchange.com/questions/28357/ensure-minimal-length-of-last-line/28358#28358>

5. <http://tex.stackexchange.com/questions/27780/one-letter-word-at-the-end-of-line>

6. <http://tex.stackexchange.com/questions/27588/repetition-of-a-word-on-two-lines>

et de les afficher. Leur correction consistera en général en l'introduction d'une espace insécable dans le paragraphe :

```
\usepackage[homeoarchy]{impnatty}
```

Lorsque cette option est activée, seul LuaTeX (via la commande `lualatex`) pourra effectuer le rendu du document.

Cette option n'est effective que si l'option `draft` est activée.

Les espaces insécables insérées sont colorées de deux couleurs :

- Les mots entiers sont colorés en `red` et cette couleur peut être ajustée par l'option `homeoarchywordcolor` :
- Les mots partiels sont colorés en `orange` et cette couleur peut être ajustée par l'option `homeoarchycharcolor` :

Une séquence de glyphes est considérée comme problématique si :

- Le nombre de mots entiers trouvés dans les deux lignes consécutives est supérieur à 1. Ce paramètre peut être ajusté par l'option `homeoarchymaxwords` :
- Le nombre de caractères trouvés dans les deux lignes consécutives est supérieur à 3. Ce paramètre peut être ajusté par l'option `homeoarchymaxchars` :

`rivers` Une lézarde est un alignement vertical d'espaces dans un paragraphe. L'option `rivers` permet de colorer les lézardes afin de les identifier. Cette option ne corrige pas les lézardes détectées :

```
\usepackage[rivers]{impnatty}
```

Lorsque cette option est activée, seul LuaTeX (via la commande `lualatex`) pourra effectuer le rendu du document.

Cette option n'est effective que si l'option `draft` est activée.

Les espaces insécables insérées sont colorées en `lime`. Cette couleur peut être ajustée par l'option `riverscolor`.

## 2.3 Numérotation des chapitres

`frenchchapters` Concernant la numérotation des chapitres, le lexique indique : « Dans un titre, on compose en chiffres romains grandes capitales les numéros de chapitres, à l'exception de l'ordinal « premier » en toutes lettres malgré la tendance actuelle qui tend à lui substituer la forme cardinale Chapitre I. »

L'option `frenchchapters` du paquet implémente cette recommandation :

```
\usepackage[frenchchapters]{impnatty}
```

Si vous souhaitez bénéficier de la forme ordinale « premier » sans pour autant utiliser une numérotation des chapitres en chiffres romains, il est possible de redéfinir la macro `frenchchapter`, par exemple :

```
\let\frenchchapter\arabic % numérotation en chiffres arabes
\let\frenchchapter\babylonian % numérotation en chiffres babyloniens
```

## 2.4 Lignes orphelines et veuves

Il est fortement recommandé de ne pas laisser de lignes orphelines dans un document. Pour cela, nous vous conseillons d'utiliser le paquet `nowidow`:

```
\usepackage[all]{nowidow}
```

Voir la documentation de ce paquet pour plus d'options.

## 2.5 Mode brouillon

`draft` Le paquet `impnatty` dispose d'un mode brouillon permettant de visualiser les pénalités (espaces insécables) ajoutés par les options `nosingleletter` et `lastparline`, ainsi que les informations ajoutées par les options `homeoarchy` et `rivers`. En mode brouillon, les emplacements des espaces insécables insérés sont marqués par des rectangles de couleur.

Pour activer le mode brouillon, utilisez l'option `draft`, par exemple:

```
\usepackage[draft,lastparline]{impnatty}
```

Cet document est générée avec l'option `draft` afin d'en montrer les effets.

## 3 Implementation

```
1 \ProvidesPackage{impnatty}
2 \RequirePackage{ifluatex}
3 \RequirePackage{kvoptions}
4 \SetupKeyvalOptions{
5   family=impnatty,
6   prefix=int,
7 }
8 \DeclareBoolOption{draft}
9 \DeclareBoolOption{frenchchapters}
10 \DeclareBoolOption{hyphenation}
11 \DeclareBoolOption{nosingleletter}
12 \DeclareBoolOption{parindent}
13 \DeclareBoolOption{lastparline}
14 \DeclareBoolOption{homeoarchy}
15 \DeclareBoolOption{rivers}
16 \DeclareStringOption[red]{homeoarchywordcolor}
17 \DeclareStringOption[orange]{homeoarchycharcolor}
18 \DeclareStringOption[brown]{nosinglelettercolor}
19 \DeclareStringOption[teal]{lastparlinecolor}
20 \DeclareStringOption[lime]{riverscolor}
21 \DeclareStringOption[1]{homeoarchymaxwords}
22 \DeclareStringOption[3]{homeoarchymaxchars}
23 \DeclareStringOption[Opt]{lastparlineminlength}
24 \ProcessKeyvalOptions*
25 \RequirePackage{xcolor}
26 \def\usecolor#1{\csname\string\color@#1\endcsname\space}
27 \ifinthyphenation
```

No page finishes with an hyphenated word

Discourage hyphenation on  
two lines in a row  
Number chapters

No single letter

```

28 \brokenpenalty=10000
29 \doublehyphendemerits=1000000000
30 \fi

31 \ifintfrenchchapters
32 \let\frenchchapter\Roman
33 \renewcommand{\thechapter}{%
34 \ifnum\value{chapter}=1
35 premier%
36 \else
37 \frenchchapter{chapter}%
38 \fi
39 }
40 \fi

41 \ifintnosingleletter
42 \ifluatex
43 \RequirePackage{luatexbase,luacode}
44 \begin{luacode}
45 local glyph_id = node.id "glyph"
46 local glue_id = node.id "glue"
47 local hlist_id = node.id "hlist"
48
49 local prevent_single_letter = function (head)
50 while head do
51 if head.id == glyph_id then -- glyph
52 if unicode.utf8.match(unicode.utf8.char(head.char),"%a") then -- some kind of l
53 if head.prev.id == glue_id and head.next.id == glue_id then -- only
54
55 local p = node.new("penalty")
56 p.penalty = 10000
57
58 \ifintdraft
59 local w = node.new("whatsit","pdf_literal")
60 w.data = "q \usecolor{\intnosinglelettercolor} 0 0 m 0 5 1 2 5 1 2 0 1 b
61
62 node.insert_after(head,head,w)
63 node.insert_after(head,w,p)
64 \else
65 node.insert_after(head,head,p)
66 \fi
67 end
68 end
69 end
70 head = head.next
71 end
72 return true
73 end
74
75 luatexbase.add_to_callback("pre_linebreak_filter",prevent_single_letter,"~")
76 \end{luacode}
77 \else
78 \PackageError{The nosingleletter option only works with LuaTeX}
79 \fi
80 \fi

```

Paragraph indentation

Last line of paragraph

Detect homeoarchies

```
81 \ifintparindent
82 \setlength{\parindent}{1em}
83 \fi

84 \ifintlastparline
85   \ifluatex
86     \newlength{\lastparlineminlength}
87     \setlength{\lastparlineminlength}{\intlastparlineminlength}
88     \RequirePackage{luatexbase,luacode}
89     \begin{luacode}
90       local glyph_id = node.id "glyph"
91       local glue_id = node.id "glue"
92       local hlist_id = node.id "hlist"
93
94       last_line_minimal_length = function (head)
95         while head do
96           local _w,_h,_d = node.dimensions(head)
97           if head.id == glue_id and head.subtype ~= 15
98             and (_w < 2 * tex.parindent or _w < tex.skip['lastparlineminlength'].width)
99           then
100
101             -- we are at a glue and have less then 2*\parindent of \lastparlineminlength to
102             local p = node.new("penalty")
103             p.penalty = 10000
104
105             \ifintdraft
106               local w = node.new("whatsit","pdf_literal")
107               w.data = "q \usecolor{\intlastparlinecolor} 0 0 m 0 5 1 2 5 1 2 0 1 b Q"
108
109               node.insert_after(head,head.prev,w)
110               node.insert_after(head,w,p)
111             \else
112               node.insert_after(head,head.prev,p)
113             \fi
114           end
115
116           head = head.next
117         end
118         return true
119       end
120
121       luatexbase.add_to_callback("pre_linebreak_filter",last_line_minimal_length,"lastparline")
122     \end{luacode}
123   \else
124     \setlength{\@tempskipa}{\intlastparlineminlength}
125     \ifdim \@tempskipa < 2\parindent
126       \setlength{\@tempskipa}{2\parindent}
127     \fi
128     \setlength{\parfillskip}{0pt plus\dimexpr\textwidth-\@tempskipa}
129   \fi
130 \fi

131 \ifinthomeoarchy
132 \ifintdraft
```

```

133 \ifluatex
134   \RequirePackage{luatexbase,luacode}
135   \begin{luacode}
136     local glyph_id = node.id "glyph"
137     local glue_id = node.id "glue"
138     local hlist_id = node.id "hlist"
139
140     compare_lines = function (line1,line2)
141       local head1 = line1.head
142       local head2 = line2.head
143
144       local char_count = 0
145       local word_count = 0
146
147       while head1 and head2 do
148         if (head1.id == glyph_id and head2.id == glyph_id
149             and head1.char == head2.char) -- identical glyph
150         or (head1.id == glue_id and head2.id == glue_id) then -- glue
151
152           if head1.id == glyph_id then -- glyph
153             char_count = char_count + 1
154           elseif char_count > 0 and head1.id == glue_id then -- glue
155             word_count = word_count + 1
156           end
157           head1 = head1.next
158           head2 = head2.next
159         elseif (head1.id == 0 or head2.id == 0) then -- end of line
160           break
161         elseif (head1.id ~= glyph_id and head1.id ~= glue_id) then -- some other kind of node
162           head1 = head1.next
163         elseif (head2.id ~= glyph_id and head2.id ~= glue_id) then -- some other kind of node
164           head2 = head2.next
165         else -- no match, no special node
166           break
167         end
168       end
169       -- analyze last non-matching node, check for punctuation
170       if ((head1 and head1.id == glyph_id and head1.char > 49)
171           or (head2 and head2.id == glyph_id and head2.char > 49)) then
172         -- not a word
173       elseif char_count > 0 then
174         word_count = word_count + 1
175       end
176       return char_count,word_count,head1,head2
177     end
178
179     compare_lines_reverse = function (line1,line2)
180       local head1 = node.tail(line1.head)
181       local head2 = node.tail(line2.head)
182
183       local char_count = 0
184       local word_count = 0
185
186       while head1 and head2 do

```

```

187         if (head1.id == glyph_id and head2.id == glyph_id
188             and head1.char == head2.char)          -- identical glyph
189             or (head1.id == glue_id and head2.id == glue_id) then -- glue
190
191             if head1.id == glyph_id then -- glyph
192                 char_count = char_count + 1
193             elseif char_count > 0 and head1.id == glue_id then -- glue
194                 word_count = word_count + 1
195             end
196             head1 = head1.prev
197             head2 = head2.prev
198         elseif (head1.id == 0 or head2.id == 0) then -- start of line
199             break
200         elseif (head1.id ~= glyph_id and head1.id ~= glue_id) then -- some other kind of node
201             head1 = head1.prev
202         elseif (head2.id ~= glyph_id and head2.id ~= glue_id) then -- some other kind of node
203             head2 = head2.prev
204         elseif (head1.id == glyph_id and head1.char < 48) then -- punctuation
205             head1 = head1.prev
206         elseif (head2.id == glyph_id and head2.char < 48) then -- punctuation
207             head2 = head2.prev
208         else -- no match, no special node
209             break
210         end
211     end
212     -- analyze last non-matching node, check for punctuation
213     if ((head1 and head1.id == glyph_id and head1.char > 49)
214         or (head2 and head2.id == glyph_id and head2.char > 49)) then
215         -- not a word
216     elseif char_count > 0 then
217         word_count = word_count + 1
218     end
219     return char_count, word_count, head1, head2
220 end
221
222 highlight = function (line, nend, color)
223     local n = node.new("whatsit", "pdf_literal")
224
225     -- get dimensions
226     local w, h, d = node.dimensions(line.head, nend)
227     local w_pts = w/65536 -- scaled points to points
228
229     -- set data
230     n.data = "q " .. color .. " 0 0 m 0 5 1 " .. w_pts .. " 5 1 " .. w_pts .. " 0 1 b Q"
231
232     -- insert node
233     n.next = line.head
234     line.head = n
235     node.slide(line.head)
236 end
237
238 highlight_reverse = function (nstart, line, color)
239     local n = node.new("whatsit", "pdf_literal")
240

```

```

241
242      -- get dimensions
243      local w,h,d = node.dimensions(nstart,node.tail(line.head))
244      local w_pts = w/65536 -- scaled points to points
245
246      -- set data
247      n.data = "q " .. color .. " 0 0 m 0 5 1 " .. w_pts .. " 5 1 " .. w_pts .. " 0 1 b Q"
248
249      -- insert node
250      node.insert_after(line.head,nstart,n)
251  end
252
253  homeoarchy = function (head)
254      local cur_line = head
255      local prev_line -- initiate prev_line
256
257      local max_char = tonumber(\inhomeoarchymaxchars)
258      local max_word = tonumber(\inhomeoarchymaxwords)
259
260      while head do
261          if head.id == hlist_id then -- new line
262              prev_line = cur_line
263              cur_line = head
264              if prev_line.id == hlist_id then
265                  -- homeoarchy
266                  char_count,word_count,prev_head,cur_head = compare_lines(prev_line,cur_line)
267                  if char_count >= max_char or word_count >= max_word then
268                      local color
269                      if word_count >= max_word then
270                          color = "q \usecolor{\inhomeoarchywordcolor}"
271                      else
272                          color = "q \usecolor{\inhomeoarchycharcolor}"
273                      end
274
275                      -- highlight both lines
276                      highlight(prev_line,prev_head,color)
277                      highlight(cur_line,cur_head,color)
278                  end
279              end
280          end
281          head = head.next
282      end
283      return true
284  end
285
286  luatexbase.add_to_callback("post_linebreak_filter",homeoarchy,"homeoarchy")
287
288  homoioteleuton = function (head)
289      local cur_line = head
290      local prev_line -- initiate prev_line
291
292      local max_char = tonumber(\inhomeoarchymaxchars)
293      local max_word = tonumber(\inhomeoarchymaxwords)
294

```

```

295     local linecounter = 0
296
297     while head do
298         if head.id == hlist_id then -- new line
299             linecounter = linecounter + 1
300             if linecounter > 1 then
301                 prev_line = cur_line
302                 cur_line = head
303                 if prev_line.id == hlist_id then
304                     -- homoioteleuton
305                     char_count, word_count, prev_head, cur_head = compare_lines_reverse(prev_line,
306                     if char_count >= max_char or word_count >= max_word then
307                         local color
308                         if word_count >= max_word then
309                             color = "q \usecolor{\inthomeoarchywordcolor}"
310                         else
311                             color = "q \usecolor{\inthomeoarchycharcolor}"
312                         end
313
314                         -- highlight both lines
315                         highlight_reverse(prev_head, prev_line, color)
316                         highlight_reverse(cur_head, cur_line, color)
317                     end
318                 end
319             end
320         end
321         head = head.next
322     end
323
324     return true
325 end
326
327 luatexbase.add_to_callback("post_linebreak_filter", homoioteleuton, "homoioteleuton")
328 \end{luacode}
329 \else
330 \PackageError{The homeoarchy option only works with LuaTeX}
331 \fi
332 \fi
333 \fi
334 \ifintrivers
335 \ifintdraft
336 \ifluatex
337 \RequirePackage{luatexbase,luacode}
338 \begin{luacode}
339 local glyph_id = node.id "glyph"
340 local glue_id = node.id "glue"
341 local hlist_id = node.id "hlist"
342
343 river_analyze_line = function(line, dim1, dim2, precision)
344     local head = line.head
345
346     while head do
347         if head.id == glue_id then -- glue node
348             local w1, h1, d1 = node.dimensions(line.glue_set, line.glue_sign, line.glue_order, line.h

```

Detect rivers

```

349         local w2,h2,d2 = node.dimensions(line.glue_set,line.glue_sign,line.glue_order,line.h
350         --print("dim1: "..dim1.." ; dim2: "..dim2.." ; w1: "..w1.." ; w2: "..w2)
351         if w1 > dim2 + precision then -- out of range
352             return false,head
353         elseif w1 < (dim2 + precision) and w2 > (dim1 - precision) then -- found
354             return true,head
355         end
356     end
357     head = head.next
358 end
359
360 return false,head
361 end
362
363 rivers = function (head)
364     local prev_prev_line
365     local prev_line
366     local cur_line = head
367     local cur_node
368     local char_count
369
370     local linecounter = 0
371
372     while head do
373         if head.id == hlist_id then -- new line
374             linecounter = linecounter + 1
375             prev_prev_line = prev_line
376             prev_line = cur_line
377             cur_line = head
378             if linecounter > 2 then
379                 cur_node = cur_line.head
380                 char_count = 0
381
382                 while cur_node do
383                     if cur_node.id == glyph_id then -- glyph
384                         char_count = char_count + 1
385                     elseif cur_node.id == glue_id and char_count > 0 and cur_node.next then -- gl
386                         -- prev_line
387                         local w1,h1,d1 = node.dimensions(head.glue_set,head.glue_sign,head.glue_ord
388                         local w2,h2,d2 = node.dimensions(head.glue_set,head.glue_sign,head.glue_ord
389                         -- if we allow up to 45° diagonal rivers, then there can be up to + or - li
390                         local w_p,h_p,d_p = node.dimensions(prev_line.head,cur_line.head) -- calcul
391                         found_p,head_p = river_analyze_line(prev_line,w1,w2,h_p)
392
393                     if found_p then
394                         -- prev_prev_line
395                         local w1,h1,d1 = node.dimensions(prev_line.glue_set,prev_line.glue_sign,
396                         local w2,h2,d2 = node.dimensions(prev_line.glue_set,prev_line.glue_sign,
397                         -- if we allow up to 45° diagonal rivers, then there can be up to + or -
398                         local w_p,h_p,d_p = node.dimensions(prev_prev_line.head,prev_line.head)
399                         found_pp,head_pp = river_analyze_line(prev_prev_line,w1,w2,h_p)
400
401                     if found_pp then
402                         local n_pp = node.new("whatsit","pdf_literal")

```

```

403         n_pp.data = "q \usecolor{\intriverscolor} 0 0 m 0 5 1 5 5 1 5 0 1 b Q"
404         node.insert_after(prev_prev_line,head_pp.prev,n_pp)
405
406         local n_p = node.new("whatsit","pdf_literal")
407         n_p.data = "q \usecolor{\intriverscolor} 0 0 m 0 5 1 5 5 1 5 0 1 b Q"
408         node.insert_after(prev_line,head_p.prev,n_p)
409
410         local n_c = node.new("whatsit","pdf_literal")
411         n_c.data = "q \usecolor{\intriverscolor} 0 0 m 0 5 1 5 5 1 5 0 1 b Q"
412         node.insert_after(cur_line,cur_node.prev,n_c)
413     end
414 end
415 end
416     cur_node = cur_node.next
417 end
418 end
419     end
420     head = head.next
421 end
422
423 return true
424
425 end
426
427
428 luatexbase.add_to_callback("post_linebreak_filter",rivers,"rivers")
429     \end{luacode}
430 \else
431     \PackageError{The homeoarchy option only works with LuaTeX}
432 \fi
433 \fi
434 \fi

```

## Change History

|     |                                                 |     |                                                              |
|-----|-------------------------------------------------|-----|--------------------------------------------------------------|
| 0.1 | General : First version                         | 0.7 | General : Add homoioteleuton detection                       |
| 0.2 | General : Add nosingleletter option             | 0.8 | General : Add river detection                                |
| 0.3 | General : Add parindent and lastparline options | 0.9 | General : River detection returns false by default           |
| 0.4 | General : Add draft mode                        | 1.0 | General : Improve documentation, simplify internal variables |
| 0.5 | General : Add homeoarchy detection              | 1.1 | General : Fix French documentation                           |
| 0.6 | General : Words contain at least one character  | 1.2 | General : Fix French documentation                           |

|                                            |                                                                                                                            |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 1.3                                        | 1.5                                                                                                                        |
| General : Fix French documentation . . . 1 | General : Fix support for TexLive 2016<br>(new <code>luatex</code> compatibility). Thanks<br>to Michal Hoftich . . . . . 1 |
| 1.4                                        | 1.6                                                                                                                        |
| General : Fix release date . . . . . 1     | General : Add <code>lastparlineminlength</code><br>option . . . . . 1                                                      |

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

| Symbols                                                                                                                                                                                                                                                                                                     | H                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | N                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\@tempskipa</code> . . . . .<br>... 124, 125, 126, 128                                                                                                                                                                                                                                                | <code>\homeoarchy</code> . . . . . 2<br><code>\hyphenation</code> . . . . . 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <code>\newlength</code> . . . . . 86<br><code>\nosingleletter</code> . . . . . 2                                                                                                                                                                 |
| B                                                                                                                                                                                                                                                                                                           | I                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | P                                                                                                                                                                                                                                                |
| <code>\begin</code> . . . . . 44, 89, 135, 338<br><code>\brokenpenalty</code> . . . . . 28                                                                                                                                                                                                                  | <code>\ifdim</code> . . . . . 125<br><code>\ifintdraft</code> 58, 105, 132, 335<br><code>\ifintfrenchchapters</code> 31<br><code>\ifinthomeoarchy</code> . . . 131<br><code>\ifinthyphenation</code> . . . 27<br><code>\ifintlastparline</code> . . . 84<br><code>\ifintnosingleletter</code> 41<br><code>\ifintparindent</code> . . . . 81<br><code>\ifintrivers</code> . . . . . 334<br><code>\ifluatex</code> . . . 42, 85, 133, 336<br><code>\ifnum</code> . . . . . 34<br><code>\inthomeoarchycharcolor</code><br>... 272, 311<br><code>\inthomeoarchymaxchars</code><br>... 257, 292<br><code>\inthomeoarchymaxwords</code><br>... 258, 293<br><code>\inthomeoarchywordcolor</code><br>... 270, 309<br><code>\intlastparlinecolor</code> 107<br><code>\intlastparlineminlength</code><br>... 87, 124<br><code>\intnosinglelettercolor</code> .<br>... 60<br><code>\intriverscolor</code> . . .<br>... 403, 407, 411 | <code>\PackageError</code> 78, 330, 431<br><code>\parfillskip</code> . . . . . 128<br><code>\parindent</code> . . . . .<br>... 2, 82, 101, 125, 126<br><code>\ProcessKeyvalOptions</code> .<br>... 24<br><code>\ProvidesPackage</code> . . . . 1 |
| C                                                                                                                                                                                                                                                                                                           | R                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | S                                                                                                                                                                                                                                                |
| <code>\color@</code> . . . . . 26<br><code>\csname</code> . . . . . 26                                                                                                                                                                                                                                      | <code>\renewcommand</code> . . . . . 33<br><code>\RequirePackage</code> . . . 2,<br>3, 25, 43, 88, 134, 337<br><code>\rivers</code> . . . . . 3<br><code>\Roman</code> . . . . . 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <code>\setlength</code> . . . . .<br>... 82, 87, 124, 126, 128<br><code>\SetupKeyvalOptions</code> . . 4<br><code>\space</code> . . . . . 26<br><code>\string</code> . . . . . 26                                                                |
| D                                                                                                                                                                                                                                                                                                           | T                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | U                                                                                                                                                                                                                                                |
| <code>\DeclareBoolOption</code> 8,<br>9, 10, 11, 12, 13, 14, 15<br><code>\DeclareStringOption</code><br>... 16,<br>17, 18, 19, 20, 21, 22, 23<br><code>\def</code> . . . . . 26<br><code>\dimexpr</code> . . . . . 128<br><code>\doublehyphendemerits</code> .<br>... 29<br><code>\draft</code> . . . . . 4 | <code>\textwidth</code> . . . . . 128<br><code>\thechapter</code> . . . . . 33                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <code>\usecolor</code> . . . . . 26,<br>60, 107, 270, 272,<br>309, 311, 403, 407, 411                                                                                                                                                            |
| E                                                                                                                                                                                                                                                                                                           | V                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                  |
| <code>\else</code> . . . . . 36, 64,<br>77, 111, 123, 329, 430<br><code>\end</code> . . . . . 76, 122, 328, 429<br><code>\endcsname</code> . . . . . 26                                                                                                                                                     | <code>\value</code> . . . . . 34                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                  |
| F                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                  |
| <code>\fi</code> . . . . . 30, 38, 40,<br>66, 79, 80, 83, 113,<br>127, 29, 30, 331,<br>332, 333, 432, 433, 434<br><code>\frenchchapter</code> . . . 32, 37<br><code>\frenchchapters</code> . . . . . 3                                                                                                      | L                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                  |
|                                                                                                                                                                                                                                                                                                             | <code>\lastparline</code> . . . . . 2<br><code>\lastparlineminlength</code><br>... 86, 87, 101<br><code>\let</code> . . . . . 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                  |