

Le paquet `impnattypo`

Raphaël Pinson
`raphink@gmail.com`

0.3 en date du 2011/09/13

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 concerne la ponctuation, l'usage des petites capitales ou encore l'ordre des parties d'un livre, certaines autres recommandations méritent d'être automatisées pour être implémentées en \LaTeX .

C'est le but de ce paquet, qui implémente plusieurs règles édictées dans ce lexique afin de les rendre plus facilement applicables aux textes édités avec \LaTeX .

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 ».

En faisant un peu de zèle, 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]{imnatty}
```

`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`.¹

Lorsque Lua \TeX est utilisé, la solution de Patrick Gundlach² est utilisée. Avec les autres moteurs de rendu, c'est la solution native de Enrico Gregorio³ qui fait office d'implémentation :

```
\usepackage[lastparline]{imnatty}
```

`nosingletter` Il est également recommandé d'éviter les coupures isolant une lettre. La solution proposée par Patrick Gundlach⁴ permet de remédier à cela en utilisant Lua \TeX . Pour activer cette fonctionnalité, il faut utiliser l'option `nosingletter` :

```
\usepackage[nosingletter]{imnatty}
```

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

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]{imnatty}
```

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
```

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

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

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

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

2.4 Lignes orphelines

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.

3 Implémentation

No page finishes with an hyphenated word
Discourage hyphenation on two lines in a row
Number chapters

```
1 \ProvidesPackage{impnatty}
2 \usepackage{ifluatex}
3 \newif\if@impnatty@frenchchapters
4 \DeclareOption{frenchchapters}{\@impnatty@frenchchapterstrue}
5 \newif\if@impnatty@hyphenation
6 \DeclareOption{hyphenation}{\@impnatty@hyphenationtrue}
7 \newif\if@impnatty@nosingleletter
8 \DeclareOption{nosingleletter}{\@impnatty@nosinglelettertrue}
9 \newif\if@impnatty@parindent
10 \DeclareOption{parindent}{\@impnatty@parindenttrue}
11 \newif\if@impnatty@lastparline
12 \DeclareOption{lastparline}{\@impnatty@lastparlinetrue}
13 \ProcessOptions
14 \if@impnatty@hyphenation
15   \brokenpenalty=10000
16   \doublehyphendemerits=1000000000
17 \fi
18 \if@impnatty@frenchchapters
19   \let\frenchchapter\Roman
20   \renewcommand{\thechapter}{%
21     \ifnum\value{chapter}=1
22       premier%
23     \else
24       \frenchchapter{chapter}%
25     \fi
26   }
27 \fi
28 \if@impnatty@nosingleletter
29   \ifluatex
30     \usepackage{luatexbase,luacode}
31     \begin{luacode}
32
33       local prevent_single_letter = function (head)
34         while head do
35           if head.id == 37 then -- glyph
36             if head.prev.id == 10 and head.next.id == 10 then -- only if we are at a one letter
```

No single letter

```

37
38     local p = node.new("penalty")
39     p.penalty = 10000
40
41     -- Uncomment the next lines to show the position of the ~'s
42     -- local p = node.new("rule")
43     -- p.width = 2* 2^16
44     -- p.height = 2^16
45     -- p.depth = 2^16
46
47     p.next = head.next
48     head.next.prev = p
49     head.next = p
50
51     end
52 end
53 head = head.next
54 end
55 return true
56 end
57
58     luatexbase.add_to_callback("pre_linebreak_filter",prevent_single_letter,"~")
59     \end{luacode}
60 \else
61     \PackageError{The nosingleletter option only works with LuaTeX}
62 \fi
63 \fi
64 \if@impnatty@parindent
65 \setlength{\parindent}{1em}
66 \fi
67 \if@impnatty@lastparline
68 \ifluatex
69     \usepackage{luatexbase,luacode}
70     \begin{luacode}
71     last_line_twice_parindent = function (head)
72     while head do
73         local _w,_h,_d = node.dimensions(head)
74         if head.id == 10 and head.subtype ~= 15 and (_w < 2 * tex.parindent) then
75
76             -- we are at a glue and have less then 2*\parindent to go
77             local p = node.new("penalty")
78             p.penalty = 10000
79             p.next = head
80             head.prev.next = p
81             p.prev = head.prev
82             head.prev = p
83         end
84
85         head = head.next

```

Paragraph indentation

Last line of paragraph

```
86     end
87     return true
88 end
89
90     luatexbase.add_to_callback("pre_linebreak_filter",last_line_twice_parindent,"lastparline"
91     \end{luacode}
92 \else
93     \setlength{\parfillskip}{0pt plus\dimexpr\textwidth-2\parindent}
94 \fi
95 \fi
```