Le paquet impnattypo

Raphaël Pinson
raphink@gmail.com

0.4 en date du 2011/09/19

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 recommendations de cet ouvrage est implémentée dans le module frenchb pour babel, certaines autres recommendations méritent encore d'être automatisées pour être implémentées en LaTeX.

C'est le but de ce paquet, initié par une question sur le site tex.stackexchange.com [1], et 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 fonctionalité, 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]{impnattypo}
```

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 renfoncement 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`. [2]

Lorsque LuaTEXest utilisé, la solution de Patrick Gundlach [3] est utilisée. Avec les autres moteurs de rendu, c'est la solution native de Enrico Gregorio [4] qui fait office d'implémentation:

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

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

nosingleletter

Il est également recommendé d'éviter les coupures isolant une lettre. La solution proposée par Patrick Gundlach [5] permet de remédier à cela en utilisant LuaTEX. Pour activer cette fonctionalité, il faut utiliser l'option `nosingleletter`:

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

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 `nosingleletterecolor`.

homeoarchy

Lorsque deux lignes consécutives commencent 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. C'est pourquoi l'option `homeoarchy` de ce paquet se contente de les détecter et de les afficher. Leur correction consistera en général en l'introduction d'un espace insécable dans le paragraphe.

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

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

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

Les espaces insécables insérés sont colorés 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 considéré comme problématique si :

- Le nombre de mots entiers matchant est supérieur à 1. Ce paramètre peut être ajusté par l'option `homeoarchymaxwords` ;
- Le nombre de caractères matchant est supérieur à 3. Ce paramètre peut être ajusté par l'option `homeoarchymaxchars` ;

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 recommendation :

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

Si vous souhaitez bénéficier du 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

Il est fortement recommendé 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 impnattypo dispose d'un mode brouillon permettant de visualiser les pénalités (espaces insécables) ajoutés par les options `nosingleletter` et `lastparline`. 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]{impnattypo}
```

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

3 Implémentation

```
\ProvidesPackage{impnattypo}
\RequirePackage{ifluatex}
\RequirePackage{kvoptions}
\SetupKeyvalOptions{
   family=impnattypo,
   prefix=impnattypo,
}
\DeclareBoolOption{draft}
\DeclareBoolOption{frenchchapters}
\DeclareBoolOption{hyphenation}
\DeclareBoolOption{nosingleletter}
\DeclareBoolOption{parindent}
\DeclareBoolOption{lastparline}
\DeclareBoolOption{homeoarchy}
\DeclareStringOption[red]{homeoarchywordcolor}
\DeclareStringOption[orange]{homeoarchycharcolor}
\DeclareStringOption[brown]{nosingleletterscolor}
\DeclareStringOption[teal]{lastparlinecolor}
\DeclareStringOption[1]{homeoarchymaxwords}
\DeclareStringOption[3]{homeoarchymaxchars}
\ProcessKeyvalOptions*
\RequirePackage{xcolor}
\def\usecolor#1{\csname\string\color@#1\endcsname\space}
```

No page finishes with an hyphenated word

Discourage hyphenation on two lines in a row

Number chapters

```
\ifimpnattypohyphenation
   \brokenpenalty=10000
   \doublehyphendemerits=1000000000
\fi
\ifimpnattypofrenchchapters
   \let\frenchchapter\Roman
   \renewcommand{\thechapter}{%
     \ifnum\value{chapter}=1
       premier%
     \else
       \frenchchapter{chapter}%
     \fi
   }
\fi
```

No single letter

```
\ifimpnattyponosingleletter
   \ifluatex
      \RequirePackage{luatexbase,luacode}
      \begin{luacode}

      local prevent_single_letter = function (head)
```

```
        while head do
          if head.id == 37 then
            if (head.char >= 65 and head.char <= 122) or (head.char >= 192 and head.char <= 255
               if head.prev.id == 10 and head.next.id == 10 then

                 local p = node.new("penalty")
                 p.penalty = 10000

                 \ifimpnattypodraft
                    local w = node.new("whatsit","pdf_literal")
                    w.data = "q \usecolor{\impnattyponosinglelettercolor} 0 0 m 0 5 l 2 5 l 2 0

                    node.insert_after(head,head,w)
                    node.insert_after(head,w,p)
                 \else
                    node.insert_after(head,head,p)
                 \fi
               end
            end
          end
          head = head.next
        end
        return true
      end

      luatexbase.add_to_callback("pre_linebreak_filter",prevent_single_letter,"~")
      \end{luacode}
   \else
      \PackageError{The nosingleletter option only works with LuaTeX}
   \fi
\fi
```

Paragraph indentation

```
\ifimpnattypoparindent
\setlength{\parindent}{1em}
\fi
```

Last line of paragraph

```
\ifimpnattypolastparline
   \ifluatex
      \RequirePackage{luatexbase,luacode}
      \begin{luacode}
      last_line_twice_parindent = function (head)
        while head do
          local _w,_h,_d = node.dimensions(head)
          if head.id == 10 and head.subtype ~= 15 and (_w < 2 * tex.parindent) then

              -- we are at a glue and have less then 2*\parindent to go
              local p = node.new("penalty")
              p.penalty = 10000

              \ifimpnattypodraft
                 local w = node.new("whatsit","pdf_literal")
```

```
                w.data = "q \usecolor{\impnattypolastparlinecolor} 0 0 m 0 5 l 2 5 l 2 0 l b Q
                node.insert_after(head,head.prev,w)
                node.insert_after(head,w,p)
             \else
                node.insert_after(head,head.prev,p)
             \fi
          end

          head = head.next
        end
        return true
      end

      luatexbase.add_to_callback("pre_linebreak_filter",last_line_twice_parindent,"lastparline"
      \end{luacode}
   \else
      \setlength{\parfillskip}{0pt plus\dimexpr\textwidth-2\parindent}
   \fi
\fi
```

Detect homeoarchies

```
\ifimpnattypohomeoarchy
 \ifimpnattypodraft
   \ifluatex
      \RequirePackage{luatexbase,luacode}
      \begin{luacode}
      compare_lines = function (line1,line2)
        local head1 = line1.head
        local head2 = line2.head

        local char_count = 0
        local word_count = 0

        while head1 and head2 do
           if (head1.id == 37 and head2.id == 37
                and head1.char == head2.char)          -- identical glyph
              or (head1.id == 10 and head2.id == 10) then  -- glue

              if head1.id == 37 then -- glyph
                 char_count = char_count + 1
              elseif head1.id == 10 then -- glue
                 word_count = word_count + 1
              end
              head1 = head1.next
              head2 = head2.next
           elseif (head1.id == 0 or head2.id == 0) then -- end of line
              break
           elseif (head1.id ~= 37 and head1.id ~= 10) then -- some other kind of node
              head1 = head1.next
           elseif (head2.id ~= 37 and head2.id ~= 10) then -- some other kind of node
```

```
            head2 = head2.next
         else -- no match, no special node
            break
         end
      end
      -- analyze last non-matching node, check for punctuation
      if ((head1 and head1.id == 37 and head1.char > 49)
           or (head2 and head2.id == 37 and head2.char > 49)) then
         -- not a word
      elseif char_count > 0 then
         word_count = word_count + 1
      end
      return char_count,word_count,head1,head2
   end

   highlight = function (line,nend,color)
      local n = node.new("whatsit","pdf_literal")

      -- get dimensions
      local w,h,d = node.dimensions(line.head,nend)
      local w_pts = w/65536 -- scaled points to points

      -- set data
      n.data = "q " .. color .. " 0 0 m 0 5 l " .. w_pts .. " 5 l " .. w_pts .. " 0 l b Q"

      -- insert node
      n.next = line.head
      line.head = n
      node.slide(line.head)
   end

   homeoarchy = function (head)
      local cur_line = head
      local prev_line -- initiate prev_head

      local max_char = tonumber(\impnattypohomeoarchymaxchars)
      local max_word = tonumber(\impnattypohomeoarchymaxwords)

      while head do
        if head.id == 0 then -- new line
          prev_line = cur_line
          cur_line = head
          if prev_line.id == 0 then
             char_count,word_count,prev_head,cur_head = compare_lines(prev_line,cur_line)
             if char_count >= max_char or word_count >= max_word then
                local color
                if word_count >= max_word then
                   color = "q \usecolor{\impnattypohomeoarchywordcolor}"
                else
                   color = "q \usecolor{\impnattypohomeoarchycharcolor}"
```

```
                end

                -- highlight both lines
                highlight(prev_line,prev_head,color)
                highlight(cur_line,cur_head,color)
              end
            end
          end
          head = head.next
        end
        return true
      end

      luatexbase.add_to_callback("post_linebreak_filter",homeoarchy,"homeoarchy")
      \end{luacode}
   \else
      \PackageError{The homeoarchy option only works with LuaTeX}
   \fi
 \fi
\fi
```