

The `mynote` \LaTeX class

Luca Peliti

February 26, 2016

Abstract. The `mynote` \LaTeX class is inspired by the `iopart` article class of the IOP, but has been adapted to the writing of notes in either English, French, or Italian. Most of the IOP article class commands can be used within it, but a few have been added, and are described here.

1. Introduction

The `mynote` \LaTeX class aims at giving a convenient interface for writing up easily readable notes. It is inspired by the `iopart` article class, which can be obtained from <http://www.iop.org/Journals/textstyle>, together with its documentation. The present guide only describes the changes made on that class. Note, however, that `mynote` does *not* need the presence of `iopart.cls` to typeset.

2. Installation

The installation procedure depends on the structure of the local \TeX system. The files `mynote.cls`, `mynoteams.sty`, `setstack.sty`, `mynote10.clo` and `mynote12.clo` should be put where `latex` can find them. On a standard Unix-like installation, e.g., they can be put in `texmf/tex/latex/local/MyNote/`, where `texmf` is the root directory of the \TeX installation. Once this is done, calling `texconfig rehash` as root allows to refresh the filename database, making them available to the system.

The file `mynote.ltx` contains a template for using the class. It should be put where the *editor* used on the source files can find it.

3. Options

There are two sets of options:

Size options: `10pt` (default) and `12pt`. They define the normal font size as in `iopart` but keeping the text width and height of the IOP `12pt` option.

Language options: `english` (default), `french` and `italian`. The `french` and `italian` options load the corresponding `babel` packages. The class always loads the `utf8` option of the package `inputenc` for typesetting accented characters.

4. Commands

On top of the commands of `iopart`, a few new commands have been defined:

Date: The `\date{#}` command in the title page sets the date. It can have `\today` as an argument, in which case the date will be written according to the conventions of the chosen language. If the argument is empty, the date will not be printed.

No sections: The `\nosection` command at the end of the title page allows the correct typesetting of a note with no sections.

Average: The `\average{#}` command typesets angular brackets around its argument: `\average{A}` yields $\langle A \rangle$.

Abs: The `\abs{#}` command typesets vertical lines around its argument: `\abs{A}` yields $|A|$.

$k_B T$: The `\kt` macro yields the product of Boltzmann's constant k_B times the temperature T : `\kt`.

Given: The `\given` command yields the vertical separation bar appearing in conditional probabilities: `\P(a\given b)` yields $P(a | b)$.

Roman letters: Some mathematical symbols are best typeset in Roman face:

The differential: The command `\D` yields the Roman 'd' in mathematical environment: `\D x` yields dx .

Euler's constant: The command `\E` yields the Roman 'e': `\E^x` yields e^x .

The imaginary unit: The command `\I` yields the Roman 'i': `\I^2=-1` yields $i^2 = -1$.

The subscript 'c': The command `\C` yields the Roman 'c': `\T_C` yields T_c .

Units: The class loads the `units.sty` and `upgreek.sty` packages for physical units. It defines the macro `\micro` which yields the micro symbol μ , e.g., for micrometers: $1\mu\text{m} = 10^{-6}\text{m}$.

Some geometrical notations: Some useful geometrical notations:

Point: The command `\point{#}` typesets its argument in Roman face in all environments: `\textit{the point \point{A} is\dots}` yields *the point A is...*

Line: The command `\lin{#}` typesets its argument in Roman face in all environments, and sets a line over it: `\textit{the line \lin{AB}is\dots}` yields *the line \overline{AB} is...*

Angle: The command `\ang{#}` typesets its argument in Roman face in all environments, and puts an angle symbol before it: `\textit{the angle \ang{ABC}is\dots}` yields *the angle $\angle ABC$ is...*

Vectors: The vector command `\vec{#}` has been redefined to yield the bold math face: `\vec{w}`, `\vec{\alpha}` yield respectively \boldsymbol{w} , $\boldsymbol{\alpha}$. Note that, in order to obtain this behavior, the *mynote* class loads the `bm` package.

5. Environments

Three new environments, for *unnumbered* theorems, proofs and comments, have been defined. Their syntax is

```
\begin{environment}["option"]
  "text"
\end{environment}
```

Proposition: The environment `proposition` starts a new paragraph, with “option” (if present) or `\propositionname`, i.e., **Theorem** (en), **Teorema** (it), **Théorème** (fr)) in **boldface**, followed by a full stop. The “text” follows in *italics*. If “option” is empty, no text is printed at the beginning of the paragraph.

Thus we have, e.g.,

```
\begin{proposition}
  The sum of the area of the squares\dots
\end{proposition}
```

which yields

Theorem. *The sum of the area of the squares...*

and

```
\begin{proposition}[Lemma]
  The sum of the area of the squares\dots
\end{proposition}
```

which yields

Lemma. *The sum of the area of the squares...*

On the other hand,

```
\begin{proposition}[]
  The sum of the area of the squares\dots
\end{proposition}
```

yields

The sum of the area of the squares...

Proof: starts a new paragraph with “option” (if present) or `\proofname` (*Proof* (en), *Dimostrazione* (it), *Démonstration* (fr)) in *italics*, followed by a full stop. The “text” follows in normal font, and is closed by `\proofendsymbol` (a full square: ■). If the “option” is empty, no text is printed at the beginning of the paragraph. Thus we have, e.g.,

```

\begin{proof}
  Let the triangle \point{ABC}
  \dots\ as we wished to prove.
\end{proof}

```

which yields

Proof. Let the triangle ABC... as we wished to prove. ■

and

```

\begin{proof}[\proofname\ (Hermite)]
  Let the triangle \point{ABC}\dots\ as we wished to prove.
\end{proof}

```

which yields

Proof (Hermite). Let the triangle ABC ... as we wished to prove. ■

Comment: Starts a new paragraph, with “option” (if present) or `\commentname` (COMMENT (en), COMMENTO (it), REMARQUE (fr)) in SMALL CAPS, followed by full stop. The “text” follows in normal face. If the “option” is empty, no text is printed at the beginning of the paragraph. Thus we have, e.g.,

```

\begin{comment}
  If the triangle \point{ABC}\dots
\end{comment}

```

which yields

COMMENT. If the triangle ABC...

6. Footnotes

The footnote markers are given by default by symbols selected in order `\dag` † (1), `\ddag` ‡ (2), `\S` § (3), `\l` || (4), `\P` ¶ (5), `$^+$` + (6), `*` * (7), `\sharp` ‡ (8), `\dag\dag` †† (9) unless optional argument of [num] is used to specify the required symbol, 1=`\dag`, 2=`\ddag`, etc. This can be accomplished by writing

```

\footnote[3]{Text of footnote.}

```

which yields §.

When the option `italian` is present, the footnote symbols are numbers as exponents. When the option `french` is present, the footnote symbols are numbers as exponents, followed by a right parenthesis.

§ Text of footnote.

7. AMS symbols

In order to load \mathcal{AMS} symbols, use the command `\usepackage{mynoteams}` in the preamble. Directly loading the \mathcal{AMS} packages will lead to error. The following commands can be defined by loading (via `\usepackage{#}`) *both* `mynoteams` and the IOP package `setstack.sty`, which I have joined to the MyNote package: `\overset` `\underset` `\sideset` `\substack` `\boxed` `\leftroot` `\uproot` `\dddo` `\ddddot` `\varrow` `\harrow`.