

# The tango document class

Michel BOVANI

July 8, 2024  
v. 0.7.1

---

---

## Contents

<b>1</b>	<b>What the tango document class is</b>	<b>1</b>
<b>2</b>	<b>Options overview</b>	<b>2</b>
<b>3</b>	<b>Packages loaded by the tango document class</b>	<b>4</b>
<b>4</b>	<b>Fonts</b>	<b>5</b>
4.1	The tango font system . . . . .	5
4.2	Replacing the tango font system. . . . .	5
4.3	Font commands and text font commands . . . . .	6
<b>5</b>	<b>Using tango</b>	<b>6</b>
5.1	Generalities . . . . .	6
5.2	Modified standard commands . . . . .	6
5.3	Using two or more output formats . . . . .	6
5.4	All the color themes . . . . .	7
<b>6</b>	<b>Language settings</b>	<b>7</b>
<b>7</b>	<b>Statements</b>	<b>8</b>
7.1	Predefined statements . . . . .	8
7.2	Defining your own statements . . . . .	9
7.3	Redefining existing statements . . . . .	9
<b>8</b>	<b>Exercises</b>	<b>11</b>
8.1	Typesetting exercises . . . . .	11
8.2	The secexo and chapexo environments . . . . .	12
	Example of an exercises subsection. . . . .	12
<b>9</b>	<b>Lists</b>	<b>12</b>
9.1	General principles. . . . .	12
9.2	List based environments . . . . .	13
9.3	Redefining labels of itemize lists . . . . .	14
<b>10</b>	<b>Other environments</b>	<b>14</b>
10.1	The remark and example environments . . . . .	14
10.2	The alert environment . . . . .	15
10.3	The proof environment . . . . .	15
<b>11</b>	<b>Figures, tables, floats and captions</b>	<b>16</b>

---

---

## 1 WHAT THE TANGO DOCUMENT CLASS IS

Tango is a Latex document class (`\documentclass[<options>]{tango}`) for use by mathematics teachers (roughly speaking, from high school to undergraduate level). It makes it possible to typeset several types of documents, from the short handout to the complete textbook. The “maths dedicated aspect” essentially consists in:

- ☞ loading maths packages;
- ☞ offering dedicated commands and environments for statements (definitions, propositions, theorems...), exercises, numbering.

Tango is reasonably configurable, even if certain choices are imposed: mandatory use of lualatex, opentype fonts and recent installation, in particular. The tango class defines eight output formats, five of which are suitable for tablets, e-readers and smartphones; a system currently under development can be used to introduce a particular input for a given output format (for example, forcing a page break for the A5 format only). Finally, the tango class can be used for projects of a more general nature than maths lessons: science courses, but even more general usage with some extra configuration effort (see the Candide directory in the examples).

Another class of documents for presenting school work (training exercises, test or exam subjects) will soon be available, called “bravo”.

## 2 OPTIONS OVERVIEW

Please note that the table below only lists the options actually managed by the tango class. Other options, if used when calling the class, can simply be passed to certain packages. This is the case, for example, with the babel language options, or the math-style option, which is systematically passed to the unicode-math package.

Default values are in bold red. Bold options marked with a **star\*** can be used in the preamble and in the body of the document with the `\tgoresetup{ }` command: for example, in order to change the colors inside the document, you can use `\tgoresetup{ColorTheme=Navy}`. Other options can only be mentioned when the tango class is called.

**Table 1:** The tango class options

Option	Usage
french	french=true/ <b>false</b> . Adaptation to the French language and reserved names for the tango class (theorem gives <i>e.g.</i> théorème). Mandatory use for French-speaking users. This option is also passed to the babel package.
<language options>	Tango will pass other language options to the babel package. Note that <ul style="list-style-type: none"> <li>☞ Only English and French languages are fully implemented.</li> <li>☞ Even English-speaking people must pass a language option (english, british, UKenglish, USenglish or american) for babel to the tango class.</li> </ul> See section 6 below.
no-statement	no-statement=true/ <b>false</b> . The thm, defin, lem, coro, propo environments (for theorems, definitions, lemmas, corollaries, propositions) are defined by default. The no-statement option allows you to define them later, on request (see <code>\newstatement</code> and <code>\renewstatement</code> commands in section 7 page 8).
no-hyperref	no-hyperref=true/ <b>false</b> . If you’re only planning a single paper output....
no-indent/indent	no-indent=true/ <b>false</b> . Indents are present by default. The option noindent (meaning no-indent = true) returns us to the left margin. The indent option also exists, but is not very useful!
no-titleindent/titleindent	Title indents are present by default only in French (because in French typography, the first paragraph after a title must be indented: so I prefer in this case the titles to be indented too). You can use no-titleindent in French and titleindent in English to change the default behavior. These options must be called after the language options.

*Continued next page*

**Table 1:** The tango class options

Option	Usage
<b>ThmNamePos*</b>	ThmNamePos=<value> where <value> can be <b>left</b> or right. The name of the statement (Theorem, Definition, etc.) is displayed to the left or right of the box.
PubliClass	PubliClass=<value> where <value> can be <b>article</b> or book. Indicates the base document class used (Latex standard article document class or Latex standard book document class.) The tango class is designed so that short handouts or monographs composed with the article version can then be grouped together as chapters in the same volume with the book version. See the mathematics directory in the examples documentation. With the article version, the <code>\part</code> and <code>\tableofcontents</code> commands are disabled. The <code>\tgoshorttoc</code> command can be used to obtain a summary at the beginning of a monograph.
oneside/twoside	The option <code>twoside</code> makes recto/verso, <code>oneside</code> simply makes recto. The default is <b>oneside</b> . For now, there's no adjustment of inner and outer margins with <code>twoside</code> but it's easy to adapt it to your needs with the <code>\geometry</code> command in the preamble. Note that one may prefer <code>twoside=false</code> to <code>oneside</code> (why not?).
FontSize	FontSize=<value> where <value> can be 9pt, <b>10pt</b> , 11pt ou 12pt.
<b>ColorTheme*</b>	ColorTheme=<value> where <value> can be Blue, Navy, USAF, Azur, Red, Framboise, Brique, Sienne, Caramel, Olive, Tannen, <b>GrayGray</b> , BlackAndWhite, Ink or Steel. These values are worth trying out: each defines a set of two colors, one solid (ColorOne) for titles and one very light (ColorTwo) for box backgrounds.
<b>ColorOne*/ColorTwo*</b>	These options allow the user to define the colors of a personal theme separately; the argument must be a suitable argument for <code>\definecolor</code> . See <code>xcolor</code> package (loaded without options by the tango class) documentation.  <i>Examples</i> <pre data-bbox="571 1193 1366 1341"> \ColorOne={rgb}{0.75,0.1,0.05} \ColorOne={cmyk}{0,0.8,0.8,0.2} \ColorTwo={{rgb}{0.75,0.96,0.75}} (note that external braces may be used, but are not mandatory).</pre>
<b>ColorOneNamed* / ColorTwoNamed*</b>	These options allow the user to define the colors of a personal theme separately; the argument must be a suitable argument for <code>\colorlet</code> . See <code>xcolor</code> package (loaded without options by the tango class) documentation.  <i>Examples</i> <pre data-bbox="571 1541 995 1615"> \ColorOneNamed={red!70!blue} \ColorTwoNamed={yellow!12}</pre> <p data-bbox="536 1630 1366 1729">If you want to use a predefined set of colors, say <code>dvipsnames</code>, you may reload the <code>xcolor</code> package with the <code>dvipsnames</code> option in your preamble, and then use the <code>\tgosetup</code> command:</p> <pre data-bbox="571 1742 1264 1774"> \tgosetup{ColorOneNamed={RedOrange!80!BrickRed}}</pre>

*Continued next page*

Table 1: The tango class options

Option	Usage
Output	Output=<value> where <value> can be <b>A4paper</b> , Letter, A5paper, BigTablet, Tablet, SmallTablet, eReader or Smartphone. BigTablet, Tablet and SmallTablet correspond to 13-, 11- and 8.5-inch screens respectively. eReader is for a 6-inch e-reader. It should be noted that in the case of eReader and Smartphone, the requested format is larger than the actual format, so that the font sizes are much smaller than required. So the FontSize=9pt option produces a result on their screens that may be difficult to read.
Numbers	<p>Numbers=&lt;value&gt; where &lt;value&gt; can be TextOldStyle, OldStyle, MathOldStyle, FullOldstyle. If you don't ask for anything, you get the "lining" numbers.</p> <ul style="list-style-type: none"> <li>☞ Numbers=TextOldStyle activates oldstyle (or lower-case) digits (like these 0123456789) of the erewhon font (basic text font) for the entire document.</li> <li>☞ Numbers=OldStyle activates oldsyle (or lower-case) digits for all text fonts used in the document.</li> <li>☞ Numbers=MathOldStyle activates lower-case digits only in mathematics (I can't think of any good reason to do this!).</li> <li>☞ Numbers=FullOldStyle activates lower-case digits for all text fonts as well as for mathematics.</li> </ul> <p>It should be noted that it is always possible to switch locally from the digits selected with this option to one of their variants (for example, section titles are systematically numbered using "lining" digits, since these titles are themselves capitalized).</p>
StylisticSet	StylisticSet={<value1>, <value2>,...} where <value1>, <value2>, etc. are to be taken from upint, leqslant, smaller, subsetneq and parallelslant. These options concern the Erewhon-Math font, which is a (very) extended open-type version by Daniel FLIPO of the math fonts of the Fourier-Gutenberg system. For more information, see the Erewhon-Math documentation (texdoc erewhon-math in any good terminal).
CharacterVariant	CharacterVariant={<value1>, <value2>,...} where <value1>, <value2>, etc. can be chosen from zero, hslash, emptyset, epsilon, kappa, pi, phi, rho, sigma, theta, partial, Ecal, Qcal and Tcal. These options concern the Erewhon-Math font, which is a (very) extended opentype version by Daniel FLIPO of the math fonts of the Fourier-Gutenberg system. For more information, see the Erewhon-Math documentation (texdoc erewhon-math in any good terminal).

### 3 PACKAGES LOADED BY THE TANGO DOCUMENT CLASS

I only mention here the packages directly loaded by the tango class. Each of these packages can in turn load others...

- ☞ `\RequirePackage{xcolor}`. The xcolor package is used by the tango class to define the colors of the various themes. It is called without any options, but you can reload it in the preamble of your document with options such as dvipsnames, svgnames or x11names.
- ☞ `\RequirePackage{mathtools}`. The mathtools package loads the amsmath package, to which it adds a complement.
- ☞ `\RequirePackage{geometry}`. Used by the tango class in order to define the page design of the various outputs formats. You can use the `\geometry` command in the document preamble to modify this layout.

- ☞ `\RequirePackage{enumitem}`. Used by the tango class to configure the various list-based environments in its own way (see section 9). These settings can be modified by the user in the preamble.
- ☞ `\RequirePackage{titletoc}`. Used by the tango class to fix the table of contents appearance which can be modified by the user in the preamble.
- ☞ `\RequirePackage[pagestyles, toctitles, newlinetospace, clearempty]{titlesec}`. Used by the tango class to configure the appearance of titles at different levels. This can be modified by the user in the preamble. The default English version adds the `noindentafter` option.
- ☞ `\RequirePackage{ccaption, caption}`. Used by the tango class to configure the captions appearance for figures and tables. (See section 11).
- ☞ `\RequirePackage{fontspec}\RequirePackage{unicode-math}`. The tango class requires `lualatex`, `opentype fonts`, and `Unicode` settings.
- ☞ `\RequirePackage{fourier-orns}`. See the section 4 below.
- ☞ `\RequirePackage{iflang}`. Used by tango to send a class error message if no language option is given.
- ☞ `\RequirePackage{babel}`. The babel package is called without options. It is up to you to pass babel options to the tango class. See section 6 below.
- ☞ `\RequirePackage[colorlinks, linkcolor=ColorOne, urlcolor=ColorOne]{hyperref}`. This hyperref configuration can be modified using the `\hypersetup` command in your preamble. You can also use the `no-hyperref` option.
- ☞ `\RequirePackage{array, graphicx, microtype, numprint, float, afterpage}`. These packages are provided but not explicitly used by tango.

## 4 FONTS

### 4.1 The tango font system

Tango's main font is Erewhon, an opentype version of Utopia (a PostScript type 1 font which was designed in 1989 by Robert SLIMBACH and once donated by Adobe to the X-consortium and featured on TeXlive). Unlike Utopia, Erewhon, designed by Michael SHARPE, covers a large number of languages and offers a wide range of functions, available through opentype features. It should be noted that Adobe distributes a commercial opentype version of Utopia, which is less rich in glyphs than Erewhon, but has an intermediate font-weight and optical character sizes.

The tango mathematical font is Erewhon Math, a scientific complement to Erewhon designed by Daniel FLIPO. Erewhon Math is in a way the opentype counterpart of Fourier-GUTenberg, the scientific complement to Utopia type 1 designed by me in 2002. Erewhon Math offers a large number of functions made accessible by Daniel through the `fourier-otf` package. The tango class does not load `fourier-otf`, but uses all its functions through a system of options.

The sans-serif font of the tango system is Noto Sans, scaled. Noto Sans is a sans-serif font for text in Latin, Cyrillic and Greek characters, with multiple weights and widths.

The main (chapter) title font is Roboto Condensed Bold (as a condensed version of Noto Sans is not available on TeXlive).

The monospaced font is Inconsolata4, scaled.

Finally, tango calls the `fourier-orns` package, a symbols and logos complement to the `fourier` system which has now an opentype version.

### 4.2 Replacing the tango font system

It is, of course, possible to use the `fontspec` `\setmainfont`, `\setsansfont` and `\setmonofont` commands and the `unicode-math` `\setmathfont` command in the preamble in order to modify the parameters of the tango class: this is not advisable, in my humble opinion. But one day there will probably be another class in the tango distribution which will use another font system. Such a document class could be called `charlie`, or `foxtrot` or `zulu`....

### 4.3 Font commands and text font commands

Here, I call “Font commands” commands whose scope can only be limited by a group or an environment. I call “Text font commands”, on the other hand, commands whose scope is limited to their argument, which must not contain a paragraph break. For example, `\bfseries` is a “font command” and `\textbf` is a “text font command”. Here is what is provided by tango (see table 2, 6). All these commands should work with Erewhon and with Noto Sans as well.

Table 2: Tango opentype commands

Command	Nature	Example	Result
<code>\superiors</code>	font command	<code>M{\superiors rs}</code>	$M^{\text{rs}}$
<code>\scinferiors</code>	font command	<code>{\scinferiors C6H12O3}</code>	$C_6H_{12}O_3$
<code>\smartfrac</code>	font command	<code>2{\smartfrac{1}{3}}</code>	$2\frac{1}{3}$
<code>\fullsc</code>	text font command	<code>\fullsc{Chapter}</code>	CHAPTER
<code>\textsup</code>	text font command	<code>M\textsup{rs}</code>	$M^{\text{rs}}$
<code>\textscinf</code>	text font command	<code>\textscinf{C6H12O3}</code>	$C_6H_{12}O_3$
<code>\smartfrac</code>	text font command	<code>2\smartfrac{3}{4}</code>	$2\frac{3}{4}$
<code>\olddigits</code>	font command	<code>{\olddigits01234}</code>	01234
<code>\liningdigits</code>	font command	<code>{\olddigits01\liningdigits01}</code>	0101
<code>\propdigits</code>	font command	<code>{\propdigits011\olddigits011}</code>	011011
<code>\tabulardigits</code>	font command	<code>{\tabulardigits011\olddigits011}</code>	011011

## 5 USING TANGO

### 5.1 Generalities

The tango interface is not so different from that of the standard article and book document classes. Configuration is essentially a matter of choosing options.

For advanced use, it is useful to refer to the documentation for the various packages: first, of course, `unicode-math`, `fontspec`, `mathtools` and `amsmath`. But if you want to change the style of titles, for example, you could take a look at the `titlesec` documentation, and to change the style of lists (`enumerate`, `itemize`, etc.), take a look at `enumitem`. The `geometry` package is essential for anyone wanting to obtain a particular layout. And so on.

### 5.2 Modified standard commands

Since documents designed with the `article` option can become chapters in a book composed later with the `book` option, the `\part` command inherited from the `article` class has been deactivated (only for `article-typeset` documents)

Similarly, the `article` version benefits from a `\tgotitle` command that lets you define the main title of the document. The `\tgotitle*` version exists but does nothing more or less than the unstarred command. However, in the `book` version, `\tgotitle` is an alias for `\chapter` (and `\tgotitle*` is an alias for `\chapter*`), so the unnumbered title of a handout very easily becomes the numbered title of a chapter in a book. This, of course, assumes that the `\includeonly/\include` mechanism is used... A model (currently written in French) is provided in the examples directory.

The command `\tableofcontents` can only be used with the `book` option; it is deactivated with the `article` option: in this case, you have to use `\tgoshorttoc` instead (of course `\tgoshorttoc` is in turn deactivated in the `book` version). Note that, if you need multiple or partial tocs, you have to use a dedicated package.

### 5.3 Using two or more output formats

What I’m describing here is still rudimentary and experimental. It will be developed further if it proves successful.

For each output format, two particular commands are provided. For example, in case of the Smartphone output option, you have `\SmartphoneCommand` and `\XSmartphoneCommand`. The `\SmartphoneCommand` argument is only taken into account if `Output=Smartphone` have been used. Conversely, the `\XSmartphoneCommand` argument is only taken into account if this option has not been used.

#### Examples

`\SmartphoneCommand{\pagebreak}` asks for a page break only for the smartphone output.

`\XSmartphoneCommand{\[< long math formula >\]}`  
`\SmartphoneCommand{\begin{multine*}<split formula>\end{multine*}}`  
 asks for a multiline version of a formula only for the smartphone output

Provided commands are:

`\AfourCommand` and `\XAfourCommand` for the A4 format;

`\LetterCommand` and `\XLetterCommand` for the letter format;

`\AfiveCommand` and `\XAfiveCommand` for the A5 format;

`\BigTabletCommand` and `\XBigTabletCommand` for the big tablet format;

`\TabletCommand` and `\XTabletCommand` for the tablet format;

`\SmallTabletCommand` and `\XSmallTabletCommand` for the small tablet format;

`\eReaderCommand` and `\XeReaderCommand` for the e-reader format;

`\SmartphoneCommand` and `\XSmartphoneCommand` for the smartphone format.

## 5.4 All the color themes



## 6 LANGUAGE SETTINGS

A language option *must* be passed to the babel package via the tango class: this is true *even* for English-speaking people. A class error message will be generated otherwise. If you ignore this message you will get no hyphenation patterns at all!

For now, only the English and French languages are fully supported. With other languages, you benefit from the babel facilities, but the “tango words” will remain defined by:

```
\newcommand\TgoTheoremName{Theorem}
\newcommand\TgoDefinitionName{Definition}
\newcommand\TgoPropositionName{Proposition}
\newcommand\TgoLemmaName{Lemma}
\newcommand\TgoCorollaryName{Corollary}
\newcommand\TgoRemarkName{Remark}
```

```

\newcommand\TgoExampleName{Example}
\newcommand\TgoContentsName{Contents}
\newcommand\TgoExerciseName{Exercise}
\newcommand\TgoExercisesSubsectionName{Exercises}
\newcommand\TgoExercisesSectionName{Exercises}

```

If you want to use tango with a new language, here is what you can do:

1. Translate these names in your language (for now it is done only for the french language), using `\renewcommand`.
2. (Better) Mail me all the necessary information (at least babel language option used and your translation of the tango names) and I will adapt the tango class to your language.

## 7 STATEMENTS

### 7.1 Predefined statements

The environments offered by the tango document class for the statements were defined using the `\newstatement` command, whose syntax is:

```

\newstatement{<environment>}{<counter>}{<statement-name>}[<optional-font-
command>]

```

For example, the `thm`, `defin`, `propo`, `coro` and `lem` environments have been defined using something like:

```

\newcounter{thm} \newstatement{thm}{thm}{Theorem}[\itshape]
\newcounter{defin} \newstatement{defin}{defin}{Definition}
\newcounter{propo} \newstatement{propo}{propo}{Proposition}[\itshape]
\newcounter{coro} \newstatement{coro}{coro}{Corollary}
\newcounter{lem} \newstatement{lem}{lem}{Lemma}

```

If you type:

```

\begin{propo}
The sum of two odd integers is an even integer.
\end{propo}
\begin{thm}[pons asinorum]
The angles at the base of an isosceles triangle are equal.
\end{thm}

```

you will obtain:

#### **PROPOSITION 1**

*The sum of two odd integers is an even integer.*

#### **THEOREM 1 (pons asinorum)**

*The angles at the base of an isosceles triangle are equal.*

As you can see, each statement defined with `\newstatement` has an optional bracketed argument (which is displayed within parentheses after the statement name and number). Please, note that these environments should not be used inside a list-based environment (like `center`, `quote`, `enumerate`, `itemize`...).

For each statement defined with `\newstatement` there is also



☞ a starred form with no statement number:

```
\begin{coro*}
The angles at the base of an isosceles triangle are equal.
\end{coro*}
```

#### COROLLARY

The angles at the base of an isosceles triangle are equal.

☞ and an optional first argument delimited with < > to replace locally the full theorem name.

```
\begin{thm*}<D'Alembert-Gauss theorem>[fundamental theorem of algebra]
Every non-constant single-variable polynomial with
complex coefficients has at least one complex root.
\end{thm*}
```

#### D'ALEMBERT-GAUSS THEOREM (fundamental theorem of algebra)

*Every non-constant single-variable polynomial with complex coefficients has at least one complex root.*

## 7.2 Defining your own statements

If you need a new statement type, say an axm environment for axioms, it's easy:

```
\newcounter{axio} \newstatement{axm}{axio}{Axiom}
```

Note that it is up to you to define the counter for this new environment: if this counter does not still exist, it has to be created with `\newcounter`. Of course, the counter name may be different from the environment name. If you have used the no-statement option, you will have to define all the statements you need, with the environment names you prefer.

## 7.3 Redefining existing statements

The `\newstatement` command has a `\renewstatement` companion to redefine existing statement environment. Precisely, you need to use `\renewstatement{foo}...` instead of `\newstatement{foo}...` if and only if the `foo` environment has already been defined.

#### Examples

1. You can redefine the `propo` environment in such a way that it uses the same counter as the `thm` environment and is displayed in the upright font:

```
\renewstatement{propo}{thm}{Proposition}
\setcounter{thm}{0}
\begin{propo}
The sum of two odd integers is an even integer.
\end{propo}
\begin{thm}
The angles at the base of an isosceles triangle are equal.
\end{thm}
```

#### PROPOSITION 1

The sum of two odd integers is an even integer.

#### THEOREM 2

*The angles at the base of an isosceles triangle are equal.*

2. If you have already written *all* your algebra book in Latex, with propositions, theorems and corollaries and you realize that, *now*, you want there is no more difference between propositions, theorems and corollaries: the three types of statements should be typeset in italic, named “Proposition” and numbered with the same counter, you can just use in your preamble:

```
\renewstatement{coro}{thm}{Proposition}[\itshape]
\renewstatement{propo}{thm}{Proposition}[\itshape]
\renewstatement{thm}{thm}{Proposition}[\itshape]
\setcounter{thm}{0}
\begin{propo}
The sum of two odd integers is an even integer.
\end{propo}
\begin{thm}
The sum of two even integers is an even integer.
\end{thm}
\begin{coro}
If two integers have the same parity, their sum is even.
\end{coro}
```

and you will get

**PROPOSITION 1**

*The sum of two odd integers is an even integer.*

**PROPOSITION 2**

*The sum of two even integers is an even integer.*

**PROPOSITION 3**

*If two integers have the same parity, their sum is even.*

instead of

**PROPOSITION 1**

*The sum of two odd integers is an even integer.*

**THEOREM 1**

*The sum of two even integers is an even integer.*

**COROLLARY 1**

*If two integers have the same parity, their sum is even.*

3. And finally `\renewstatement` and `\newstatement` have a starred form with which statement numbers are displayed *before* the statement name. More, if the statement counter is the subsection or subsubsection counter, the concerned statement titles are displayed in the same form and placement as subsection or subsubsection titles; of course the placement depends on the value of `notitleindent`.

```

\renewstatement*{thm}{thm}{Theorem}
\begin{thm}<D'Alembert-Gauss theorem>[fundamental theorem of algebra]
Every non-constant single-variable polynomial with
complex coefficients has at least one complex root.
\end{thm}
\renewstatement*{thm}{subsection}{Theorem}[\itshape]
\begin{thm}<D'Alembert-Gauss theorem>[fundamental theorem of algebra]
Every non-constant single-variable polynomial with
complex coefficients has at least one complex root.
\end{thm}
\renewstatement*{thm}{subsubsection}{Theorem}
\begin{thm}<D'Alembert-Gauss theorem>[fundamental theorem of algebra]
Every non-constant single-variable polynomial with
complex coefficients has at least one complex root.
\end{thm}

```

## 2 D'ALEMBERT-GAUSS THEOREM (fundamental theorem of algebra)

Every non-constant single-variable polynomial with complex coefficients has at least one complex root.

### 7.4 D'ALEMBERT-GAUSS THEOREM (fundamental theorem of algebra)

*Every non-constant single-variable polynomial with complex coefficients has at least one complex root.*

#### 7.4.1 D'ALEMBERT-GAUSS THEOREM (fundamental theorem of algebra)

Every non-constant single-variable polynomial with complex coefficients has at least one complex root.

## 8 EXERCISES

### 8.1 Typesetting exercises

Tango provides the `\exo` command which lets you typeset a numbered exercise (the counter is `tgoexo`). There is a starred version, which is not numbered. As for the statements, the bracketed option provides a complementary title between parenthesis and the “< >” marked first option provides a substitution title.

```

\exo
Prove that if  $G$  is a finite group and  $H$  a subgroup of  $G$ ,
then the order of  $H$  divides the order of  $G$ .
\exo[Lagrange theorem]
Prove that if  $G$  is a finite group and  $H$  a subgroup of  $G$ ,
then the order of  $H$  divides the order of  $G$ .

```

#### Exercise 1

Prove that if  $G$  is a finite group and  $H$  a subgroup of  $G$ , then the order of  $H$  divides the order of  $G$ .

#### Exercise 2 (Lagrange theorem)

Prove that if  $G$  is a finite group and  $H$  a subgroup of  $G$ , then the order of  $H$  divides the order of  $G$ .

```

\newcommand\tgostar{\raisebox{-0.5ex}{\large\textborn}}
\exo*{Chinese remainder problem}[\tgostar\tgostar\tgostar]
Let  $n_1, \dots, n_k$  be integers greater than 1 and let us denote by  $N$ 
the product of the  $n_i$ . Prove that if the  $n_i$  are pairwise coprime, and if
 $a_1, \dots, a_k$  are any integers, then the system
\begin{align*}
x &\equiv a_1 \pmod{n_1} \\
&\vdots \\
x &\equiv a_k \pmod{n_k}
\end{align*}
has a solution, and any two solutions are congruent modulo  $N$ .

```

### Chinese remainder problem (\*\*\*)

Let  $n_1, \dots, n_k$  be integers greater than 1 and let us denote by  $N$  the product of the  $n_i$ . Prove that if the  $n_i$  are pairwise coprime, and if  $a_1, \dots, a_k$  are any integers, then the system

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

has a solution, and any two solutions are congruent modulo  $N$ .

## 8.2 The secexo and chapexo environments

These environments are designed to present a part devoted to exercises at the end of a section (in the case of secexo) or a chapter (in the case of chapexo). Within these environments, you can use the command \exo

The title, in capitals and centred, is “EXERCISES” by default. The optional argument in square brackets allows you to choose another. This title is added to the table of contents.

Finally, there is a starred form (\begin{secexo\*}... \end{secexo\*}) which allows you to compose the exercises within these sections in a smaller font size (except if the main font size is nine points).

```

\begin{secexo*}[Example of an exercises subsection]
\exo Prove that the sum of two even integers is an even integer.
\exo Prove that the sum of two odd integers is an even integer.
\end{secexo*}

```

### EXAMPLE OF AN EXERCISES SUBSECTION

#### Exercise 3

Prove that the sum of two even integers is an even integer.

#### Exercise 4

Prove that the sum of two odd integers is an even integer.

## 9 LISTS

### 9.1 General principles

All the horizontal motion in tango is performed in steps equal to an integer multiple of the \TgoStandardMargin length parameter. When the tango class is loaded, the value of this parameter is set to 1.5em, as are paragraph indentation and title indentation (unless these two values are fixed to zero). Successive labels in the enumerate and itemize environments are therefore successively positioned at 1.5em, 3em, 4.5em, etc. from the main margin. It could be unadvised to change these settings (or this should be done using the enumitem package).

## 9.2 List based environments

There are two main lists configurations. The first, active by default, is set with `\tgoStandardLists`; the second is set with `\tgoSmartLists`. With `\tgoStandardLists` the text corresponding to a given level is shifted to the right (by a value equal to `\TgoStandardMargin...`) with respect to the corresponding label. With `\tgoSmartLists`, conversely, the text corresponding to a given level is shifted to the left (by a value equal to `\TgoStandardMargin...`) in relation to the corresponding label.

Here is the default behaviour of the `enumerate` environment (`enumerate` is not different).

- Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna.
- Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.
  - Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla.
    - ✦ Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.
      - Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla.
    - ✦ Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam.
- Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.

And now with `\tgoSmartLists`

- Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna.
- Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.
  - Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla.
    - ✦ Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.
      - Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla.
    - ✦ Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam.
- Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.

Note that `\tgoSmartLists` seems most interesting when you use mainly the first level of these environments. Since the `\tgoSmartLists` doesn't support very well a construction like

```
\begin{enumerate}
\item\begin{enumerate}\item
...
```

that is frequent in exercises questions, the `secexo` and `chapexo` environments set locally the lists to `\tgestandardlists`. It is not the case with the `\exo` command which has no local context.

### 9.3 Redefining labels of itemize lists

Tango provides a `\renewtgoLabels` command which permits to modify the labels of the `itemize` environment. Arguments are numbered from 1 to 4 (there is four levels of `itemize` lists in `Latex`) and must be used like in the examples below.

#### Examples

- `\renewtgoLabels1{\lefthand}` will replace the first label by the `\lefthand` symbol.
- ☞ `\renewtgoLabels4{\textbullet}1{\openbullet}3{\textemdash}` changes the label for the first (again), the third and the fourth levels. Level 2 remains unchanged. As you can see, the order of the arguments is unimportant.

There is also a starred form that restore the initial label (as at the load of the class) which are `\textbullet`, `•`, for the level 1, `\textopenbullet`, `◦`, for the level 2, `\starredbullet`, `✦`, for the level 3 and `\textperiodcentered`, `·` for the level 4 (note that this is the `tango` default, but *not* the `Latex` default).

#### Examples

- `\renewtgoLabels*` restore all the initial values.
- `\renewtgoLabels*2{\textendash}` restore initial values except for the second level which is changed to `–`.

## 10 OTHER ENVIRONMENTS

In addition to the statements environments, the `tango` class offers the following environments:

### 10.1 The remark and example environments

```
\begin{remark}
This is a very good remark.
\end{remark}
\begin{example}
This is a very short example.
\end{example}
```

*Remark.* — This is a very good remark.

#### Example

This is a very short example.

These two environments have a bracketed optional argument which displays a replacement title. For example:


```
\begin{example}[Remarks]
\begin{enumerate}
\item A first (very good) remark.
\item A second remark (which is best, if possible).
\end{enumerate}
\end{example}
```

#### Remarks

1. A first (very good) remark.
2. A second remark (which is best, if possible).


## 10.2 The alert environment

```
\begin{alert}
You should not even try to use \verb+\expandafter\expandafter\expandafter+
unless you are actually a very experimented \TeX{} user, because you could get
a terrible headache.
\end{alert}
```

 You should not even try to use `\expandafter\expandafter\expandafter` unless you are actually a very experimented  $\TeX$  user, because you could get a terrible headache.

The alert environment also support an optional bracketed argument.

```
\begin{alert}[\bomb]
You should not even try to use \verb+\expandafter\expandafter\expandafter+
unless you are actually a very experimented \TeX{} user, because you could get
a terrible headache.
\end{alert}
```

 You should not even try to use `\expandafter\expandafter\expandafter` unless you are actually a very experimented  $\TeX$  user, because you could get a terrible headache.

## 10.3 The proof environment

The proof environment is adapted from the `amsthm` package. It also support a bracketed optional argument for a replacement title. The `\qedhere` command is also usable.

```
\begin{thm}
The sum of two odd integers is an even integer.
\end{thm}
\begin{proof}
Let  $p$  and  $q$  be two odd integers. We have to prove that there
exists an integer  $n$  such that  $p+q=2n$ .
We know that there is an integer  $p_1$  and an integer  $q_1$  such that
 $[p=2p_1+1\text{ and } q=2q_1+1, \]$ 
hence
\begin{align*}p+q&=(2p_1+1)+(2q_1+1)\\
&=2\underbrace{(p_1+q_1+1)}_n\qedhere\end{align*}
\end{proof}
```

### THEOREM 1

*The sum of two odd integers is an even integer.*

#### *Proof*

Let  $p$  and  $q$  be two odd integers. We have to prove that there exists an integer  $n$  such that  $p + q = 2n$ . We know that there is an integer  $p_1$  and an integer  $q_1$  such that

$$p = 2p_1 + 1 \text{ and } q = 2q_1 + 1,$$

hence

$$\begin{aligned}p + q &= (2p_1 + 1) + (2q_1 + 1) \\ &= 2\underbrace{(p_1 + q_1 + 1)}_n\end{aligned}$$



## 11 FIGURES, TABLES, FLOATS AND CAPTIONS

Thanks to the caption package, the following have been defined:

- `\tgofigcaption` (for figures);
- and `\tgotabcaption` (for tables).

These two commands provide captions similar in appearance to what could be obtained with `\caption`, but also work in the case of non-floating objects.

In this way it is possible to include small tables and figures in a fixed position, with nothing in their appearance to distinguish them from floating objects (whose use is still recommended in most cases).