

NoTeX

ElBi21 - Leonardo BIASON

Somewhere in 2024

Chapter 1

Introduction

This is an attempt to transform the old project of NoTeX into an actual class, done in a proper way. The class is based out of the report class, and all the parameters are defined in the `notex.cls` class file. The class comes with a packet, called `notexmacros`. Such packet adds plenty of macros that can be used by the user, and that were pretty much useful to me while taking notes. This class aims also to reduce drastically the number of errors and thus increase the compile time. If it will be worth, it will be published also on the CTAN archive.

Chapter 2

The notex class

The class file is organized in various structures: **declaration**, **packages importation**, **definition of colors**, **custom commands**, **loading of the class**.

2.1 Declaration

The declaration of the class simply consists of the following lines of command:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{notex}[2024 ElBi21 LaTeX class for taking notes]
```

2.2 Importing the packages needed

The following packages are imported into the `notex.cls` class file:

























```
\RequirePackage{xcolor}           % Package for colors
\RequirePackage{tcolorbox}        % Package for boxes
\RequirePackage[a4paper]{geometry} % Package for the geometry of the file
\RequirePackage{listings}         % Package for code

\RequirePackage[T1]{fontenc}      % Package for font encoding
\RequirePackage{fouriernc}        % Package with the font
\RequirePackage{sourcecodepro}

\RequirePackage{notexmacros}      % Package with macros
```

2.3 Custom colors

There are some colors that are added from the class. This list reports all of them, alongside a quick preview:

Color Preview	Name of the color	Color Preview	Name of the color
	maindoccol		theoremBG
	theoryLINE		theoremTitleBG
	practiceLINE		exerciseBG
	curiosityLINE		exerciseTitleBG
	remarkLINE		exampleBG
	lemmaBG		exampleTitleBG
	lemmaTitleBG		solutionBG
	definitionBG		solutionTitleBG
	definitionTitleBG		remarkBG
	corollaryBG		remarkTitleBG
	corollaryTitleBG		curiosityBG
	proofBG		curiosityTitleBG

2.4 Commands of the class

This class provides various commands, which allow to customize to your liking the template. Here is a list of all the supported (and provided) commands:

```
\maincol{format}{code}
```

Examples: `\maincol{HTML}{303342}`
`\maincol{RGB}{30, 25, 76}`

Sets the main color for the document.

`format` specifies the format of the color. Supports all the formats of the `xcolor` package;

`code` is the code of the color. Must match the format given by the previous parameter

2.5 Loading of the class

2.6 Custom boxes

The NoTeX class adds also some custom boxes, which are helpful when there is the need to mark some important parts of the text, such as definitions, theorems, proofs, code, etc... Here is a showcase of all the boxes and how they can interact with each other.

Uniform Resource Identifier (URI)

A **Uniform Resource Identifier (URI)** for short) is a unique sequence of characters which identifies a logical or physical resource. An example of URI might be:

`https://thisisanexample.com/usernames`

Spectral Theorem

Given a matrix A which is **squared** ($A \in \mathbb{R}^{d \times d}$) and **symmetric** ($A = A^\top$), then **all the eigenvalues** λ_i take **real values** and **all the eigenvectors** are **orthogonal**

This can be easily proved as follows:

The ∇ is equal to 1

The matrices are equal

Running time of the algorithm

If the algorithm receives as an input an ordered list of items, then its running time will be at most $O(n)$. If the list isn't ordered, then it will take at most $O(n \log(n))$

In order to prove this algorithm, we must first consider the running time of the sub-algorithm. Indeed, we'll obtain that:

$$\text{runtime} = O(n \log(n)) + O(n)$$

Sum of internal angles of triangles

Given any triangle, the sum of its internal angles will always be equal to 180° . Moreover, if the triangle is equilateral, then all its angles will be equal to 60°

The **definition**, **theorem**, **lemma** and **corollary** boxes can be called with:

```
\begin{<box_type>}{<term>}
  Your text...
\end{<box_type>}
```


where `<box_type>` can be one between `definition`, `theorem`, `lemma` and `corollary`, while instead `<term>` denotes the title of the box

It is instead possible to invoke the **proof** box by using the following command:

```
\begin{proof}
  Your proof...
\end{proof}
```

The proof box can be used anywhere, it is however thought for the **theorem** and **lemma** boxes (since definitions and corollary should not need proofs).

EXERCISE

 2.6.0

Solve the following equation:

$$x^2 + \sin(3x - 4) - 7 = 6$$

EXAMPLE

 2.6.0

An example of limit is the following:

$$\lim_{x \rightarrow 0} \log_{10}(x + 1) = +\infty$$

```
TestCode.py
1 def main():
2     for i in range(40):
3         print(i)
4     return None
```

```
Terminal
$ mpicc 01-mpi-collectives.c -o 01-mpi-collectives.o
```

Hello there
AAA

Chapter 3

The notexmacros package

In this chapter the `notexmacro` package will be explained. Such package contains a collection of macros which can turn to be useful while using the class. The package is automatically required by the class, so you don't have to install anything else.

The macros are divided into two groups: the math macros and the text macros. The math macros are used within a math environment, while the text macros are not. There are some general purpose macros which can be used both inside and outside a math environment.

3.1 The math macros

`\eq`

Adds space around a `=`. An example follows:

`a \eq b` \rightarrow $a = b$

`\thus`

Adds substantial space around a `\Rightarrow`, and it can be used to define a logical implication (we have *A*, thus we can get *B*). An example follows:

`a \thus b` \rightarrow $a \Rightarrow b$

`\nextline` and `\prevline`

Adds a `\Rightarrow` which can be used at the end (with `\nextline`) or at the beginning (with `\prevline`) of an equation. This can be used while passing from one line to the other of an equation which would usually need more than one line. An example follows:

`ax + b \nextline` \rightarrow $ax + b \Rightarrow$
`\prevline ax + c` \rightarrow $\Rightarrow ax + c$

3.1.1 Specific macros for statistical distributions

`\cov`, `\bino`, `\berno`, `\unif`, `\geom`, `\poiss` and `\multin`

Adds the function of the covariance and the following distributions: the binomial distribution, the Bernoulli distribution, the uniform distribution, the geometric distribution, the Poisson distribution and the multinomial distribution. An example follows:

<code>\cov(X)</code>	–becomes→	$\text{Cov}(X)$
<code>X \sim \bino \sim \berno \sim \unif</code> <code>\sim \geom \sim \poiss \sim \multin</code>	–becomes→	$X \sim \text{Bin} \sim \text{Bern} \sim \text{Unif}$ $\sim \text{Geom} \sim \text{Poisson} \sim \text{Multi}$

3.2 The text macros

`\angbrack{}`

Wraps the content between two angle brackets. Works both in a `math` and in a non-`math` environment. An example follows:

`\angbrack{10}` –becomes→ $\langle 10 \rangle$

`\st`, `\nd`, `\rd` and `\nth`

Adds respectively the `st`, the `nd`, the `rd` and the `th` after a number. Works both in a `math` and in a non-`math` environment. An example follows:

`1\st`, `2\nd`, `3\rd`, `4\nth` –becomes→ 1^{st} , 2^{nd} , 3^{rd} , 4^{th}