The `neoschool` Class

Version : 1.1.0 — Author : Razik Ikhlef

razik.ikhlef@csilyon.fr

Abstract

The `neoschool` class provides secondary school teachers with a set of tools to create educational materials (assessments, lecture notes, exercise sheets with solutions, and more). To meet various needs, it offers eleven predefined color themes, diverse class options for layout and typography, specialized environments, dedicated commands, and preformatted header styles for each document type. It integrates dozens of commonly used LaTeX packages (see the list below), which significantly lightens the preamble and minimizes incompatibilities as much as possible. Multilingual, it supports English, French, and German.

Contents

1 Packages Loaded

The following packages are automatically loaded by the `neoschool` class.

`adforn`
`adjustbox`
`algpseudocode`
`amssymb`
`babel`
`bookmark`
`calc`
`changespace`
`cuted`
`environ`
`fancyvrb`
`fontawesome5`
`forest`
`iftex`
`ifthen`
`kvoptions`
`lastpage`
`listings`
`marginnote`
`mathtools`
`microtype`
`minted`
`mismath`
`multicol`
`needspace`
`pdftexcmds`
`pgffor`
`pgfplots`
`qrcode`
`scrlayer-scrpage`
`silence`
`siunitx`
`tabularray`
`tasks`
`tcolorbox`
`textcase`
`tikz`
`tikzpagenodes`
`tikzsymbols`
`ulem`
`xcolor`
`xhfill`
`xkeyval`
`xsim`
`xstring`

Compiling with `pdflatex` loads the packages `fontenc` (with option `T1`), `inputenc` (with option `utf8`), `newpxtext`, and `newpxmath`.

Compiling with `lualatex` makes available the packages `fontspec`, `luacas`, `lua-ul`, `luacolor`, and applies the fonts `TeX Gyre PagellaX` and `TeX Gyre Heros` in addition to `newpxmath`.

The **graphics** option loads the packages `graphicx` and `wrapfig`. The **faketext** option (formerly `draft`) loads the packages `blindtext` and `lipsum`. The **mathastext** option loads the `mathastext` package.

The **math** option loads the packages `annotate-equations`, `bm`, `cancel`, `mathrsfs`, `nncomma`, `numprint`, `tdsfrmath` (with options `suite` and `taupe`), `tkz-euclide`, `witharrows`, and `xlop`.

The **notes=*length*** option enables framed notes (`todonotes`) in the left and right margins, alternating, and sets their width. It loads the `todonotes` package.

The **apmep** option, which allows direct compilation of exam archives from the association of the same name, defines a set of mathematical commands and loads the packages `esvect`, `fourier-orns`, `numprint` (with option `np`), `pstricks` (and many packages from the `pstricks` ecosystem), `tabularx`, and `textcomp`.

The **mathics** option enables symbolic computation via `mathics` (an open-source alternative to `Mathematica`) and loads the packages `asymptote` and `latexalpha2`.

2 Configuration Options

2.1 Language

- **english, french, german**: activates translations and conventions specific to each language. These options affect theorem and environment labels, typography, and mathematical conventions.
- **nofrenchlist**: disables the French list style (replaces dashes with periods).
- **frenchlistaspar**: treats lists as paragraphs in French (adds final punctuation and initial capitalization).
- **frenchmath**: applies French mathematical conventions (upright uppercase and Greek letters in math mode).

2.2 Overall Appearance

2.2.1 Predefined Themes

The **theme** = `theme-name` option sets the document's color theme. Available themes include: `classic` (default theme), `abyss`, `aether`, `atlantic`, `autumn`, `blossom`, `botanical`, `burgundy`, `cyprus`, `day`, `deepocean`, `duo`, `eagle`, `foresthues`, `frost`, `glacier`, `goldensummer`, `graphite`, `heather`, `heritage`, `kassio`, `magma`, `purplebliss`, `retrocafe`, `saffronsky`, `scribe`, `sepia`, `summer`, `winkle`, and `midnight` (dark).

```
1 \documentclass[theme=retrocafe]{neoschool}
```

2.2.2 Color Modes

These options modify how the theme's colors are applied.

- **unicolor**: uses a single color (based on the title color) for all environments, with variations through transparency.
- **emphasis**: focuses on theory (theorems, definitions colored; rest in gray).
- **practical**: focuses on practice (methods, applications, code colored; theory in gray).
- **academic**: two color families (theory and practice distinct, rest in gray).
- **print**: converts all colors to black and white for printing.

2.2.3 Customizing Colors

Allows overriding the chosen theme's colors.

- **globalcolor** = `color`: main text color.
- **titlecolor** = `color` or **titlehexcolor** = `hex code`: main title color.
- **headcolor** = `color`: section titles color (`\section`).
- **subcolor** = `color`: subsection titles color (`\subsection`).
- **subsubcolor** = `color`: subsubsection titles color (`\subsubsection`).
- **headfootcolor** = `color`: header and footer text color.

2.2.4 Display Options

- **noframe**: removes frames around environments (theorems, exercises...).
- **noback**: removes colored backgrounds from environments.
- **nocodeframe**: removes frames around code blocks.
- **scale**: harmonizes font sizes when compiling with `lualatex`.
- **inlinecodebox**: displays inline code (`\texttt`) in a framed and colored box.

2.3 Customizing the Abstract

The `abstracttitle` option changes the title of the `abstract` environment.

```
1 \documentclass[abstracttitle=Summary]{neoschool}
```

2.4 Typography

- **sfbody**: uses sans-serif font for the body text.
- **sfall**: uses sans-serif font for the entire document (titles, text, etc.).
- **mathastext**: uses the current text font for mathematics.
- **mainface** = `FontName`: sets the main serif font.
- **mainfaceoptions** = `options`: options for the main font (e.g., `Scale=MatchLowercase`).
- **sansface** = `FontName`: sets the sans-serif font.
- **sansfaceoptions** = `options`: options for the sans-serif font.
- **monoface** = `FontName`: sets the monospace font (for code).
- **monofaceoptions** = `options`: options for the monospace font.
- **mathface** = `MathFontName`: sets the math font.
- **mathfaceoptions** = `options`: options for the math font (`pdflatex` only).
- **facefamily** = `FamilyName`: sets a complete font family (e.g., `fira`).
- **facefamilyoptions** = `options`: options for the font family.

```
% Example with Fira Sans
\documentclass[
    facefamily=Fira Sans,
    facefamilyoptions={%
        sfdefault,
        lining
      },
    monoface=Fira Mono,
    monofaceoptions={Scale=0.85}
]{neoschool}
```

2.5 Layout

2.5.1 Margins and Spacing

- **margin** = `length`: sets the width of horizontal margins (default: `2cm`). Vertical margins are adjusted automatically.
- **notes** = `length`: enables margin notes (`todonotes`) and sets their width.
- **noindent**: removes paragraph indentation.
- **indent** = `length`: sets the indentation size (default: `1em`).
- **compact**: enables a compact mode:
  - reduces vertical spacing (paragraphs, lists, titles, environments);
  - decreases internal margins of environments;

- compresses line spacing.

2.5.2 Output Options (Multiple Layouts)

These options allow printing multiple logical pages on a single physical page.

- **2a5toa4**: prints 2 identical A5 pages on an A4 landscape sheet.
- **2a4toa3**: prints 2 identical A4 pages on an A3 landscape sheet.
- **4a5toa3**: prints 4 identical A5 pages on an A3 sheet.
- **2toa3**: prints 2 different A4 pages on an A3 landscape sheet.
- **bookleta5**: creates an A5 booklet (A5 pages on folded A4 sheets).
- **bookleta4**: creates an A4 booklet (A4 pages on folded A3 sheets).

2.5.3 Table of Contents

- **compacttoc**: reduces spacing in the table of contents.
- **monotoc**: the table of contents uses the text color (`globalcolor`).
- **twocolumntoc**: displays the table of contents in two columns.

2.5.4 Headers and Footers

- **fullheader**: enables a complete header and footer (document type, title, level / date, institution, pagination). Use with `\neoheader`.
- **headrule**: adds a rule below the header.
- **footrule**: adds a rule above the footer.
- **headfootrule**: adds both rules.

3 Document Styles

3.1 Title Styles (`\maketitle`)

These options significantly alter the appearance of the first page and title.

3.1.1 Exam Styles

Ideal for supervised tests, mock exams.

- **exam**: full style with a table for name, class, date, etc., and a grading banner. Requires `\neoheader`.
- **shortexam**: more compact exam style with essential information. Requires `\neoheader`.
- **mockexam**: specific style for mock Bac/Brevet exams, with a standardized cover page (France). Requires `\neoheader`.

```
\documentclass[exam]{neoschool}
\neoheader{
  type = Supervised Test No.1,
  school = Imaginary High School,
  level = Senior Math Specialization,
  duration = 2h,
  calculator = exam % or true/false
}
\title{Numerical Sequences}
\date{October 21, 2025}
\subject{Mathematics} % Optional
\begin{document}
\maketitle
...
```

3.1.2 Assessment Styles

Suitable for quizzes, short tests.

- **eval**: standard style, information distributed in corners. Requires `\neoheader`.
- **evalicons**: `eval` style with customizable icons via `\neoheader`.
- **evalgrade**: `eval` style with an added grading banner.
- **evaliconsgrade**: combines `evalicons` and `evalgrade`.
- **shorteval**: compact style on a single line at the top of the page. Requires `\neoheader`.

3.1.3 Bubble Styles

Visual effects for less formal documents.

- **bubbles**: centered title on a background of colored bubbles.
- **topbubbles**: bubbles confined to the top of the page, title below.
- **shortbubbles**: compact version of `bubbles`.

3.1.4 Other Title Styles

More classic or minimalist options.

- **titleornament**: adds ornaments below the title.
- **titlerule**: adds a small rule below the title.
- **titlemidrule**: adds a medium-width centered rule below the title.
- **titlefullrule**: adds a full-width rule below the title.
- **fancybox**: title in a rotated colored banner at the top left.
- **onlytitleleft** / **onlytitle** / **onlytitleright**: displays only the title, aligned left / center / right. Ignores author, date, etc.
- **shorttitle**: compact style, centered title on a line at the top of the page.
- **shortlesson**: compact style for lecture notes, with type, title, and level on one line. Requires `\neoheader`.

3.2 Header Configuration (`\neoheader`)

This command configures the information used by the `exam`, `eval`, `mockexam`, `shortlesson` styles and the `fullheader` option.

```
\neoheader{
  type = {Quiz}, % Document type
  school = {A. Turing Middle School},   % Institution name
  academy = {Lyon},                     % Academy name (for mockexam)
  level = {Eighth Grade},               % Class level
  duration = {20 minutes},              % Duration (for exam, mockexam)
  calculator = {false},                 % Calculator: true / false / exam
  leftcontent = {\faFlask},             % Left icon (for evalicons)
  rightcontent = {\faCalculator}        % Right icon (for evalicons)
}
```

3.3 Title Formatting Options

These options control the appearance (font, weight, shape, alignment) of various titles in the document.

3.3.1 Global Styles

Apply by default to all titles (sections, theorems, exercises...).

- **headstyle** = `style` (`sffamily` by default): font family (e.g., `rmfamily`, `sffamily`).
- **headweight** = `weight` (`bfseries` by default): weight (e.g., `mdseries`, `bfseries`, `sbseries` for semi-bold if available).

- **headshape** = `shape` (`scshape` by default): shape (e.g., `upshape`, `itshape`, `scshape` for small caps).

3.3.2 Main Title Style (`\maketitle`)

Overrides global styles for the main title.

- **titlestyle** = `style` (inherits from `headstyle`).
- **titleweight** = `weight` (inherits from `headweight`).
- **titleshape** = `shape` (`upshape` by default).
- **titlealign** = `alignment` (`center` by default): `left`, `center`, `right`.

3.3.3 Section Styles (`\section`)

Specific options for section titles.

- **sectionnumstyle** = `style` (`circle` by default): number appearance (`circle`, `box`, `dash`, `plain`).
- **sectiontextstyle** = `style` (`sc` by default): text case (`sc`, `upper`, `lower`).
- **sectionstyle** = `style` (`normal` by default): overall visual style (`ornaments`, `underline`, `normal`, `highlighted`, `shadedline`).
- **sectionalign** = `alignment` (`center` by default): `left`, `center`, `right`.

3.3.4 Header and Footer Styles

- **headfootstyle** = `style` (inherits from `titlestyle`): font style for header and footer text.

3.3.5 Complete Configuration Example

```
\documentclass[
    % Global style
    headstyle=sffamily,
    headweight=bfseries,
    headshape=scshape,
    % Main title
    titlestyle=rmfamily,
    titleweight=bfseries,
    titleshape=upshape,
    titlealign=left,
    % Sections
    sectionnumstyle=box,
    sectiontextstyle=upper,
    sectionstyle=underline,
    sectionalign=left,
```

```
    % Headers/footers
    headfootstyle=sffamily
]{neoschool}
```

4 Content Layout

4.1 Absolute Positioning of Objects

The `\positionobject{x}{y}{scale}{content}` command places *content* at coordinates (x, y) from the top-left corner of the page, with a *scale* factor.

```
% Logo at top right
\positionobject{15cm}{1cm}{0.5}{%
  \includegraphics[width=3cm]{logo.png}%
}

% Text at bottom left
\positionobject{2cm}{25cm}{1.2}{%
  \textit{Important note}%
}
```

4.2 Two-Column Layout

The `\splitcontent[w1][gap]{col1}{col2}` command divides the horizontal space.

- *w1*: width of the first column (default: 0.5 for 50%).
- *gap*: space between columns (default: 0.02 for 2%).
- *col1*, *col2*: column contents.

```
\splitcontent[0.45][0.03]{%
  Column 1 content (45\%)
    \lipsum[1][1-2]
}{%
  Column 2 content (52\%)
    \lipsum[1][1-2]
}
```

Column 1 content (45%) Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Column 2 content (52%) Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

4.3 Side-by-Side Layout (`sidebyside` Environment)

Creates two `tcolorbox` boxes side by side. Use `\tcblower` to switch from the left box to the right. Accepts `tcolorbox` options.

```
\begin{sidebyside}[
  title=\centering
    Comparison,
]
  Pros
  \begin{itemize}
    \item Point 1
  \end{itemize}
\tcblower % Separates the
    two columns
  Cons
  \begin{itemize}
    \item Point A
  \end{itemize}
\end{sidebyside}
```

Comparison

Pros	Cons
• Point 1	• Point A

4.4 Pairing Text and Image

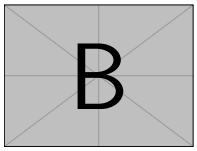
The `\textwithimage[*]{w_img}{s_img}{text}{img_path}` command combines text and an image.

- `*`: if present, places the image on the left (right by default).
- *w_img*: relative width of the image (e.g., 0.3 for 30%).
- *s_img*: image scale within its box (e.g., 0.95).
- *text*: the text.
- *img_path*: path to the image file.

```
% Image on right (30\% width)
\textwithimage{0.3}{1}{%
  Description of the image that
  will be displayed on the right.
}{example-image-a} % placeholder
    image
```

Description of the image that will be displayed on the right.


```
% Image on left (40\% width)
\textwithimage*{0.4}{0.9}{%
  Description on the right of the
    image.
}{example-image-b} % placeholder
    image
```



Description on the right of the image.

4.5 QR Codes and Content

The `\withqrcode[*][size]{url}{content}` command integrates a QR code next to content.

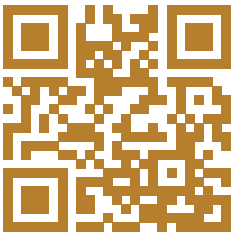
- `*`: places the QR code on the right (left by default).
- *size*: QR code size (default: 2cm).
- *url*: URL or text to encode in the QR code.
- *content*: text or other content to display alongside.

```
% QR code on left (default 2cm)
\withqrcode{https://www.google.com}{
  Scan to visit Google.
}
```

Scan to visit Google.

```
% QR code on right (3cm)
\withqrcode*[3cm]{https://en.
    wikipedia.org}{
  More information on Wikipedia.
}
```

More information on Wikipedia.



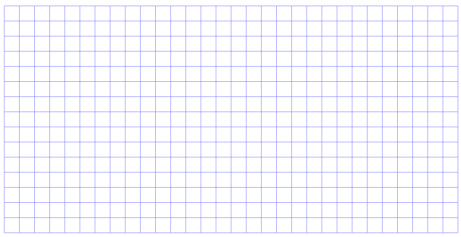
4.6 Grids and Papers

4.6.1 Customizable Grids

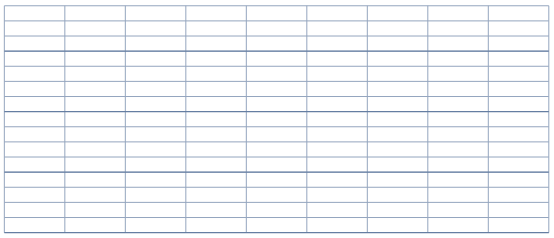
Commands for drawing grid areas.

- `\grid[color]{width}{height}`: small square grid (5mm x 5mm).
- `\customgrid[color][dx][dy]{width}{height}`: grid with custom spacing (*dx*, *dy*).
- `\frenchgrid[small_color][large_color]{width}{height}`: large square grid like Seyès paper. The starred version `\frenchgrid*` centers the grid horizontally.

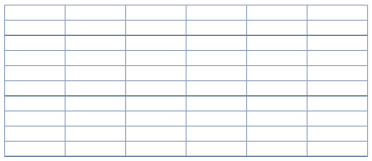
```
\customgrid[blue!50][2mm][2mm]{6cm
    }{3cm}
```



```
\frenchgrid{7.5cm}{3cm}
```



```
\frenchgrid*{5cm}{2cm} % centered
```



4.6.2 Automatic Filling

The `\gridfill` command fills the remaining vertical space on the page with a `customgrid` (5mm by default). The starred version `\gridfill*` uses `frenchgrid`.

```
% Syntax: \gridfill[*][color][dx][dy]

% Fills with a blue customgrid 2mm x 2mm
\gridfill[blue][2mm][2mm]

% Fills with a frenchgrid (default)
\gridfill*
```

4.6.3 Full Pages (Background Styles)

Applies a background style to the current page.

- `\notebook`: notebook style with ruled lines and red margin.
- `\nbminorgrid`: fine grid background (5mm squares).
- `\nbmajorgrid`: large Seyès grid background.

```
\nbminorgrid % Applies fine grid to the entire page
```

4.7 Simple Boxes (`neobox`)

Environment for creating simple `tcolorbox` boxes, with or without frames. Accepts `tcolorbox` options.

- `neobox`: box with frame.
- `neobox*`: box without visible frame (but background possible).

```
\begin{neobox}[
  colframe=red,
  colback=red!10,
  title=Standard Box
]
Content...
\end{neobox}
```

Standard Box

Content...

```
\begin{neobox*}[
  colback=yellow!20,
]
Content without visible frame...
\end{neobox*}
```

Content without visible frame...

5 Exercises

Based on the `xsim` package.

5.1 `exercise` and `solution` Environments

5.1.1 Configuring an Exercise

Options available in `\begin{exercise}[options]`.

- **points** = `number`: number of points.
- **bonus-points** = `number`: bonus points.
- **level** = `number`: difficulty level (1 to 5, displayed as stars ★).
- **subtitle** = `text`: subtitle or brief description.
- **icon** = `fa-icon-name`: FontAwesome icon (e.g., `pencil-alt`). Requires the class option `exerciseicons`.
- **topic** = `theme`: theme/chapter (for future sorting/selection).
- **subject** = `subject`: subject (for sorting/selection).
- **ID** = `identifier`: unique ID to reference the exercise (`\exercisenumber{id}`).

```
\begin{exercise}[
  points=3,
  bonus-points=1,
  level=2,
  subtitle={Application},
  ID=calc01,
  template=sober-box
]
  Calculate $3 \times (5+2)$.
\end{exercise}

\begin{solution}
  $3 \times (5+2) = 3 \times 7 = 21$.

\end{solution}

See exercise \exercisenumber{calc01}.
```

EXERCISE 1 [★★] *Application* (3 POINTS)
Calculate $3 \times (5 + 2)$.

SOLUTION OF EXERCISE 1 $3 \times (5+2) = 3 \times 7 = 21$.

See exercise 1.

5.1.2 Global Options (`\xsimsetup`)

Place in the preamble to affect the entire document.

- **exercise/template** = `template-name`: sets the default style for all exercises.
- **solution/template** = `sol-template-name`: sets the default style for all solutions (e.g., `sol-inline`, `sol-block`, `sol-dotted`).
- **solution/print** = `true/false`: shows or hides solutions globally. Can be overridden by the class option **answers**.

Other class options affecting exercises:

- **exerciseicons**: enables icon display for all exercises.
- **answers** / **answersonly**: displays solutions after each exercise / Displays only solutions.
- **shuffle**: randomly shuffles answers in multiple-choice questions (`choices`, `checkboxes`).
- **sectionthmcounter**: numbers exercises (and theorems) by section (e.g., Ex 1.1, Ex 1.2, Ex 2.1...).
- **sharedexcounter**: shares the same counter between exercises and theorems (combined with `sharedthmcounter`).
- **blocksol**: uses the `sol-block` template for solutions.

5.1.3 Exercise Templates (Display Styles)

Possible values for `template` (exercise option or `xsimsetup`).

- **box**, **elegant-box**, **shaded-box**, **slanted-box**, **sober-box**, **classic-box**, **classy-box**: various framed box styles with titles.
- **rect-box**, **rect-box-out**: simple rectangular boxes, with or without outline.
- **num-box**, **num-box-out**: compact boxes showing just the number.
- **ex-num-box**, **ex-num-box-out**: like `num-box` but with "Ex." before the number.
- **box-hrule**, **box-hrule-out**, **box-hrule-in**: title in a box followed by a horizontal rule.
- **boxed**, **boxed-out**: title in a small box (filled or outlined).
- **inline** (default if `amslikethm`): simple inline style, like a paragraph.
- **section**, **subsection**: style like a section/subsection title.
- **terminal**: style mimicking a console, with icon `>_`.
- **block**: minimalist style without box or decoration.
- **hrule**: style with a simple horizontal rule under the title.

```
\xsimsetup{exercise/template=box-
    hrule-out}
\begin{exercise}[points=2]
 Style box-hrule-out.
\end{exercise}
\xsimsetup{exercise/template=inline}
     % Back to default
```

EXERCISE 2 <<< 2 POINTS

Style box-hrule-out.

5.1.4 Multiple-Choice and Checkboxes

Environments to use inside an exercise.

- **choices** `(n)`: for single-answer multiple-choice. `\choice` for an answer, `\choice[\correct]` for the correct one. *n* is the number of columns.
- **checkboxes** `(n)`: for multiple-answer multiple-choice. `\checkbox` for an answer, `\checkbox[\correct*]` for a correct one. *n* is the number of columns.

```
\begin{exercise}[ID=qcm-deriv,
    points=1]
 What is the derivative of $f(x)=x
    ^2$?
 \begin{choices}(2) % On 2 columns
  \choice[\correct] $x\mapsto 2x$
  \choice $x\mapsto x$
  \choice $x\mapsto x^2$
  \choice $x\mapsto 2$
 \end{choices}
\end{exercise}
```

Exercise 3 (1 point) What is the derivative of $f(x) = x^2$?

- ○ $x \mapsto 2x$
- ○ $x \mapsto x$
- ○ $x \mapsto x^2$
- ○ $x \mapsto 2$

```
\begin{exercise}[ID=qcm-props, points
    =2]
 Check the true properties.
 \begin{checkboxes}(1) % On 1 column
  \checkbox[\correct*] $1+1=2$
  \checkbox $\sqrt{2}$ is rational.
  \checkbox[\correct*] $0 \times 5 = 0$
 \end{checkboxes}
\end{exercise}
```

Exercise 4 (2 points) Check the true properties.

- ☐ $1 + 1 = 2$
- ☐ $\sqrt{2}$ is rational.
- ☐ $0 \times 5 = 0$

6 Mathematical Environments (Theorems, Definitions, etc.)

Based on `tcolorbox`.

6.1 Theorem Styles (Class Options)

These class options define the default appearance of all theorem-like environments.

- **classythm**, **soberthm**, **elegantthm**, **classicthm**, **slantedthm**, **shadedthm**, **boxedthm**: varied styles with boxes, colors, etc.
- **amslikethm** (default): minimalist style similar to standard AMS environments.

```
% Applies the \texttt{elegantthm} style to all mathematical environments.
\documentclass[elegantthm]{neoschool}
```

6.2 Common Environment Options

Passed in brackets: `\begin{theorem}[options]`.

- **title** = `text`: custom title (e.g., `title=Pythagorean Theorem`).
- **label** = `name`: label for referencing (`\ref`, `\nameref`).

- **colback** = `color`: specific background color.
- **colframe** = `color`: specific frame color.
- **coltitle** = `color`: specific title color.
- **fonttitle** = `commands`: formatting commands for the title (e.g., `fonttitle=\sffamily\bfseries`).

6.3 Numbering Options (Class Options)

- **sectionthmcounter**: counters are relative to each section (e.g., Thm 1.1, Def 1.2, Thm 2.1...).
- **sharedthmcounter**: a single counter shared among all environments (Thm 1, Def 2, Lem 3...).
- **theoremgroup**: groups certain environments (theorem, lemma, corollary, proposition, property) under the same counter if not already shared by `sharedthmcounter`.
- **thmgroupcounter**: enables a specific counter for the `theoremgroup`.

6.4 Available Mathematical Environments

- **theorem** (reference: `thm`)
- **lemma** (reference: `lem`)
- **corollary** (reference: `cor`)
- **proposition** (reference: `propo`)
- **property** (reference: `prop`)
- **definition** (reference: `def`)
- **method** (reference: `meth`)
- **activity** (reference: `act`)
- **application** (reference: `appl`)
- **remark** (unnumbered)
- **remarks** (unnumbered, for multiple remarks)
- **example** (reference: `ex`)
- **examples** (unnumbered, for multiple examples)
- **proof**: environment for proofs.

```
\begin{definition}[
  title=Prime Number,
  label=prime
]
A natural number is called prime
if it has exactly two distinct
    divisors:
1 and itself.
\end{definition}

See definition~\ref{def:prime}.
```

Definition 1 (Prime Number) A natural number is called prime if it has exactly two distinct divisors: 1 and itself.

See definition 1.

```
\begin{proof}
  The proof is straightforward and
    left
  as an exercise for the reader.
\end{proof}
```

Proof. The proof is straightforward and left as an exercise for the reader. □

7 Computer Code

Two systems are available via the class options **listings** (default) or **minted**.

7.1 `listings` Option

Uses the `listings` package.

7.1.1 Available Code Styles

The class option **lststyle** = `style` sets the default coloring style for the `code` environment when the `listings` option is active. Basic styles are:

- `colorful`: full syntax highlighting. This is the base for the default style.
- `minimal`: very simple style, minimal colors, mainly bold/normal.
- `academic`: style with line numbers and a vertical line on the left.
- `modern`: style with lightly colored background and simple frame.

The `colorful` and `minimal` styles have variants by adding suffixes:

- `b`: adds a colored background (`backgroundcolor=\color{codebackColor}`).
- `n`: adds line numbers on the left (`numbers=left`). The default for `lststyle` is actually `colorfuln`.
- `f`: adds a simple frame around the block (`frame=single`).

These suffixes can be combined (in alphabetical order), e.g., `colorfulbfn` (colored background, frame, and numbers), `minimalbn` (minimalist with background and numbers), etc.

7.1.2 Changing the Default Style Mid-Document

The `\setcodestyle{style}` command changes the default `listings` style for all subsequent `code` environments in the document. The *style* names are the same as those used for the `lststyle` option (including variants with `b`, `n`, `f`).

```
\begin{code}{python}
# Default style (colorfuln)
def fact(n):
    if n ==0:
        return 1
    return n * fact(n - 1)
\end{code}

\setcodestyle{minimal} % Changes
    the style

\begin{code}{python}
# Minimal style applied
def fact(n):
    if n ==0:
        return 1
    return n * fact(n - 1)
\end{code}

\setcodestyle{colorfuln} % Back to
     initial style
```

```
# Default style (colorfuln)
def fact(n):
    if n ==0:
        return 1
    return n * fact(n - 1)


# Minimal style applied
def fact(n):
    if n ==0:
        return 1
    return n * fact(n - 1)
```

7.1.3 `code` Environment

Displays a code block. The `code` environment shows line numbers by default.

```
\begin[options]{code}{language}[title][box-style]
... code ...
\end{code}
```

- `options`: `listings` options (e.g., `numbers=none`, `frame=single`).
- `language`: code language (e.g., `python`, `latex`, `c++`...).
- `title`: optional title for the box.
- `box-style`: `tcolorbox` box style (see below).
- `code*`: variant without automatic listing numbering.

Available box styles: `box-minimal`, `box-fancy`, `box-elegant`, `box-diagonal`, `box-bevel`, `box-corner`, `box-rounded`, `box-downhill`, `box-bottomtitle`, `box-bottomtitlef`.

```
\begin{code}{python}[Square
    Function][box-elegant]
def square(x):
  """Computes the square of x."""
  return x * x
\end{code}
```

Listing 30 — Square Function

```python
def square(x):
  """Computes the square of x."""
  return x * x
```

7.1.4 Preconfigured Languages

`Python`, `Java`, `C++`, `JavaScript`, `SQL`, `LaTeX`, `Bash`, `Assembly`, `Lisp`, `JSON`, `YAML`, `TOML`, `CSV`, `Markdown`.

7.1.5 Additional Commands

- `\codeinline[lang]{code}`: displays *code* inline with highlighting (*lang* optional).
- `\codeinput[opt]{lang}{file}[title][style]`: imports and displays code from a *file*.

```
The code \codeinline[python]{x = 5}
    initializes x.
```

The code `x = 5` initializes x.

7.2 `minted` Option

Uses the `minted` package, requiring Python and Pygments installed, along with compilation using the `-shell-escape` option. Offers richer syntax highlighting.

The `code`, `code*`, `\codeinline`, and `\codeinput` work similarly, but options passed are those of `minted` (e.g., `linenos`, `highlightlines={2,3}`, `style=tango`). Box styles remain the same.

```
\documentclass[minted]{neoschool}
...
\usemintedstyle{friendly}
\begin{code}[linenos]{python}[
    Minted Example]
import math

def circle_area(radius):
  return math.pi * radius**2
\end{code}
```

Listing 32 — Minted Example

```python
import math

def circle_area(radius):
  return math.pi * radius**2
```

7.3 Algorithms and Pseudocode

The class provides a `pseudocode` environment based on `algpseudocode` and `tcolorbox` for displaying algorithms. It is automatically numbered and titled.

```
\begin{pseudocode}{Example Algorithm
    }
\Require $n \ge 0$
\Ensure $y = x^n$
\State $y \Gets 1$
\If{$n < 0$}
    \State $X \Gets 1/x$
    \State $N \Gets -n$
\Else
    \State $X \Gets x$
    \State $N \Gets n$
\EndIf
\While{$N \neq 0$}
    \If{$N$ is even}
        \State $X \Gets X \times X$
        \State $N \Gets N/2$
    \Else[$N$ is odd]
        \State $y \Gets y \times X$
        \State $N \Gets N-1$
    \EndIf
\EndWhile
\end{pseudocode}
```

Algorithm 34 — Example Algorithm

entrée $n \geq 0$
sortie $y = x^n$
  $y \leftarrow 1$
  si $n < 0$ **alors**
    $X \leftarrow 1/x$
    $N \leftarrow -n$
  sinon
    $X \leftarrow x$
    $N \leftarrow n$
  fin si
  tant que $N \neq 0$ **faire**
    si N is even **alors**
      $X \leftarrow X \times X$
      $N \leftarrow N/2$
    sinon[N is odd]
      $y \leftarrow y \times X$
      $N \leftarrow N - 1$
    fin si
  fin tant que

8 Notes and Annotations

8.1 Margin Notes (`todonotes`)

Enabled by the class option `notes=length`.

- `\tdnote[options]{text}`: Creates a note in the margin. Notes alternate between left/right sides. Options (`backgroundcolor`, `linecolor`, etc.) are those of `todonotes`.
- `\boxnote[label]{text}` + `\tdmark[label]`: Places a `tdnote` (`boxnote`) anchored at a specific point (`tdmark`) even inside environments where `todonotes` fails. The *label* must match.

```
This is some text\tdnote{Important point!}.

\boxnote[thm-imp]{Don't forget this assumption.}
\begin{theorem}
  \tdmark[thm-imp] If $x>0$, then...
\end{theorem}
```

8.2 Admonitions

Colored boxes with icons to draw attention to specific points.

- `note`: general remarks ().
- `info`: additional information ().
- `warning`: warnings ().
- **important**: key points ().
- **tip**: tips, tricks ().
- **reminder**: to remember, memorize ().
- **summary**: summary, in brief ().
- **toolbox**: list of materials, prerequisites ().

Each environment accepts `[Optional Title][Optional Icon]`. The class option **inlineadmonition** displays the title and content on the same line.

```
\begin{warning}[Caution][\faSkull]
  Do not divide by zero!
\end{warning}

\begin{tip}
  Remember to factor first.
\end{tip}
```

Caution

Do not divide by zero!

Tip

Remember to factor first.

9 Grading and Feedback

9.1 Grading Tools

- `\gradingstrip[total]`: displays a banner for the grade and comments. *total* optional overrides the total set by the class option `totalpoints` (default: 20).
- `\mrk[*][comm]{pts}`, `\mrks[*][comm]{pts}`: displays (*pts* pt) or (*pts* pts) in the margin. `*` places on left, *comm* adds a comment.

```
\gradingstrip[10] % Scale
    out of 10

Question 1 \mrks*[Details
    of the grading here...
     lorem ipsum dolor sit
     amet]{3}
```

Mark	Comments
10	

Question 1

(3 pts) Details of the grading here... lorem ipsum dolor sit amet

9.2 Answer Areas

- `\answerfield[width]{lines}` : creates a colored background answer area, with height of *lines* text lines and *width* (default: `0.975\linewidth`).
- `\answerframe[width]{lines}[options ]` : creates a framed answer area, with height of *lines* and *width* (default: `\linewidth`).
- `\vardots[length]` : draws a dotted line of *length* (default: `\linewidth`).

```
Answer:

\answerfield[6cm]{1}
```

Answer:

```
Signature: \vardots[4cm]
```

Signature: .....

9.3 Markers and Symbols

- `\cmark` : ✓ (green).
- `\xmark` : × (red).
- `\unchecked` : ☐ (for lists).
- `\done` : ☑ (for lists).
- `\wontfix` : ☒ (for lists).

```
\begin{itemize}
 \unchecked Task 1
 \done Task 2
 \wontfix Task 3
\end{itemize}
```

- ☐ Task 1
- ☑ Task 2
- ☒ Task 3

9.4 Skills-Based Assessment

`\competencies{Skill1\\Skill2...}` : creates a table to evaluate skills across 4 mastery levels (with emojis).

```
\competencies{
  Master fractions \\
  Solve a first-degree
    equation \\
  Compute a derivative
}
```

Competencies				
Master fractions				
Solve a first-degree equation				
Compute a derivative				

10 Mathematical Commands and Special Tools

10.1 Mathematical Commands

10.1.1 Highlighting and Coloring

- `\mhl[color]{expr}`: highlights the mathematical expression *expr* with *color* (default: theme-specific color).
- `\mc[color]{expr}`: colors the mathematical expression *expr* with *color* (default: theme-specific color).
- `\mathbox<background>[border]{content}`: quickly frames mathematical *content* in a `tcolorbox`. The *background* color (default: white) and *border* (default: theme-specific color) are customizable.

```
$f(x) = \mhl[cyan!20]{x^2} + \mc[blue]{3x} - 1$
```

$f(x) = x^2 + 3x - 1$

```
$\mathbox{E=mc^2}$ % White background, theme border

$\mathbox<yellow!20>[red]{a^2+b^2=c^2}$ % Custom
```

$E = mc^2$

$a^2 + b^2 = c^2$

10.1.2 APMEP Support

Commands available with the class option `apmep`.

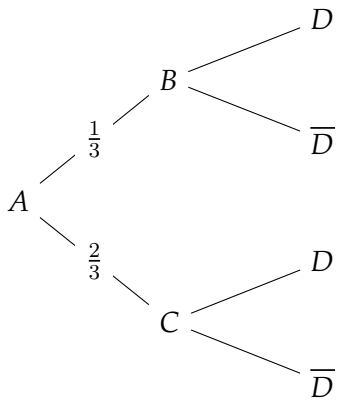
- Vectors: `\vect{u}`, `\vectt{AB}`.
- Coordinate Systems: `\Oij`, `\Oijk`, `\Ouv`.
- Symbols/Commands: `\euro` (€), `\cg` (∥), `\cd` (⊥), `\pg` (≥), `\pp` (≤), `\barre{x}` ($\overline{x}$), `\ds` (`\displaystyle`).

10.2 Special Tools

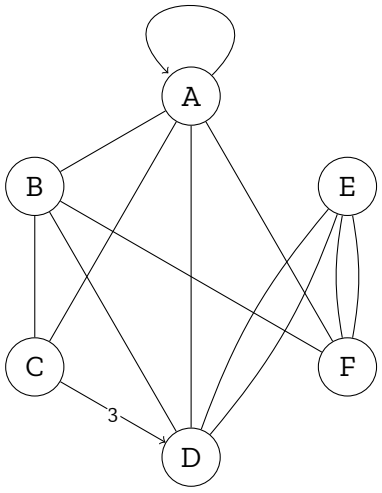
10.2.1 Trees and Graphs

- **Trees** (`neotree` environment): uses `forest`. Syntax is that of `forest`, with options passed to the environment. The `w=val` option adds a weight to a branch.
- **Graphs** (`\neograph` command, requires `lualatex`): uses TikZ `graphs`. Syntax is TikZ's for graphs.

```
\begin{neotree}[l=2cm, s sep=1cm]
  A [B, w=\frac{1}{3} [D] [\overline{D
    }]]
    [C, w=\frac{2}{3} [D] [\overline{D
    }]]
\end{neotree}
```




```
% Requires lualatex
\neograph{
  A -- {B, C, D, F},
  B -- {C, D, F},
  C -> ["3"] D,
  D -- [bend left=10] {E},
  E -- [bend left=10] {D},
  E -- [bend left=10] {F},
  F -- [bend left=10] {E},
  A -- [loop] A
}
```



10.2.2 Math Grid (`mathgrid`)

Environment for aligning blocks of equations (`align*`) in a grid.

- `\begin{mathgrid}{n}`: starts a grid with *n* columns.
- `\neoline`: starts a new row in the grid.
- `\neocol[span]{content}`: adds a column containing *content* (an `align*` block). *span* (optional, default 1) indicates how many columns the content spans.

```
\begin{mathgrid}{2} % 2
    columns
 \neoline % Row 1
 \neocol{ % Col 1
  A &= 1+1 \\  A &= 2
 }
 \neocol{ % Col 2
  B &= 3 \times 4 \\ B &= 12
 }
 \neoline % Row 2
 \neocol[2]{ % Single column (
    span=2)
  C &= A+B \\ C &= 2+12 \\ C
    &= 14
 }
\end{mathgrid}
```

$$A = 1 + 1 \qquad B = 3 \times 4$$
$$A = 2 \qquad B = 12$$
$$C = A + B$$
$$C = 2 + 12$$
$$C = 14$$