

Mini manuel de la classe Prepamath

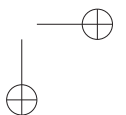
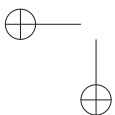


Table des matières

Chap **1** Organisation d'un document

- Général
- Détails

1
1
7

Introduction

Il s'agit d'un mini manuel de la classe prepamath, un manuel détaillé fait également partie des fichiers fournis. En plus de ces deux manuels, une première (bonne) approche est de lire le fichier source de test en regardant le pdf produit.

En cas de problème, j'assure une hotline technique aux adresses suivantes :

- Jean-Come.Charpentier@wanadoo.fr ;
- 05 46 06 22 37 ou 06 37 25 17 60 (uniquement en cas d'urgence).

Installation du système

Le fichier `premapath-final.zip` contient tous les packages et tout le système des fontes commerciales qui sont utilisés par la classe Prepamath. C'est un fichier destiné aux metteurs en page.

Le fichier `premapath-auteur.zip` contient exactement la même chose mais sans les fichiers relatifs aux fontes commerciales. Si vous avez un système $\text{\TeX}$ suffisamment récent¹, ce fichier zip, une fois décompressé, ne servira qu'à obtenir le fichier de classe `prepamath.cls`, les fichiers de tests et les manuels. En d'autres termes, pour les auteurs ayant une distribution $\text{\TeX}$ assez récente, le paragraphe suivant peut être sauté sans problème. En ce qui concerne le fichier de classe, soit on le copie dans le répertoire où se situera le document maître, soit on le copie dans l'arborescence $\text{\TeX}$. L'emplacement logique sera :

`[TEXMF]/tex/latex/prepamath/`

`[TEXMF]` représente la racine de l'arborescence $\text{\TeX}$. Le répertoire `prepamath` sera à créer.

L'extraction du fichier `prepamath-auteur.zip` donne un répertoire TDS qui contient l'ensemble des fichiers utilisés par `prepamath` organisés selon la TDS ($\text{\TeX}$ Directory Structure). En pratique, cela veut dire qu'il faudrait plutôt extraire ce fichier en étant placé à la racine d'une arborescence $\text{\TeX}$ personnelle et surtout pas en étant placé à la racine d'une arborescence $\text{\TeX}$ globale où serait déjà installé un autre système $\text{\TeX}$... sinon, vous allez perdre un certains nombres de fichiers et l'auteur du présent fichier zip déclinera toute responsabilité !

Installation des fontes commerciales

Le fichier `premapath-final.zip`, une fois décompressé, donne exactement les mêmes répertoires que ceux produits par `prepamath-auteur.zip` avec en plus un répertoire `fontes` où seront disposés tous les fichiers relatifs à la gestion des

1. En pratique, une distribution datant de 2005 ou plus ne posera pas de problème.

fontes commerciales organisés là aussi selon la TDS. Le risque est moindre avec ce répertoire puisque toute cette machinerie n'existe pas sur une arborescence TEXMF habituelle mais il est préférable de placer ces fichiers dans une arborescence TEXMF personnelle ou locale (pas dans l'arborescence principale du système).

Une fois les fichiers installés, il faudra (ou non selon la distribution $\text{T}_{\text{E}}\text{X}$) rafraîchir la base de données des fichiers $\text{T}_{\text{E}}\text{X}$ (commandes, `mktexlsr`, `texhash` ou apparentées) et, pour les fontes commerciales, rafraîchir la base de données des fichiers `map` (commande `updmap` ou apparentée). En ligne de commande, il faudra réaliser un :

```
updmap --enable Map=block.map  
updmap --enable Map=FF-DIN.map  
updmap --enable Map=helvetica-rounded.map  
updmap --enable Map=mathtime.map
```

ou `updmap-sys` à la place de `updmap` selon l'endroit où les fichiers ont été placés.

Options de la classe

Les points les plus importants à connaître au tout début sont :

- La classe accepte des options pour compiler des sources sauvegardés aux codages `latin1`, `latin9`, `ascii`, `ansinews`, `applemac` et `utf8`. Si on ne spécifie rien, la classe suppose l'utilisation du codage `utf8`.
- Pour une compilation finale, donc avec les fontes commerciales, il faut spécifier l'option de classe `realfonts` ;
- Pour avoir les repères photographiques (normalement obligatoires avant flashage, il faut mettre l'option de classe `crop` ;
- Comme la classe fait appel à `PSTricks` pour tout ce qui est graphique, la chaîne de compilation est obligatoirement `latex`, `dvips`, `ps2pdf` (ou un autre mais qui commence obligatoirement par `latex` et surtout pas par `pdflatex`. Si on veut absolument passer par une compilation par `pdflatex` il faudra compiler avec l'option `nopstricks`.
- La classe permet de produire un pdf avec des liens hypertextes. Cela n'a aucun intérêt si le document final doit être imprimé mais sinon on doit utiliser l'option de classe `hyperref`.

Chapitre

1

Organisation d'un document

Plan du chapitre

1. Éléments administratifs
2. Parties d'un ouvrage
3. Commandes particulières
4. Tableaux
5. Commandes diverses
6. Objets NathanLive

1 Éléments administratifs

Un certain nombre de précisions doivent être apportées à un ouvrage de la collection. Deux sont obligatoires, les autres optionnelles. Ces précisions se font avec des macros (parfois classiques) au niveau du préambule du document.

- Le titre de l'ouvrage est indiqué avec `\title{<titre>}`. Il s'agit d'une indication obligatoire.
- On indique le (ou les) auteur(s) indiqué(s) avec `\author{<nom-auteur>}` qui a la même syntaxe que la commande classique. Il s'agit là aussi d'une indication obligatoire.
- On peut préciser une date avec la commande `\date{<date>}`. Si on n'utilise pas cette commande, c'est la date de compilation qui sera prise en compte.
- On peut également préciser un numéro de version du document grâce à la commande `\version{<numéro-version>}`. La valeur par défaut est 1.0.
- En fin de travail, on doit indiquer un numéro d'ISBN. Ce numéro étant inconnu pendant la majeure partie du travail, il n'est pas obligatoire d'utiliser la commande `\ISBN{<numéro-ISBN>}`.

2 Parties d'un ouvrage

2.1 Commandes de sectionnement

Comme avec la classe `book`, on dispose des commandes `\frontmatter` qui indique l'avant-propos de l'ouvrage et `\mainmatter` pour sa partie principale. Pour l'instant, il ne faut pas utiliser `\backmatter`.

La partie `\frontmatter` contient les pages de garde, les remerciements de l'auteur et de l'éditeur (fichier `avant-propos.tex`), la table des matières (obtenue avec `\tableofcontents`) et le texte sur les méthodes de travail (fichier `methode.tex`).

La partie `\mainmatter` contient tous les chapitres, y compris les chapitres spéciaux pour le bac blanc, le brevet blanc ou le devoir de synthèse. Ces chapitres spéciaux utilisent la commande `\devoirsynthese` avec en premier argument le type de chapitre spécial et en deuxième argument le titre. Par exemple, pour un bac blanc, cela donnera :

```
\specialchapter{Bac blanc}  
                {Baccalauréat 2009, académie de Nancy}
```

Chaque chapitre commence par la commande `\chapter` dont l'argument est le titre. Comme dans les classes standard, on a accès aux commandes de sectionnement habituelles (`\section`, `\subsection`, etc.). Toutefois, la commande de hiérarchie la plus basse (`\subparagraph`) ne devrait pas être utilisée. Le paramètre optionnel de la commande `\chapter` a un rôle tout à fait spécial : n'en faites pas usage ou lisez le manuel complet. Les autres commandes de sectionnement n'ont pas de paramètre optionnel (ni de version étoilée). On peut, si besoin, utiliser la commande `\part` pour regrouper plusieurs chapitres d'un même thème.

En plus des commandes de sectionnement habituelles, la classe `prepamath` définit le niveau de sectionnement `\thumb` (ou `\onglet`). Il est indépendant des niveaux à l'intérieur d'un chapitre et sert à changer d'onglet. Dans les chapitres normaux, il y a trois onglets : un pour le cours, un pour les énoncés des exercices et un pour les corrigés de ces exercices. La commande permet de passer à l'onglet suivant.

Il est également possible d'indiquer le ou les thèmes du chapitre en utilisant la commande `\thema`. L'argument de cette commande indique les noms des thèmes, séparés par des virgules.

On peut également, en toute fin d'ouvrage, avoir des annexes. Pour cela, il faudra indiquer la commande `\appendix` avant les chapitres concernés. Si une annexe devait faire une ou deux pages, il faudrait abandonner la structure habituelle des chapitres et utiliser la commande `\simplechapter`.

Parmi les annexes, on peut disposer d'un index. Celui-ci s'obtient avec les commandes usuelles de $\text{\LaTeX}$ c'est-à-dire `\index` pour indexer un mot (avec la syntaxe habituelle) et `\printindex` pour l'afficher. Il faut utiliser le fichier de style `prepamath.ist` lors de l'élaboration de l'index par l'outil externe (par exemple `makeindex`).

ORGANISATION D'UN DOCUMENT • CHAP. 1

2.2 Exercices

Un chapitre normal commence obligatoirement par un exercice introductif (ou plusieurs éventuellement), également appelé exercice-type. Cet exercice est donc dans la partie cours. On le compose à l'intérieur de l'environnement `exointro`. Normalement, on donne sa solution à la fin de la partie cours à l'intérieur de l'environnement `solintro`.

L'environnement `exointro` demande un argument obligatoire qui est le nom de l'établissement où l'exercice a été posé.

Par exemple, le code :

```
\begin{exointro}{Lycée Lacourbe}
  Combien vaut  $2+2$  ?
\end{exointro}
...
\begin{solintro}
   $2+2=4$ .
\end{solintro}
```

donnera le résultat suivant :

Exercice type

Lycée Lacourbe

Combien vaut $2 + 2$?

Voir corrigé page 3

→ Solution de l'exercice type

Lycée Lacourbe

$2 + 2 = 4$.

Voir énoncé page 3

Dans le cas où il y a plusieurs exercices-type, il faudra que les solutions soient données dans le même ordre que les énoncés.

Une fois la partie cours terminée, une commande `\onglet` (ou `\thumb`), fait passer à la partie énoncé des exercices. Normalement, on commence par poser les QCM puis les exercices classiques.

Les QCM sont saisis individuellement à l'intérieur d'un environnement `qcm`. Cet environnement demande deux arguments : le titre du QCM et la durée supposée, en minutes, que prendra l'élève pour y répondre.

Pour les QCM, une commande très utile est `\choix` qui affiche une lettre dans un cadre. Chaque appel de la commande fera passer à la lettre suivante. Ainsi, le code :

```
\choix\quad \choix
```

donnera le résultat : **a** **b** .

GÉNÉRAL

DÉTAILS

En cas de besoin, on peut toujours spécifier ce qui doit être affiché dans le cadre avec l'argument optionnel de la commande.

Parfois, l'utilisation des commandes `\choix` a lieu dans un environnement si bien que la numérotation reprendra à zéro lorsqu'on sera dans un nouvel environnement. Toutefois, assez souvent il faut reprendre la numérotation dans un même environnement, on utilise alors la commande `\resetchoix`. C'est par exemple le cas, lorsqu'on a plusieurs questions dans un même exercice et que chaque question propose un choix de plusieurs réponses possibles.

Les exercices classiques sont saisis à l'intérieur d'un environnement `exo`. Cet environnement demande quatre arguments :

- 1 le titre ;
- 2 l'établissement de provenance ;
- 3 la difficulté (de 1 à 3) ;
- 4 la durée de résolution (en minutes sans spécifier l'unité)

Après chaque QCM et chaque exercice, il faut saisir la solution à l'intérieur d'un environnement `corrige`. Les solutions sont composées grâce à la commande `\AfficheCorrige`. Cette commande doit normalement être précédée d'une commande `\onglet` puisque les pages des solutions sont dans la dernière partie des chapitres.

Le code suivant :

```
\begin{qcm}{Vrai ou faux}{1}
  Indiquer si les affirmations suivantes sont vraies ou
  fausses :
  \begin{enumerate}
    \item 119 est un nombre premier.
    \item 120 est un nombre premier.
  \end{enumerate}
\end{qcm}
\begin{corrige}
  \begin{enumerate}
    \item  $119 = 17 \times 7$  donc 119 n'est pas premier.
    \item 120 a beaucoup de diviseurs : il n'est donc pas
    premier.
  \end{enumerate}
\end{corrige}
\begin{exo}{Recherche de pgcd}{École d'Athènes}{1}{2}
  Calculer  $\text{PGCD}(12;90)$ .
\end{exo}
\begin{corrige}
   $90 = 7 \times 12 + 6$  puis  $12 = 2 \times 6 + 0$ 

  Le dernier reste non nul est 6 donc  $\text{PGCD}(12;90)=6$ .
\end{corrige}
```

ORGANISATION D'UN DOCUMENT • CHAP. 1

Voici la correction des exercices
`\AfficheCorrige`
donne le résultat :

GÉNÉRAL

DÉTAILS

1 **QCM** **Vrai ou faux**

1 min

→ Corrigé
p. 6

Indiquer si les affirmations suivantes sont vraies ou fausses :

- 1 119 est un nombre premier.
- 2 120 est un nombre premier.

2 **Recherche de pgcd**



2 min

→ Corrigé
p. 6

École d'Athènes

Calculer $\text{pgcd}(12; 90)$.

Voici la correction des exercices

1 **QCM** **Vrai ou faux**

→ Énoncé
p. 6

- 1 $119 = 17 \times 7$ donc 119 n'est pas premier.
- 2 120 a beaucoup de diviseurs : il n'est donc pas premier.

2 **Recherche de pgcd**

→ Énoncé
p. 6

École d'Athènes

$$90 = 7 \times 12 + 6 \text{ puis } 12 = 2 \times 6 + 0$$

Le dernier reste non nul est 6 donc $\text{pgcd}(12; 90) = 6$.

2.3 **Organisation des fichiers**

Théoriquement, on peut tout à fait n'avoir qu'un seul fichier source pour l'ensemble d'un ouvrage mais en pratique, on le déconseille fortement pour des raisons de lisibilité et de facilité de gestion. Pour les ouvrages de la collection, on propose que tout le monde organise un ouvrage en un fichier maître qui ne fait qu'appeler les autres fichiers avec des commandes `\include`² et tous les fichiers appelés dans un sous-répertoire (`src` dans la structure proposée en exemple). Le découpage se fait en deux fichiers par chapitre : un fichier pour la partie cours et un fichier pour les exercices (les corrections n'étant que la suite d'instructions `\onglet` et `\AfficheCorrige`).

2. On pourrait le faire avec des commandes `\input` mais on perd certaines possibilités.

ORGANISATION D'UN DOCUMENT • CHAP. 1

3 Commandes particulières

3.1 Listes

On retrouve exactement les environnements standards des classes $\text{\LaTeX}$ habituelles (`itemize`, `enumerate` et `description`). Il est interdit de modifier leur comportement.

3.2 Théorèmes

La classe `prepamath` offre de nombreux environnements dédiés aux théorèmes et aux structures apparentées. On a ainsi accès à :

- `\begin{thm}... \end{thm};`
- `\begin{thms}... \end{thms};`
- `\begin{coro}... \end{coro};`
- `\begin{coros}... \end{coros};`
- `\begin{defin}... \end{defin};`
- `\begin{defins}... \end{defins};`
- `\begin{prop}... \end{prop};`
- `\begin{props}... \end{props};`

pour respectivement un théorème, des théorèmes, un corollaire, des corollaires, une définition, des définitions, une proposition, des propositions.

Il existe des environnements donnant une composition plus discrète avec :

- `\begin{Remarque}... \end{Remarque};`
- `\begin{Remarques}... \end{Remarques};`
- `\begin{Exemple}... \end{Exemple};`
- `\begin{Exemples}... \end{Exemples};`

dont les significations sont immédiates.

Tous les environnements de ces deux grands types acceptent un argument optionnel qui permet de spécifier un titre qui sera ajouté entre parenthèses.

Il existe enfin cinq environnements donnant des cadres pour attirer l'attention. Il s'agit de :

- `\begin{problematique}... \end{problematique};`
- `\begin{aretenir}... \end{aretenir};`
- `\begin{methode}... \end{methode} ne;`
- `\begin{attention}... \end{attention};`
- `\begin{anecdote}... \end{anecdote};`

GÉNÉRAL

DÉTAILS

4 Tableaux

La syntaxe des tableaux est celle permise par les packages `array`, `colortbl`, `cellspace` et `tabularx`.

Les tableaux des ouvrages de la collection devraient toujours avoir une ligne de titre obtenue grâce à la commande `\tabulartitle` saisie au début de la première cellule. Si besoin, la couleur utilisée pour les tableaux est `TabularColor`. La classe permet également de spécifier des colonnes de titres avec les motifs de tableau `L`, `R`, `C`, `P`, `M`, `B`, `x` qui sont les équivalents des types `l`, `r`, `c`, `p`, `m`, `b`, `X` avec la même syntaxe d'appel mais avec la couleur de colonne `TabularColor`.

5 Commandes diverses

La classe `prepamath` propose un certain nombre de raccourcis mathématiques :

Macro		Signification
<code>\nep</code>	<code>e</code>	Base du logarithme népérien. On le compose en romain.
<code>\im</code>	<code>i</code>	Base de la partie imaginaire des nombres complexes.
<code>\ch</code>	<code>ch</code>	Cosinus hyperbolique
<code>\sh</code>	<code>sh</code>	Sinus hyperbolique
<code>\Re</code>	<code>Re</code>	Partie réelle des nombres complexes (redéfinition)
<code>\Im</code>	<code>Im</code>	Partie imaginaire des nombres complexes (redéfinition)
<code>\Arg</code>	<code>arg</code>	Argument d'un nombre complexe
<code>\dd</code>	<code>d</code>	Différentiel (composé en romain)
<code>\card</code>	<code>card</code>	Cardinal d'un ensemble
<code>\PGCD</code>	<code>pgcd</code>	Plus grand diviseur commun
<code>\PPCM</code>	<code>ppcm</code>	Plus petit multiple commun
<code>\esp</code>	<code>E</code>	Espérance mathématique
<code>\var</code>	<code>V</code>	Variance
<code>\ent</code>	<code>E</code>	Partie entière
<code>\Moy{x}</code>	$\bar{x}$	Moyenne
<code>\gvec{AB}</code>	$\overrightarrow{AB}$	Vecteur
<code>\N</code>	$\mathbb{N}$	Ensemble des entiers naturels
<code>\Z</code>	$\mathbb{Z}$	Ensemble des entiers relatifs
<code>\Q</code>	$\mathbb{Q}$	Ensemble des rationnels
<code>\R</code>	$\mathbb{R}$	Ensemble des réels
<code>\C</code>	$\mathbb{C}$	Ensemble des complexes

Il existe également toutes les commandes disponibles grâce aux packages de l'AMS.

La classe donne la possibilité de simuler des touches de calculatrices avec les commandes `\T` (à contenu textuel) et `\Tm` (à contenu mathématique). Par exemple :

ORGANISATION D'UN DOCUMENT • CHAP. 1

`\T{ABC} \Tm{\mu}`

donne le résultat **ABC** μ .

Pour éviter une frappe fastidieuse, la classe offre les raccourcis `\TTA` à `\TTZ` pour les lettres de A à Z. On peut également accéder aux chiffres très rapidement avec la syntaxe `\T0` à `\T9`. Enfin, certaines touches classiques de calculatrices ont été prévues :

<code>\Tpourcent</code> %	<code>\Tcos</code> COS	<code>\Tcalc</code> CALC	<code>\Tdiv</code> ÷
<code>\Tcarre</code> x²	<code>\Ttan</code> TAN	<code>\Trcl</code> RCL	<code>\Tplus</code> +
<code>\Tcube</code> x³	<code>\Tasin</code> ASIN	<code>\Tmem</code> MEM	<code>\Tmoins</code> -
<code>\Tpuissance</code> xⁿ	<code>\Tacos</code> ACOS	<code>\Tdel</code> DEL	<code>\Tegal</code> =
<code>\Tlog</code> log	<code>\Tatan</code> ATAN	<code>\Tac</code> AC	<code>\Texe</code> EXE
<code>\Tln</code> ln	<code>\Tasin</code> SIN⁻¹	<code>\Tseconde</code> SECONDE	<code>\Tenter</code> ENTER
<code>\Tneg</code> (-)	<code>\Tacos</code> COS⁻¹	<code>\Tshift</code> SHIFT	<code>\Tans</code> ANS
<code>\Tsin</code> SIN	<code>\Tatan</code> TAN⁻¹	<code>\Tmul</code> x	

GÉNÉRAL

DÉTAILS

6 Objets NathanLive

Il est possible d'introduire des renvois vers des pages Nathan sur Internet avec quatre commandes principales :

`\NathanLiveAudio[texte document]{texte ListOf}{URL}`

`\NathanLiveProg[texte document]{texte ListOf}{URL}`

`\NathanLiveVideo[texte document]{texte ListOf}{URL}`

`\NathanLiveWeb[texte document]{texte ListOf}{URL}`

Chacune de ces commandes signale par un texte qu'une aide de la part de Nathan est disponible. Sa version étoilée place également un pictogramme en rapport en début de texte.

Le premier argument (optionnel) indique le texte qui sera composé à l'endroit où la commande est utilisée.

Le deuxième argument (premier argument obligatoire) est le texte qui sera affiché au niveau de la liste des objets correspondants. On verra ces listes plus loin.

Le dernier argument obligatoire qui peut être vide n'a de sens que si on compose

avec l'option `hyperref`. Dans ce cas, le texte complet devient un lien hypertexte vers cette URL.

Voici le résultat de ces quatre commande (non étoilée) sans indiquer de texte document, c'est-à-dire avec le texte par défaut :

Retrouvez le son dont il est question dans cet exercice

Retrouvez ce programme en ligne

Retrouvez le corrigé de cet exercice en vidéo

Retrouvez le site web

Lorsqu'on utilise ces commandes, chacune crée une entrée dans la liste des objets correspondante. Il y a donc quatre listes différentes qu'on peut afficher avec les quatre commandes :

- `\ListOfAudio`
- `\ListOfProg`
- `\ListOfVideo`
- `\ListOfWeb`

Bien que la maquette actuelle préconise de n'utiliser que la liste des vidéos.

Il est possible d'afficher ces cinq pictogrammes de façon isolée avec les commandes :

- `\PictoAudio` : 
- `\PictoProg` : 
- `\PictoVideo` : 
- `\PictoWeb` : 
- `\PictoNathanLive` : 

En pratique, seule la dernière de ces commandes est sensée être utilisée. Elles admettent toutes un argument optionnel qui doit obligatoirement faire partie de la liste Audio, Prog, Video ou Web. Si cet argument optionnel existe et est correct, il provoquera une entrée dans la liste des objets NathanLive (voir plus bas).

Chacune des commandes `\NathanLiveXxx` provoque également la création d'une entrée dans une liste globale de tous les objets NathanLive. On affiche cette liste avec la commande `\printNL` mais comme cette liste n'a pas vocation à être imprimée sur le document final, on peut également créer un fichier externe avec la commande

`\exportNL[fichier]`

l'argument optionnel indique le nom du fichier où se fera l'exportation. Si cet argument n'est pas précisé, ce sera le nom du fichier maître suivi de l'extension `.nl` qui sera utilisé.