

# **classe-tex3R**

Vincent Crombez & Frédéric Léothaud

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Principe des options</b>	<b>4</b>
1.1 Document minimal . . . . .	4
1.2 Options de mise en page . . . . .	5
1.3 Options pour les environnements d'exercice . . . . .	5
1.4 Options pour les commandes d'exercice . . . . .	5
1.5 Remarques . . . . .	5
<b>2 Type du document</b>	<b>6</b>
<b>3 Environnements de cours</b>	<b>7</b>
<b>4 Commandes et environnements pour les exercices</b>	<b>8</b>
4.1 Présentation générale . . . . .	8
4.2 Exemples d'apparence selon les options . . . . .	9
4.2.1 Premier exemple . . . . .	9
4.2.2 Deuxième exemple . . . . .	9
4.2.3 Troisième exemple . . . . .	10
4.2.4 Quatrième exemple . . . . .	10
4.3 Remarque . . . . .	10
<b>5 Autres environnements et commandes</b>	<b>10</b>
<b>6 Booléens et compteurs</b>	<b>11</b>

# Introduction

La classe-`tex3R` est un travail collaboratif mené par deux professeurs de mathématiques. Le but a été dans un premier temps de rendre compatibles un certain nombre d'outils, notamment le très riche package `\ProfCollege` ou encore les exercices aléatoires générés sur **Coopmaths** tout comme le travail sur les exercices de Brevet réalisé par l'**APMEP**. Les packages `scratch3` et `pstricks` nous donnèrent du fil à retordre, pour finalement arriver au résultat actuel, d'une stabilité à toute épreuve (jusqu'à preuve du contraire).

La classe-`tex3R` se veut comme une classe clé en main, que les puristes jugeront sans doute beaucoup trop lourde au vu du nombre de packages inclus. Elle a cependant le mérite d'être robuste, et de ne nécessiter le chargement d'aucun package supplémentaire pour la plupart des usages courants et d'élaboration de documents à destination des élèves.

La classe-`tex3R` a été conçue afin de pouvoir créer des documents modulaires à partir de différentes options, permettant d'afficher ou masquer certains environnements ou commandes, pour pouvoir générer plusieurs documents à partir d'un même fichier source.

La classe-`tex3R` fait partie d'un projet comprenant :

- Une classe, un style et des templates de documents pratiques
- Un environnement LaTeX portable
- Une extension (basée sur `latex-workshop`)

L'utilisateur n'étant pas friand de l'apparence proposée dans la classe-`tex3R` trouvera son bonheur du côté du `style-tex3R`, où il pourra personnaliser l'apparence de l'ensemble des environnements et outil proposés, tout en gardant les commandes et environnements proposés par la classe-`tex3R`.

# 1 Principe des options

## 1.1 Document minimal

```
\documentclass{classe-tex3R}
\usepackage{style-tex3R}

\begin{luacode}
  Format = 'fiche' --diapo/fiche
  Chapitre = '' --nom du chapitre
  Numero = '' --numéro du chapitre
  Niveau = '' --niveau de classe
\end{luacode}
\parametrage
%FIN PARAMÉTRAGE PRÉAMBULE

\begin{document}

\begin{luacode}
  Type = 'basique' --activite/bilan/corrige/cours/DM/DS/flash/interro/TD
  Impression = false --true
  Header = false --true
  Taille = nil --'14pt'
  Stretch = false --true
  Compteur = '1'

  Correction = false --true
  Enonce = false --true
  Visible = false --true

  Competence = false --true
  Bareme = false --true
  Difficulte = false --true
  Source = false --true
  Theme = false --true
\end{luacode}
\parametrage
%FIN PARAMÉTRAGE DOCUMENT

%Le texte du document

\end{document}
```

Pour des raisons inhérentes à  $\text{\LaTeX}$ , le paramétrage doit être coupé en deux : une partie dans le préambule et une partie dans le corps du document.

Dans le préambule, `Format` permet de basculer d'un document A4 à un document au format 16:10. Par ailleurs, `Chapitre` permet de donner le nom du chapitre, `Numero` le numéro du chapitre et `Niveau` le niveau de classe à qui s'adresse le document.

Ensuite, dans le paramétrage du corps du document, un certain nombre d'options vont permettre de mettre en forme automatiquement le document, en fonction de ce qui est voulu par l'utilisateur.

## 1.2 Options de mise en page

L'option `Type` permet de modifier l'apparence du titre de la page en fonction de l'option choisie.

L'option `Impression` permet de passer les couleurs du document en niveau de gris pour une impression (seulement les couleurs prédéfinies, pas les images insérées).

L'option `Header` permet, si le document est sur plusieurs pages, de répéter ou non le titre en début de chaque nouvelle page.

L'option `Taille` permet de modifier globalement la taille de police du document.

L'option `Stretch` permet de combler les espaces vides entre les différents environnements, afin d'avoir une page plus aérée.

## 1.3 Options pour les environnements d'exercice

L'option `Correction` permet de masquer ou non le contenu des environnements `correction` du document.

L'option `Enonce` permet de masquer ou non le contenu des environnements `enonce` du document.

L'option `Visible` permet de masquer ou non le contenu des environnements `visible` du document.

## 1.4 Options pour les commandes d'exercice

L'option `Competence` permet de masquer ou non le nom des compétences dans les énoncés d'exercice.

L'option `Bareme` permet de masquer ou non le barème dans les énoncés d'exercice.

L'option `Difficulte` permet de masquer ou non les étoiles de difficulté dans les énoncés d'exercice.

L'option `Source` permet de masquer ou non la source d'où provient l'énoncé de l'exercice.

L'option `Theme` permet de masquer ou non le thème dans les énoncés d'exercice.

## 1.5 Remarques

Dans `style-tex3R`, des paramètres prédéfinis pour chaque type de document en fonction de l'usage courant permettent d'éviter l'affichage de l'ensemble des options. Ainsi, il est possible de remplacer l'assignation des options par :

```
\begin{luacode}
  mesParametres('basique') --activite/bilan/corrige/cours/DM/DS/flash/interro/TD
\end{luacode}
\parametrage
```

En dehors d'un paramétrage prédéfini, les options sont toutes initialisées à false, hormis Type (initialisé à basique), Taille (12pt pour la fiche et 20pt pour le diapo) et Compteur initialisé à '1'.

## 2 Type du document

Avant de choisir le Type, dans les exemples suivant on considèrera que les paramètres suivants ont été appliqués dans le préambule du document :

```
Chapitre = 'Théorème de Pythagore'
Numero = '7'
Niveau = '4'
```

L'option Type permet de choisir l'apparence du titre. Voici ci-dessous l'apparence des différents titre prédéfinis pour chaque Type :

- 'basique' :

- 'activite' :

 **Activité** 4<sup>ème</sup>

- 'bilan' :

 **Bilan** 4<sup>ème</sup>

- 'flash' :

 **Questions Flash** 4<sup>ème</sup>

- 'corrige' :

 **Corrigé | Théorème de Pythagore** 4<sup>ème</sup>

- 'cours' :

 **Chapitre VII | Théorème de Pythagore** 4<sup>ème</sup>

- 'TD' :

 **TD | Théorème de Pythagore** 4<sup>ème</sup>

- 'DM' :

 **DM n°1 | NOM :** Prénom : 4<sup>ème</sup>

- 'DS' :

 **DS n°1** | NOM : \_\_\_\_\_ Prénom : \_\_\_\_\_ 4<sup>ème</sup>

- 'interro' :

 **Interrogation n°1** | NOM : \_\_\_\_\_ Prénom : \_\_\_\_\_ 4<sup>ème</sup>

Une fois le Type de document choisi, la commande \titreactif devient disponible afin d'imprimer un titre au besoin à l'emplacement voulu, ce qui est pratique lorsque l'on veut faire tenir deux fiches identiques sur deux demi-pages.

### 3 Environnements de cours

Les environnements application, convention, definition, exemple, methode, propriete, preuve et remarque sont spécifiquement pensés pour les fiches de cours, même s'ils restent disponibles pour les autres Type de document.

```
\begin{application}
  Le contenu.
\end{application}

\begin{convention}
  Le contenu.
\end{convention}

\begin{definition}
  Le contenu.
\end{definition}

\begin{exemple}
  Le contenu.
\end{exemple}

\begin{methode}
  Le contenu.
\end{methode}

\begin{propriete}
  Le contenu.
\end{propriete}

\begin{preuve}
  Le contenu.
\end{preuve}

\begin{remarque}
  Le contenu.
\end{remarque}
```

Application :

Le contenu.

Convention :

Le contenu.

Définition :

Le contenu.

Exemple :

Le contenu.

Méthode :

Le contenu.

Propriété (admise) :

Le contenu.

Preuve :

Le contenu.

Remarque :

Le contenu.

Les deux environnements enonce et correction sont plutôt prévus pour les fiches d'exercice, mais ils changent

d'apparence en Type= ' cours ' pour être comme ci-dessous.

```
\begin{enonce}
  Le contenu.
\end{enonce}

\begin{correction}
  Le contenu.
\end{correction}
```

Énoncé :

Le contenu.

Correction :

Le contenu.

Tous les environnements de cours peuvent prendre une option permettant de changer le texte affiché, ce qui est parfois pratique.

```
\begin{propriete}[Propriété (admise)]
  Le contenu.
\end{propriete}

\begin{remarque}[Remarques]
  Le contenu.
\end{remarque}
```

Propriété (admise) :

Le contenu.

Remarques :

Le contenu.

## 4 Commandes et environnements pour les exercices

### 4.1 Présentation générale

En dehors du Type= ' cours ', les environnements `enonce` et `correction` sont pensés pour être utilisés comme suit :

```
\bareme{} %#1=nombre
\competence{} %#1=nom
\difficulte{} %#1=nombre
\source{} %#1=nom
\theme{} %#1=nom

\begin{enonce}
  %L'énoncé de l'exercice.
\end{enonce}

\begin{correction}
  %La correction de l'exercice.
\end{correction}
```

Un exercice est composé des cinq commandes `\bareme`, `\competence`, `\difficulte`, `\source` et `\theme`, puis des deux environnements `enonce` et `correction`. Grâce au jeu des options, il sera possible de cacher tout ou partie des éléments composant l'exercice.

### 4.2 Exemples d'apparence selon les options



Dans la suite, nous prendrons comme référence cette structure d'exercice et observerons son apparence en fonction des paramètres choisis au début du document.

```
\bareme{6}
\competence{Modéliser, Raisonner}
\difficulte{2}
\source{Groupe de travail TeX3R}
\theme{Pythagore}

\begin{enonce}
  Ceci est l'énoncé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir observer
  et juger son allure globale.
\end{enonce}

\begin{correction}
  Ceci est le corrigé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir
  observer et juger son allure globale.
\end{correction}
```

Comme indiqué précédemment, tous les paramètres étant initialisés à `false`, il n'est pas nécessaire de les indiquer. Dans un souci de clarté de l'exemple, nous les indiquerons ici, systématiquement, même si cela n'est pas nécessaire.

#### 4.2.1 Premier exemple

```
Correction = true
Enonce = true

Bareme = true
Competence = true
Difficulte = true
Source = true
Theme = true
```

*Pythagore*

**1 ★★**

Modéliser, Raisonner

**/6 points**

Ceci est l'énoncé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir observer et juger son allure globale.

*Groupe de travail TeX3R*

#### **1 | Correction**

Ceci est le corrigé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir observer et juger son allure globale.

#### 4.2.2 Deuxième exemple

```
Correction = false
Enonce = true

Bareme = false
Competence = true
Difficulte = false
Source = false
Theme = false
```

**1**

Modéliser, Raisonner

Ceci est l'énoncé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir observer et juger son allure globale.

### 4.2.3 Troisième exemple

```
Correction = true
Enonce = false

Bareme = true
Competence = true
Difficulte = true
Source = true
Theme = true
```

#### 1 | Correction

Ceci est le corrigé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir observer et juger son allure globale.

### 4.2.4 Quatrième exemple

```
Correction = false
Enonce = true

Bareme = false
Competence = false
Difficulte = true
Source = false
Theme = true
```

*Pythagore*

#### 1 ★★

Ceci est l'énoncé de l'exercice, qui s'étend sur plusieurs lignes afin de pouvoir observer et juger son allure globale.

## 4.3 Remarque

Il n'est pas nécessaire de compléter les commandes `\bareme`, `\competence`, `\difficulte`, `\source` et `\theme`. Il est cependant recommandé de les laisser, même vides car elles redéfinissent des commandes internes pour chaque exercice. En leur absence, le contenu appliqué serait celui de l'exercice précédent, ce qui n'est pas un comportement souhaité.

## 5 Autres environnements et commandes

```
\montitre[]{}
%#1=logo (optionnel), #2=contenu
```

La commande `\montitre` permet de mettre un autre titre que ceux prédéfinis, au besoin. Le paramètre `Header` est compatible avec ce titre.

```
Ceci est un texte \important{très important}.
```

Ceci est un texte très important.

```
{\tailletexte{0} Texte normal}
{\tailletexte{2} Texte plus grand}
{\tailletexte{-3} Texte plutôt petit}
%#1 va de -4 à 4
```

Texte normal  
Texte plus grand  
Texte plutôt petit

```
\partie{Une partie}
\souspartie{Une sous-partie}
```

La commande `\chapitre` est utilisée pour les méga-documents, voir en template.

```
\begin{visible}[false]
  %Contenu visible si Visible = false
\end{visible}

\begin{visible}[true]
  %Contenu visible si Visible = true
\end{visible}

\visiblecmd[true]{Contenu visible si Visible
  = true}
\visiblecmd[false]{Contenu visible si Visible
  = false}
```

```
\saut{ligne}
\saut{fiche}
\saut{diapo}
```

```
\lignes{seyes}{1}
\lignes{carreau}{2}
\lignes{surligne}{3}
Ceci est un texte qui s'étend sur plusieurs
lignes et qui sera plus facilement
lisible si on le surligne une ligne sur
deux avec des couleurs différentes.
```

## 6 Booléens et compteurs

```
\newcounter{compteurexercice}
\newcounter{compteurinterro}
\newcounter{compteurflash}
\newcounter{compteurDM}
\newcounter{compteurDS}
\newcounter{compteurTD}
```

## A Une partie

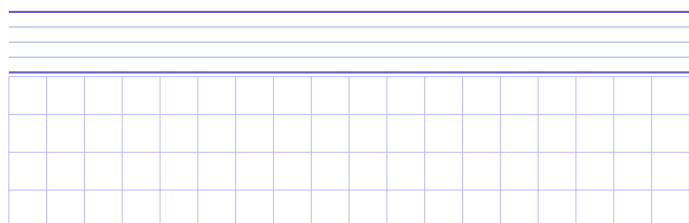
### 1 Une sous-partie

L'environnement `visible` et son équivalent en commande `\visiblecmd` permettent de commuter entre deux contenus en fonction du booléen choisi pour le paramètre `Visible`, ce qui est parfois pratique.

`\saut{ligne}` effectue un saut de ligne, relativement à la hauteur actuelle du texte.

`\saut{fiche}` effectue un saut de page uniquement en format fiche.

`\saut{diapo}` effectue un saut de page uniquement en format diapo.



Ceci est un texte qui s'étend sur plusieurs lignes et qui sera plus facilement lisible si on le surligne une ligne sur deux avec des couleurs différentes.

Ces compteurs sont utilisés dans les environnements de la classe `tex3R`. Voici un rappel des commandes de compteurs :

```
\setcounter{compteurexercice}{3}
\stepcounter{compteurexercice}
\thecompteurexercice
```

```
\ifdiapo
\iffiche
\ifheader
\ifprint

\ifactivite
\ifbasique
\ifbilan
\ifcorrige
\ifcours
\ifTD
\ifflash
\ifDM
\ifDS
\ifinterro

\ifcorrection
\ifenonce
\ifvisible

\ifbareme
\ifdifficulte
\ifcompetence
\ifsource
\iftheme

\ifstretch

\ifsubfile
```

Les booléens suivants sont initialisés lors du paramétrage initial à true ou false.

Le booléen `\ifsubfile` est utilisé pour l'appel d'exercices en subfile (voir template).

Voici un exemple d'utilisation d'un booléen, pour afficher du contenu en fonction du paramétrage initial :

```
\ifdiapo Choix1 \else Choix2 \fi
```