classe-rivieres

style-rivieres

Vincent Crombez & Frédéric Léothaud

Table des matières

Introduction

La `classe-rivieres` 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**.

La `classe-rivieres` 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-rivieres` 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 finalement arriver à générer plusieurs documents lors d'une seule compilation.

Enfin, ce document, qui se veut à la fois comme un guide d'utilisation et d'installation de LaTeX, a été élaboré afin de permettre aux personnes n'étant pas familières de LaTeX de pouvoir se lancer et de profiter des différents outils qu'elles pourront trouver ici.

1 L'exemple du professeur Euclide

1.1 Introduction

Le professeur Euclide souhaite créer une fiche de TD pour ses élèves de 4ème, sur les équations. En ce samedi pluvieux, il se dit qu'il est temps de se lancer sur LaTeX, et décide de suivre la documentation de `classe-rivieres`. Étant d'un naturel aguerri en ce qui concerne l'informatique, il installe sans encombre son éditeur et sa distribution, en suivant le guide d'installation. Le voici dans VS Code, il créé son nouveau fichier qu'il intitule `Equations.tex` (il a bien noté l'importance de préciser l'extension du fichier dans VS Code), et se retrouve dans sa nouvelle page, qu'il construit comme indiqué dans la documentation :

```
\documentclass{classe-rivieres}

\begin{document}

\end{document}
```

Il lance alors la compilation de son document, afin de vérifier que cette structure de base est correcte, mais son éditeur lui renvoie une erreur. Après de multiples recherches sur la documentation et sur Internet, il se rend compte que son document étant actuellement vide, il est normal qu'une erreur apparaisse. Il recommence alors avec cette structure :

```
\documentclass{classe-rivieres}

\begin{document}
Test
\end{document}
```

Et la magie opérant, le fichier se compile à merveille. Il est alors temps pour le professeur Euclide de se retrousser les manches et de se mettre sérieusement au travail.

1.2 Paramètres de la fiche

N'ayant pas encore très bien compris ces histoires de graines, il se lance dans la construction de son document en utilisant la commande `\parametres`.

Il a déjà sélectionné un certain nombre d'exercices qui tiendront vraisemblablement sur plusieurs pages, et il souhaiterait avoir une en-tête de fiche de TD sur chacune de ses feuilles. Il complète donc son fichier de la manière suivante :

```
\documentclass{classe-rivieres}

\begin{document}

\parametres{type=TD,enonce}

\begin{enonce}
  Résoudre les équations suivantes :

    \begin{enumerate}
    \item $4x+5=8x-3$
        \item $8x-3=2x+5$
  \end{enumerate}
\end{enonce}

\end{document}
```

Après compilation, il est satisfait du résultat, et décide d'ajouter les compétences travaillées et le niveau de difficulté de l'exercice.

```
\documentclass{classe-rivieres}

\begin{document}
```

```
\parametres{type=TD,enonce,difficulte,competence}

\difficulte{1}
\competence{Calculer}

\begin{enonce}
    Résoudre les équations suivantes :

    \begin{enumerate}
        \item $4x+5=8x-3$
        \item $8x-3=2x+5$
    \end{enumerate}
\end{enonce}

\end{document}
```

Après lecture de la documentation de l'excellent package `ProfCollege`, il remarque qu'il peut générer de manière quasi-automatique la correction de son exercice, ce qu'il fait :

```
\documentclass{classe-rivieres}

\begin{document}

\parametres{type=TD,enonce,difficulte,competence}

\difficulte{1}
\competence{Calculer}

\begin{enonce}
    Résoudre les équations suivantes :

    \begin{enumerate}
        \item $4x+5=8x-3$
        \item $-5x-3=2x+5$
    \end{enumerate}
\end{enonce}

\begin{correction}
  \begin{enumerate}
    \item \ResolEquation{4}{5}{8}{-3}
    \item \ResolEquation{-5}{-3}{2}{5}
  \end{enumerate}
\end{correction}

\end{document}
```

Pour l'instant la correction n'est pas affichée sur la fiche, mais il se dit que si plus tard il en a besoin, elle sera disponible.

Étant décidé à peaufiner son exercice, avant de passer au suivant, il décide de rajouter la source et le thème, même si par goût personnel il préfère ne pas les afficher.

```
\documentclass{classe-rivieres}

\begin{document}

\parametres{type=TD,enonce,difficulte,competence}

\difficulte{1}
\competence{Calculer}
\source{Professeur Euclide}
\theme{Équations}

\begin{enonce}
    Résoudre les équations suivantes :

    \begin{enumerate}
        \item $4x+5=8x-3$
        \item $-5x-3=2x+5$
    \end{enumerate}
\end{enonce}

\begin{correction}
    \begin{enumerate}
        \item \ResolEquation{4}{5}{8}{-3}
        \item \ResolEquation{-5}{-3}{2}{5}
    \end{enumerate}
\end{correction}

\end{document}
```

Enfin, après réflexion, il décide de rajouter le chapitre et le niveau concerné, afin que ceux-ci apparaissent dans le titre de sa fiche :

```
\documentclass{classe-rivieres}

\begin{document}

\chapitre{5}{Équations}
\niveau{4}

\parametres{type=TD,enonce,difficulte,competence}

\difficulte{1}
\competence{Calculer}
\source{Professeur Euclide}
\theme{Équations}

\begin{enonce}
    Résoudre les équations suivantes :
```

```
    \begin{enumerate}
        \item $4x+5=8x-3$
        \item $-5x-3=2x+5$
    \end{enumerate}
\end{enonce}

\begin{correction}
    \begin{enumerate}
        \item \ResolEquation{4}{5}{8}{-3}
        \item \ResolEquation{-5}{-3}{2}{5}
    \end{enumerate}
\end{correction}

\end{document}
```

Le professeur Euclide, fort satisfait de son travail écrit plusieurs autres exercices dans son document, et obtient exactement ce qu'il voulait, à savoir plusieurs fiches de TD, à entête numérotée.

1.3 Utilisation de `multiTD`

Le professeur Euclide ayant désormais terminé sa fiche de TD, il s'intéresse à l'option `multiTD` qui lui semble alléchante. Après lecture, il décide de s'en tenir aux réglages de base, et modifie simplement son préambule de document comme suit :

```
\documentclass[multiTD]{classe-rivieres}
```

La compilation se termine sans problèmes, même si le professeur Euclide la trouve un peu longue. C'est alors qu'il comprend que son éditeur n'a pas généré un mais bien cinq documents `.pdf` ! Les cinq documents ont un nom différent de son fichier de départ, pour pouvoir être identifiés facilement. Il y retrouve sa fiche de TD classique, son corrigé, une fiche avec les énoncés et les corrigés, un diaporama avec l'ensemble des exercices et un deuxième diaporama comprenant les exercices et leur correction. Le professeur Euclide réfléchit au temps qu'il aurait mis à créer tous ces documents par lui-même, et finit par pardonner à son éditeur la lenteur de la compilation.

1.4 Utilisation de `subfiles`

Alors que la fiche de TD du professeur Euclide est terminée, ce dernier réalise en relisant la documentation qu'il pourrait stocker chacun de ses exercices dans un fichier à part, et les inclure ensuite dans la fiche de TD. Il commence à en percevoir l'intérêt pour deux raisons. La première est que son document de TD est tout de même peu

lisible, car ses exercices sont nombreux. La seconde est que si ses exercices étaient stockés de manière externe, ils seraient beaucoup plus faciles à manipuler, à corriger et à réutiliser dans d'autres types de documents, dans un DS par exemple. Le professeur Euclide se lance donc bravement dans le découpage de sa fiche de TD. Il utilise l'arborescence recommandée dans la documentation, récupère le fichier `Sommaire.tex` et créé un certain nombre de fichiers `.tex` dont voici l'arborescence :

```
Chapitre 5 - Équations
├── TD
│   └── Equations.tex
├── Exercices
│   ├── Images
│   ├── equations-basiques.tex
│   ├── equations-niveau2.tex
│   ├── probleme-aire.tex
│   └── probleme-vitesse.tex
├── Sommaire.tex
└── main-exercices.tex
```

Le professeur Euclide n'a pas voulu prendre le risque de mettre des espaces dans les noms de ses fichiers, mais cette précaution n'est (normalement) pas nécessaire.

Le professeur Euclide ayant habilement découpé ses exercices en subfiles, voici à quoi ressemblent désormais ses différents fichiers :

```
%Le fichier Equations.tex (la fiche de TD)
\documentclass{classe -rivieres}

\graphicspath{{../Exercices/Images/}}

\begin{document}

\chapitre{5}{Équations}
\niveau{4}

\parametres{type=TD,enonce ,difficulte ,competence}

\subfile{../Exercices/equations-basiques.tex}

\subfile{../Exercices/equations-niveau2.tex}

\subfile{../Exercices/probleme-aire.tex}

\subfile{../Exercices/probleme-vitesse.tex}

\end{document}
```

```
%Le fichier equations-basiques.tex (un fichier d'exercices)
\documentclass[../main-exercices.tex]{subfiles}
```

```
\difficulte{1}
\competence{Calculer}
\source{Professeur Euclide}
\theme{Équations}

\begin{enonce}
    Résoudre les équations suivantes :

    \begin{enumerate}
        \item $4x+5=8x-3$
        \item $-5x-3=2x+5$
    \end{enumerate}
\end{enonce}

\begin{correction}
    \begin{enumerate}
        \item \ResolEquation{4}{5}{8}{-3}
        \item \ResolEquation{-5}{-3}{2}{5}
    \end{enumerate}
\end{correction}
```

```
%Le fichier main-exercices.tex
%(le fichier 'principal' pour les exercices)
\documentclass[graine=3233230]{classe-rivieres}
\graphicspath{{./Images/}}
\begin{document}

\end{document}
```

Au départ, le professeur Euclide ne voyait pas trop comment rentre l'adresse de ses exercices individuels dans sa fiche de TD, mais il s'est alors rendu compte que le fichier `Sommaire.tex` lui permettait de les obtenir directement, en plus de la visualisation de leur rendu. Comme certains de ses exercices nécessitent des images, il a créé un dossier `Images` dans son dossier `Exercices`, et il a précisé dans deux de ses fichiers le chemin d'accès pour pouvoir s'y rendre.

Le professeur Euclide est désormais comblé, il vérifie tout de même que l'option `multiTD` fonctionne toujours, ce qui est bien le cas. Il décide de s'octroyer une petite pause car c'est l'heure du goûter. Il se lancera plus tard dans la rédaction d'une fiche de cours, car il a vu des fonctionnalités intéressantes, notamment l'environnement `lignes`.

2 Installation de LaTeX et de la classe

Pour les utilisateurs aguerris, disposant déjà d'une installation LaTeX fonctionnelle, la seule chose à noter est que la compilation des documents générés via la `classe-`

`rivieres` doit être en `shell-escape`, et utiliser `lualatexmk` (ou deux compilations en `lualatex`).

Pour la suite de cette section nous allons détailler une installation possible, en utilisant MikTeX et VS Code, même si d'autres possibilités existent (TeXLive pour la distribution et Vim pour l'éditeur par exemple).

Pour les personnes ne souhaitant pas s'embêter avec l'ensemble des étapes d'installation ou souhaitant un environnement nomade, la version portable de la classe, ainsi que tout le code source est disponible sur **Github**.

2.1 Installation de MikTeX

LaTeX est un langage permettant de faire appel à des commandes. Les commandes sont créées et appelées via des packages, qui doivent être installés sur l'ordinateur lorsque l'on fait appel à eux. MikTeX est une distribution permettant de faire ceci.

Pour installer MikTeX, se rendre sur : https://miktex.org/download et télécharger l'installateur correspondant à la version du système d'exploitation. Ensuite, lancer l'installateur et suivre les étapes d'installation. Il est recommandé de sélectionner l'installation des packages 'on the fly' afin que ces derniers s'installent automatiquement lorsque c'est nécessaire.

2.2 Installation de Visual Studio Code

Pour écrire nos fichiers LaTeX (en `.tex`) , il faut un éditeur de texte, par exemple Visual Studio Code (VS Code). Pour installer VS Code, se rendre sur : https://code.visualstudio.com/ et télécharger l'installateur correspondant à la version du système d'exploitation. Ensuite, lancer l'installateur et suivre les étapes d'installation.

2.3 Configuration de VS Code

2.3.1 Extensions

VS Code est un logiciel qui nécessite quelques extensions afin de fonctionner correctement pour notre usage.

Sur le volet gauche, sélectionner Extensions, et rechercher et installer les deux plugins suivants :

- French Language Pack for Visual Studio Code
- Tex 3 Rivières

2.3.2 Paramètres

Dans VS Code :

1) Aller dans Fichier>Préférences>Paramètres
2) Dans la barre de recherche, écrire : `latex-workshop.tools`
3) Ouvrir le `.json`
4) Remplacer le contenu du `.json` ainsi ouvert par le contenu du fichier `settings.json` disponible sur Github

2.4 Installation de la classe

Créer un dossier présentant cette arborescence, dans un dossier de l'ordinateur possible à retrouver, et y copier les fichiers `classe-rivieres.cls` et `style-rivieres.sty` aux endroits indiqués.

```
texmf
└── tex
    ├── bin
    └── lualatex
        ├── classe-rivieres
        │   └── classe-rivieres.cls
        └── style-rivieres
            └── style-rivieres.sty
```

Ensuite, ouvrir MikTeX Console et aller dans Settings>Directories>+(Add) et localiser le dossier texmf. Enfin, cliquer sur Tasks>Refresh file name database

1) Localiser le dossier `texmf`
2) Refresh

2.5 Installation des polices

Par goût personnel, il est possible de télécharger des polices différentes, notamment pour les formules mathématiques et pour les lettres calligraphiées.

Les liens vers les polices proposées se retrouvent dans le dossier `Polices` sur **Github**.

2.6 Installation de Sumatra (optionnel)

S'il est possible d'utiliser le visualiseur `.pdf` intégré de l'extension LaTeX Workshop, il est parfois pratique de regarder ses documents dans son lecteur de `.pdf` habituel. Adobe Reader a le problème principal d'ouvrir les fichiers `.pdf` en lecture seule, ce qui interdit la compilation des fichiers dans l'éditeur LaTeX, tant que le fichier `.pdf` n'a pas été fermé. Sumatra a contrario, permet l'écriture sur un fichier déjà ouvert dans son lecteur, et l'affichage des modifications à chaque nouvelle compilation.

Pour installer Sumatra, se rendre sur :

https://www.sumatrapdfreader.org/download-free-pdf-viewer

et télécharger l'installateur correspondant à la version du système d'exploitation. Ensuite, lancer l'installateur et suivre les étapes d'installation.

Il est possible dans Sumatra de configurer la recherche inversée, c'est à dire qu'en double-cliquant sur du texte du `.pdf`, cela ouvre l'éditeur LaTeX à ligne correspondante, dans le fichier `.tex` concerné. Pour VS Code, le fichier .json précédemment copié prend déjà en charge la fonctionnalité, qui se nomme synctex. Dans Sumatra, il faut configurer le lecteur de la manière suivante :

Préférences>Options>Configurer la recherche avancée :

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\Code.exe" -g
"%f":"%l"
```

Le chemin ici proposé est celui par défaut de l'emplacement de VS Code sous Windows, en remplaçant simplement User par le nom du dossier utilisateur.

3 Utilisation basique de la classe

Dans cette section, nous allons présenter l'ensemble des commandes et environnements à utiliser lors de la création d'un nouveau document. Chaque commande sera proposée avec un rendu, afin de mieux le visualiser. Le rendu de chaque environnement ou commande est entièrement personnalisable dans `style-rivieres`.

3.1 Choix de l'affichage des options

L'idée principale de la classe étant de pouvoir générer plusieurs types de document à l'aide du même fichier `.tex`, les contenus du fichier sont écrits dans différents environnements qui sont affichés ou non en fonction des options.

3.1.1 Avec une graine, dans les options de classe

Lors de l'utilisation de la classe, il est tout à fait possible de spécifier quel format est souhaité, pour l'ensemble du document. Les options retenues par ce format sont définies par une « graine » dont le fonctionnement sera détaillé ci-après :

Valeur d'option	Description
2	format du document en A4 pour les fiches imprimables
3	format du document en 16:10 pour les diaporamas projetables
5	l'environnement `enonce` est affiché
7	l'environnement `correction` est affiché
11	le contenu de la commande `\theme` est affichée
13	le contenu de la commande `\difficulte` est affichée
17	le contenu de la commande `\competence` est affichée
19	le contenu de la commande `\source` est affichée
107	le contenu de la commande `\bareme` est affichée
23	l'environnement `\lignes` est activé
73	l'environnement `\lignes*` est activé
83	l'environnement `\lignes` affiche des petits carreaux
101	le style de `\lignes` surligne une ligne sur deux
89	l'environnement `\lignes*` affiche des petits carreaux
103	le style de `\lignes*` surligne une ligne sur deux
31	le titre en début de page est celui de `\titreactivite` (Activités)
41	le titre en début de page est celui de `\titrecorrige` (Corrigés)
43	le titre en début de page est celui de `\titrecours` (Cours)
59	le titre en début de page est celui de `\titreTD` (Exercices)
61	le titre en début de page est celui de `\titreDM` (Devoirs Maison)
67	le titre en début de page est celui de `\titreDS` (Devoirs Surveillés)
71	le titre en début de page est celui de `\titreinterro` (Interrogations)
97	le titre en début de page est celui de `\titreflash` (Questions Flash)
109	le style du document est celui d'une épreuve de DNB
37	pas de titre en début de page
29	le titre prédéfini s'imprime au début de chaque nouvelle page
79	affiche la table des matières du document

Parmi ces valeurs d'option, la 2 et la 3 sont incompatibles entre elles (il faut se décider sur un format de document), tout comme les valeurs d'option 31, 41, 43, 59, 61, 67, 71, 97 et 37 qui définissent le style du titre.

Une fois les valeurs d'options choisies, il suffit de les multiplier entre elles pour obtenir la graine du document, qui pourra être mise en option de classe.

Par exemple, pour un document de format fiche (2), pour lequel on souhaite avoir le

titre d'une feuille de corrigés (41), qui se mette automatiquement sur chaque nouvelle page (29), avec les énoncés visibles (5) tout comme les corrections (7), on obtient la graine suivante :

$$2 \times 41 \times 29 \times 5 \times 7 = 83230$$

Il ne reste alors plus qu'à construire le document de la manière suivante :

```
\documentclass[graine=83230]{classe-rivieres}

\begin{document}
...
\end{document}
```

En l'absence de l'option `graine`, ou en précisant simplement l'option `fiche` la graine du document par défaut est 2 (fiche A4 classique). En précisant l'option `diapo`, la graine du document est initialisée à 3.

Cette notion de graine sera utilisée pour la génération de documents multiples, dans la partie dédiée.

3.1.2 Avec les commandes `\parametres` et `\parametres*`

Les commandes `\parametres` et `\parametres*` permettent de gérer l'affichage des différents environnements et commandes de la classe.

La commande `\parametres{<clé1=valeur1,clé2=valeur,…>}` permet dans un document de sauter une page, et de continuer avec d'autres options pour la suite du document. Toutes les options doivent être spécifiées, car toute la configuration précédente est écrasée lors de l'emploi de la commande `\parametres`.

Le tableau suivant précise les clés et valeurs possibles, pour la commande `\parametres` :

Clé	Valeurs possibles	Description
enonce	true/false	affichage de l'environnement `enonce`
correction	true/false	affichage de l'environnement `correction`
theme	true/false	affichage du contenu de `\theme`
difficulte	true/false	affichage du contenu de `\difficulte`
competence	true/false	affichage du contenu de `\competence`
source	true/false	affichage du contenu de `\source`
bareme	true/false	affichage du contenu de `\bareme`
lignes	true/false/seyes/carreaux	activation de l'environnement `lignes`
lignes*	true/false/seyes/carreaux	activation de l'environnement `lignes*`
type	activite/basique/corrige/cours/TD/DM/DS/interro/brevet	type du titre
titre	true/false	affichage du titre sur chaque page

Pour toutes les clés acceptant `true` comme valeur, il n'est pas nécessaire de préciser `=true` dans la définition de la clé, afin de faciliter la lecture et la syntaxe. Par ailleurs, si rien n'est précisé pour les clés `lignes` ou `lignes*`, le style par défaut des lignes sera alors `seyes`.

La commande `\parametres` permet de remplacer le principe de graine vue précédemment, à la préférence de l'utilisateur. Ainsi, les deux syntaxes de début de document sont strictement équivalentes.

```
\documentclass[graine=83230]{classe-rivieres}

\begin{document}
...
\end{document}
```

```
\documentclass[fiche]{classe-rivieres}

\begin{document}
\parametres{type=corrige,titre,enonce,correction}
...
\end{document}
```

Il est important de noter que la commande `\parametres` doit nécessairement se trouver dans l'environnement `document` et non avant, pour pouvoir fonctionner correctement.

La commande `\parametres*` fonctionne sous la même forme que la commande `\parametres`,

à la différence près qu'elle n'écrase pas les anciennes options, elle ne redéfinit que celles pour lesquelles on lui précise des modifications. Par ailleurs les clés `type` et `titre` ne sont pas utilisables dans la commande.

3.2 Commandes et environnements

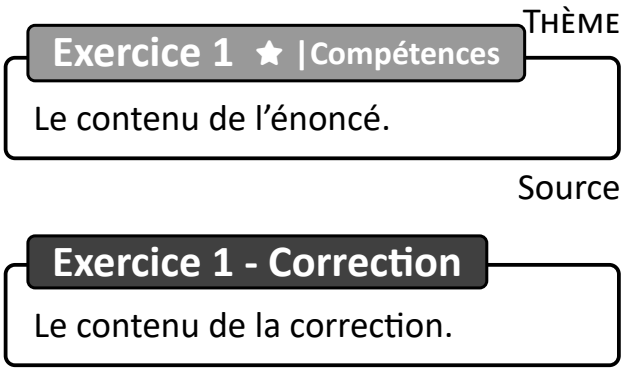
3.2.1 Pour les exercices

La structure minimale d'un exercice devrait toujours être comme suit :

```
\source{Source}
\theme{Thème}
\competence{Compétences}
\difficulte{1}

\begin{enonce}
  Le contenu de l'énoncé.
\end{enonce}

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



Même si les commandes et environnements sont relativement explicites, voici le détail de leur utilisation :

`\source`	Permet de spécifier la source d'où est tiré l'exercice
`\theme`	Permet de spécifier les thèmes de l'exercice
`\competence`	Permet de spécifier les compétences principales travaillées dans l'exercice
`\difficulte`	Permet de spécifier le niveau de difficulté de l'exercice (de 1 à votre convenance)
`enonce`	Environnement où écrire le texte pour l'énoncé de l'exercice
`correction`	Environnement où écrire le texte pour la correction de l'exercice

3.2.2 Pour le cours

Le texte du cours peut être dans les environnements suivants : `definition`, `propriete`, `methode`, `application`, `exercice`, `exemple`, `remarque`, `basique`, ce qui donne le rendu suivant :

```
\begin{definition}
  La définition.
\end{definition}

\begin{propriete}
  La propriété.
\end{propriete}

\begin{methode}
  La méthode.
\end{methode}

\begin{application}
  L'application.
\end{application}

\begin{exercice}
  L'exercice.
\end{exercice}

\begin{exemple}
  L'exemple.
\end{exemple}

\begin{remarque}
  La remarque.
\end{remarque}

\begin{basique}
  Le texte basique.
\end{basique}
```

Définition :
La définition.

Propriété :
La propriété.

Méthode :
La méthode.

Application :
L'application.

Exercice :
L'exercice.

Exemple :
L'exemple.

Remarque :
La remarque.

Preuve :
La preuve. ■

Le texte basique.

L'environnement basique n'est pas nécessaire, il est actuellement défini comme vide, mais si jamais l'on souhaite récupérer le texte écrit dans un environnement basique pour lui apporter une mise en forme ultérieurement, il peut être judicieux d'avoir utilisé cet environnement au préalable.

Si l'on souhaite modifier localement un environnement, il est tout à fait possible de le faire, avec l'option de l'environnement :

```
\begin{propriete}[Propriété (admise) :]
  La propriété.
\end{propriete}
```

Propriété (admise) :
La propriété.

3.2.3 Les environnements `lignes` et `lignes*`

Ces deux environnements permettent d'entourer du texte, pour qu'il puisse être remplacé à la demande par des lignes de format seyes, des lignes à petits carreaux, ou de surligner une ligne sur deux pour faciliter la lecture (en fonction des options présélectionnées). Les deux environnements sont strictement équivalents, mais ils apportent de la souplesse dans le document, en permettant de choisir des paramètres différents d'affichage pour chacun d'entre eux.

```
\parametres*{lignes,lignes*=false}

Du texte au dessus.

\begin{lignes}
Du texte.

Encore du texte.
\end{lignes}%
\begin{lignes*}
Toujours plus de texte.
\end{lignes*}%

Du texte après.
```

Du texte au dessus.
1 Du texte.
2 Encore du texte.
Toujours plus de texte.
Du texte après.

```
\parametres*{lignes=carreaux,
  lignes*=seyes}

Du texte au dessus.

\begin{lignes}
Du texte.

Encore du texte.
\end{lignes}%
\begin{lignes*}
Toujours plus de texte.
\end{lignes*}%
Du texte après.
```

Du texte au dessus.
3 Du texte.
4 Encore du texte.
5 Toujours plus de texte.
Du texte après.

```
\parametres*{lignes=surligne}

\begin{lignes}
  Lorem ipsum dolor [...] pariatur.
\end{lignes}%
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

3.2.4 Commandes générales

`\tracelignes{<nombre>}`	Permet de tracer <nombre> lignes seyes, occupant toute la largeur de ligne actuelle.
`\tracecarreaux{<nombre>}`	Permet de tracer <nombre> lignes à petits carreaux (une ligne faisant deux petits carreaux de hauteur), occupant toute la largeur de ligne actuelle.
`\ajoutlignes{<nombre>}`	Permet d'ajouter manuellement un nombre de lignes au prochain environnement `lignes` ou `lignes*` rencontré.
`\tailletexte{<nombre>}`	Permet de remplacer les commandes `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE` et `\huge`, en choisissant un nombre de −4 à 4, 0 étant la taille normale du texte.
`\chapitre{<nombre>}{<nom>}`	Permet de changer manuellement la valeur de `\thepart` et le nom de `\choixchapitre` pour personnaliser les titres.
`\niveau{<nombre>}`	Permet de changer manuellement la valeur (6, 5, 4 ou 3) de `\choixniveau` pour personnaliser les titres.

`\titreactivite`
`\titrecorrige`
`\titrecours`
`\titreTD`
`\titreflash`
`\titreDM`
`\titreDS`
`\titreinterro`
	Impriment manuellement un titre à l'endroit de la commande

4 Compilation de multiples documents

Une fois le document bien balisé grâce aux différents environnements, il est désormais possible d'en générer plusieurs versions, en une seule compilation. Il est nécessaire d'avoir correctement configuré son éditeur afin d'autoriser l'exécution de script externes en `--shell-escape`.

4.1 En utilisant les options par défaut de la classe

Quatre options de classe sont définies par défaut : `multiTD`, `multicours`, `multiflash` et `multiDS`.

4.1.1 L'option `multiTD`

Cette option permet de générer cinq documents `.pdf` à partir du fichier `.tex` compilé. Les graines utilisées, le principe du fichier ainsi que le nom du `.pdf` de sortie sont récapitulés dans le tableau suivant :

Graine	Fichier correspondant	Nom du fichier
590	Fiche de TD à distribuer aux élèves	TD-fiche-enonce
826	Fiche de correction	TD-fiche-correction
4130	Fiche avec les énoncés suivis des corrections	TD-fiche-enonce-correction
25665	TD complet à projeter	TD-diapo-enonce
4132065	TD à projeter, avec la correction, et les lignes activées	TD-diapo-enonce-correction-lignes

4.1.2 L'option `multicours`

Cette option permet de générer quatre documents `.pdf` à partir du fichier `.tex` compilé. Les graines utilisées, le principe du fichier ainsi que le nom du `.pdf` de sortie sont récapitulés dans le tableau suivant :

Graine	Fichier correspondant	Nom du fichier
86	Fiche de cours complète	cours-fiche
144394	Trame de cours avec les lignes	cours-fiche-trame
3741	Cours complet à projeter	cours-diapo
86043	Cours à projeter et à compléter avec les lignes	cours-diapo-trame

4.1.3 L'option `multiflash`

Cette option permet de générer cinq documents `.pdf` à partir du fichier `.tex` compilé. Les graines utilisées, le principe du fichier ainsi que le nom du `.pdf` de sortie sont récapitulés dans le tableau suivant :

Graine	Fichier correspondant	Nom du fichier
28130	Flash à distribuer	flash-fiche-enonce
4528930	Corrigé du flash	flash-fiche-correction
196910	Fiche flash avec énoncé et correction	flash-fiche-enonce-correction
42195	Diapo flash à projeter	flash-diapo-enonce
295365	Diapo flash avec l'énoncé suivi de la correction	flash-diapo-enonce-correction

4.2 En personnalisant les options de la classe

Avant de rentrer dans le détail de cette section il est primordial d'avoir bien saisi le concept de graine d'un document. Il faut alors dans un premier temps réfléchir aux valeurs des graines de chacun des documents que l'on souhaite générer à partir du même document de départ.

Une fois les graines choisies, la personnalisation des options nécessite la modification du `classe-rivieres`, il est donc recommandé d'en faire une copie au préalable.

Voici les différentes étapes à suivre :

1) Création d'un nouveau booléen : `\newif\ifmonoption`
2) Initialisation du booléen : `\monoptionfalse`
3) Création de l'option : `\DeclareOption{monoption}\monoptiontrue\multitrue`

Ces différentes commandes étant à écrire dans le fichier, il est conseillé de mettre chacune des commandes à la suite de celles du même type, dans un souci de clarté.

Il s'agit ensuite d'écrire la condition suivante, juste avant la commande :

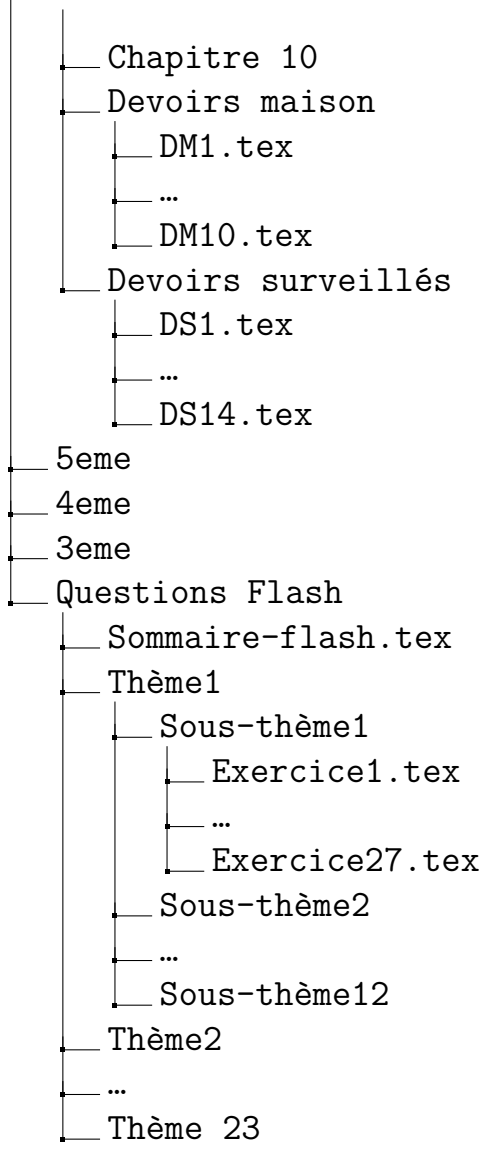
\def\conditionmacro{2} :

```
\ifcoursmultiple%pour l'option multicours
  \runlualatex{<nom-fichier-1>}{<graine1>}
  \runlualatex{<nom-fichier-2}{<graine2>}
  ...
  \runlualatex{<nom-fichier-5>}{<graine5>}
  \expandafter\stop
\fi
```

5 Création de mega-documents

Afin de créer des mega-documents, il est recommandé d'utiliser une arborescence de dossiers comme suit :

```
Mes cours
└── 6eme
    ├── 6eme.tex
    ├── Activités transversales
    │   ├── Activité1.tex
    │   ├── …
    │   └── Activité17.tex
    ├── Chapitre 1
    │   ├── Activités
    │   │   ├── Activité1.tex
    │   │   ├── …
    │   │   └── Activité7.tex
    │   ├── Cours
    │   │   └── Chapitre1-Cours.tex
    │   ├── Exercices
    │   │   ├── Exercice1.tex
    │   │   ├── …
    │   │   └── Exercice18.tex
    │   ├── Interrogations
    │   │   ├── Interrogation1.tex
    │   │   ├── …
    │   │   └── Interrogation5.tex
    │   └── TD
    │       └── Chapitre1-TD.tex
    ├── Chapitre 2
    └── …
```



Un exemple est disponible sur Github dans le dossier dédié à cet effet.

6 Personnalisation de `style-rivieres.sty`

Le fichier `style-rivieres` peut être modifié pour personnaliser complètement l'ensemble des environnements.

6.1 Polices du document

La compilation en LuaLaTeX permet d'utiliser directement toutes les polices installées sur l'ordinateur. Trois personnalisations de police sont proposées, une pour le texte standard, une pour le texte en math-mode, et une autre pour les caractères de calligraphie obtenus en math-mode à l'aide de la commande `\mathcal`. Pour ces deux

dernières, il est préférable de rechercher des polices prévues pour le math-mode de LaTeX, afin que tous les symboles mathématiques soient définis dans la police choisie.

6.2 Marges du document

Les marges sont changées dans la classe, mais la définition des longueurs associées se fait via le package de style. Il est ainsi possible de procéder à tous les ajustements nécessaires depuis le fichier de style.

6.3 Taille de police du document

De manière classique, la taille de police du document est définie globalement. Les différents éléments pourront voir leur taille ajustée soit via les commandes natives de LaTeX de mise en page du texte (`\tiny`, `\scriptsize`, …) ou via la commande personnelle `\tailletexte`.

6.4 Environnements

Les environnements pouvant être modifiés sont : `enonce`, `correction`, `definition`, `propriete`, `methode`, `application`, `exercice`, `exemple`, `remarque` et `basique`.

Pour ce faire, un certain nombre de commandes peuvent-être utilisées :

6.4.1 Les commandes personnelles

`\rivniveau`	Affiche le niveau sélectionné par la commande `\niveau`
`\rivchapitre`	Affiche le chapitre sélectionné par le deuxième argument de la commande `\chapitre`
`\rivsource`	Affiche la source sélectionnée par la commande `\source`
`\rivtheme`	Affiche le thème sélectionné par la commande `\theme`
`\rivdifficulte`	Affiche la difficulté sélectionnée par la commande `\difficulte`
`\rivcompetence`	Affiche les compétences sélectionnées par la commande `\competence`
`\rivbareme`	Affiche le barème sélectionné par la commande `\bareme`
`\titre{<logo>}{<contenu>}`	Affiche un titre ayant une apparence similaire aux titres prédéfinis.

\logoactivite Affiche le logo du type précisé.
\logocorrige
\logocours
\logoTD
\logoflash
\logoDM
\logoDS
\logointerro

6.4.2 Les compteurs

\part Valeur de la \part en cours, également modifiable par le premier argument de la commande \chapitre.
\compteurexo Pour les environnements enonce et correction.
\compteurDM Pour \titreDM.
\compteurDS Pour \titreDS.
\compteurfeuille Pour \titreTD. La valeur est réinitialisée à 1 lors de l'utilisation de \parametres.
\page Numéro de la page en cours.
\enumi Valeur de la liste (enumerate) en cours.

On rappelle ci-dessous quelques commandes générales permettant de manipuler les compteurs :

\setcounter{<nom>}{<nombre>} Réinitialise la valeur du compteur <nom> à <nombre>.
\stepcounter{<nom>} Incrémente le compteur <nom> de 1.
\the<nom> Accède à la valeur du compteur, par exemple \thepage ou \thepart

6.4.3 Les booléens

Chaque paramètre défini dans la graine ou dans les commandes \parametres est utilisable dans les environnements afin de décider de l'affichage ou non du contenu en fonction du paramètre choisi. Par défaut, les booléens sont initialisés à \true au début du document, uniquement s'ils ont été sélectionnés dans la graine de départ. Comme expliqué dans la section dédiée [REF], la commande \parametres permet de réinitialiser les principaux booléens, tandis que la commande \parametres* conserve

leur état. Les clés et valeur choisies dans les commandes \parametres permettent de changer l'état des booléens de \true à \false ou inversement afin de modifier l'affichage des commandes et environnements.

On rappelle qu'en LaTeX, la structure classique d'utilisation d'un booléen est :

```
\ifmonboolen
  %Contenu si monboolen est true
\else
  %Contenu si monbooleen est false
\fi
```

Voici deux booléens qui ne sont pas modifiables en cours de document :

\iffiche Teste si le document est en format fiche.
\ifdiapo Teste si le document est en format diapo.

Voici les booléens pouvant être modifiés uniquement par la commande \parametres :

\ifheader Teste si le titre est affiché automatiquement au début de chaque page.
\ifactivite Teste si le type de titre choisi est celui de \titreactivite.
\ifcorrige Teste si le type de titre choisi est celui de \titrecorrige.
\ifcours Teste si le type de titre choisi est celui de \titrecours.
\ifTD Teste si le type de titre choisi est celui de \titreTD.
\ifflash Teste si le type de titre choisi est celui de \titreflash.
\ifDM Teste si le type de titre choisi est celui de \titreDM.
\ifDS Teste si le type de titre choisi est celui de \titreDS.
\ifinterro Teste si le type de titre choisi est celui de \titreinterro.
\ifbasique Teste si aucun type de titre n'est choisi.

Voici les autres booléens pouvant être modifiés par les commandes \parametres et \parametres* :

\ifenonce Teste si l'option \enonce est sélectionnée.
\ifcorrection Teste si l'option \correction est sélectionnée.
\iftheme Teste si l'option \theme est sélectionnée.
\ifdifficulte Teste si l'option \difficulte est sélectionnée.
\ifcompetence Teste si l'option \competence est sélectionnée.
\ifsource Teste si l'option \source est sélectionnée.

Temporary page !

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.