

cpgekit

Kit LaTeX pour la production de documents destinés aux élèves des classes préparatoires. Il permet de traiter des documents de cours, des planches d'exercices, des devoirs et éventuellement des corrigés des épreuves écrites de concours. Il est prévu d'y ajouter un support pour les fiches de colle ainsi que celui des quiz.

Fonctionnalités gérées :

- thématable (à la Beamer) ;
- production conditionnelle des démonstrations et/ou des solutions ;
- séparation encore plus poussée entre sémantique et éléments de présentation ;
- différents formats pour la sortie PDF (impression ou visionnement sur écran) ;
- insertion déportée des démonstrations/solutions et création automatique de liens hypertextes entre elle et leurs énoncés ;
- structures pour faciliter la gestion des numérotations et leurs styles (sections, listes ...) ;
- différents type de listes, les usuels `enumerate` et `itemize` repensés, mais aussi d'autres listes pour les discussions, mise en valeur des conditions/conséquences...
- une hiérarchie des sections plus adaptée à un document de devoir avec gestion intégrée des tables de matières et affichage dans les tête/pieds de page ;
- capacité de gestion de documents longs avec des méthodes d'inclusion intelligentes de fichiers sources ;
- des structures efficaces pour la saisie de texte mathématique (modes display spécifiques, délimiteurs, dérivation, cadres pour les résultats ...)
- possibilité d'insérer de petits morceaux de texte normal à gauche et/ou à droite d'une formule en mode display ;
- extensions des commandes usuelles du mode mathématique ;
- ...

ToDo

- Guide d'installation (TeXLive et MikTeX) ;
- tutoriaux LaTeX incluant les mécanismes mis en places ;
- Guide de référence du kit.

Stratégie d'inclusion/compilation dans un fichier de devoir

- un fichier de devoir devrait utiliser la classe `cpgedev`

```
\documentclass[<options>]{cpgedev}
```

- La commande `\cpgeinclude` prend en charge l'inclusion de fichiers sources dans un document maître.
- dans un fichier de contenu, un énoncé doit être saisi dans l'un des environnements prévus pour traiter les énoncés en précisant un identifiant unique et son corrigé dans l'environnement commun `corrige` avec le même identifiant. Pour traiter les énoncés plusieurs environnement sont utilisables : `epreuve`, `enonce`, `probleme`, `exercice`... et il est possible d'en créer facilement d'autres.

```
\begin{enonce}{<identifiant>}
... contenu de l'énoncé ...
\end{enonce}
```

et ensuite

```
\begin{corrige}{<identifiant>}
<solution de l'énoncé de même identifiant>
\end{corrige}
```

Les questions sont saisies dans un environnement de liste spécifique qui s'utilise comme `enumerate` et qui porte le nom de `questions`. À la place de la commande `\item`, il faudra utiliser `\xques` pour les questions qui possèdent des solutions et `\zques` pour celles qui n'en ont pas. Il est aussi possible de mixer questions et solutions dans l'environnement de l'énoncé en faisant suivre chaque question d'un environnement `solution` qui contient la solution. Ces solutions seront collectées pour un usage ultérieur. Mais quand on rédige les solution dans un environnement `corrige` il suffit de mettre la commande `\xsol` avant chaque réponse sans besoin d'utiliser un environnement particulier pour cela. Dans ce deuxième cas, l'environnement de l'énoncé aura créé une liste interne de toute les questions `\xques` et l'environnement de corrigé va piocher dans cette liste à chaque `\xsol`.

```
\begin{enonce}{ID}
... texte d'introduction ...
\begin{questions}
\xques .. question 1 ...
\zques ... unité 2 sans question ...
\begin{questions}
\xques ... question 2.a ...
\xques ... question 2.b ...
\end{questions}
\xques ... question 3 ...
\end{questions}
\end{enonce}
\begin{corrige}{ID}
\xsol ... solution de la question 1 ...
\xsol ... solution de la question 2.a ...
\xsol ... solution de la question 2.b ...
\xsol ... solution de la question 3 ...
\end{corrige}
```

ou bien

```
\begin{enonce}{ID}
... texte d'introduction ...
\begin{questions}
  \xques ... question 1 ...
  \begin{solution}
    .... solution de la question 1 ...
  \end{solution}
  \zques ... unite 2 sans question ....
  \begin{questions}
    \xques ... question 2.a ...
    \begin{solution}
      .... solution de la question 2.a ...
    \end{solution}
    \xques ... question 2.b ...
    \begin{solution}
      .... solution de la question 2.b ...
    \end{solution}
  \end{questions}
  \xques ... question 3 ...
  \begin{solution}
    .... solution de la question 3 ...
  \end{solution}
\end{questions}
\end{enonce}
```

- Si l'option **straight** de la classe est active (**\straighttrue**) la définition de l'environnement **corrige** est changée pour que son contenu soit compilé et immédiatement intégré au fichier PDF ;
- Si l'option **straight** de la classe n'est pas active (**\straightfalse**) alors :
 - l'environnement **corrige** se contente d'écrire son contenu dans un fichier dont le nom est **<identifiant>-cor** ;
 - c'est l'environnement **enonce** qui est responsable de l'inclusion ou non du fichier de corrigé avec le même identifiant selon les options de la classe ;
 - avec l'option **solution** de la classe, dès que l'environnement **enonce** a fini de traiter son contenu, il vérifie si un fichier de corrigé qui a le même identifiant existe et a une taille non nulle. Le cas échéant il fait appel à la commande **\Solution** pour intégrer le contenu de ce dernier fichier dans le PDF. Si ce fichier n'a pas été rendu disponible auparavant, **\Solution** se contente de s'enregistrer elle même dans une autre commande avec un nom **\Solution@<identifiant>**. Si plus tard le compilateur rencontre un environnement **corrige** avec un même identifiant que l'énoncé, celui-ci écrit son contenu dans un fichier et fait appelle à la commande **\Solution@<identifiant>** pour insérer son contenu dans le PDF ;
 - avec l'option **solution*** de la classe, chaque environnement **enonce** ou similaire (**epreuve, probleme, exercice...**) enregistre une entrée **\Solution** qui lui est spécifique dans un fichier spécial (**\jobname.rec**). Plus tard c'est la commande

`\corrige`s insérée à un endroit arbitraire choisi par le rédacteur qui inclus le contenu de `\jobname.rec` et le vide pour une éventuelle nouvelle séquence.

- sans l'une des deux options `solution` ou `solution*` l'opération d'inclusion des fichiers de corrigés est ignorée.

Remarques

Le comportement avec l'option `straight` est limpide et suffit pour des situations simples. Dans le cas où elle n'est pas utilisée le système devient plus complexe mais devient beaucoup plus souple en contre partie.