



QCM de mathématiques

Objectif : aider à la conception de nouvelles questions de type qcm.

Voici une courte description de la structure d'une question en \LaTeX et de son passage vers d'autres formats. Le but est d'aider à la création d'une base de données des qcm de maths en essayant de définir une structure commune, et de faciliter l'exportation vers différents formats (papier ou web).

Ce qui n'est *pas* le but ici : gérer de beaux questionnaires papiers (c'est le but ACM), ni des questionnaires web (moodle et autres le font). De plus, aucun élément de barème n'apparaît dans l'énoncé des questions/réponses.

1 Usage basique

Une question à choix multiple (QCM) comporte une question et une liste (finie!) de réponses possibles.

Question 1

Combien font 2^{10} ?

- ☐ [Faux] 1000
- ☐ [Vrai] 1024
- ☐ [Faux] 2048

Qui se code par :

```
\begin{question}
Combien font  $2^{10}$  ?
\begin{answers}
  \bad{1000}
  \good{1024}
  \bad{2048}
\end{answers}
\end{question}
```

Usage possible prévu :

- Export vers le format structuré `yaml`.
- Export vers AMC.
- Export vers des formats web : `xml`, `moodle`, `scenarii`,...

2 Options

Titre. Exemple : `\begin{question}[Théorème de Thalès]`

Il est déconseillé d'utiliser des maths dans le titre, car les maths ne sont pas supportés après exportation par certaines plateformes (ex. scenari).

Identifiant. Exemple `\qid{1234}` ou `\qid{lille1.L2.integrale.33}`. Doit être unique ! Les jeux de caractères sont `a...z`, `A...Z`, `0...9` et les séparateurs `"."` et `"!"`. Le point `"."` sépare des mots clés. Pour indiquer que deux questions testent la compréhension de la même chose (et peuvent être interchangées de façon à avoir des versions différentes pour chaque étudiant) on ajoute à la fin la balise point d'exclamation `"!"`. Par exemple `\qid{derivee.exp!1}` (demande la dérivée de $x \mapsto e^{x^2-8}$) et `\qid{derivee.exp!2}` (demande la dérivée de $x \mapsto e^{3x^3+1}$).

Auteurs. Exemple `\qauthor{exo7}`.

Nom de l'auteur.

Classification. Exemple `\qclassification{127.01,132.08}`.

Fait référence aux catégories d'Exo7.

Tags. Exemple `\qtags{facile, L1, temps=2}`. Non normalisé. Permet(tra) de personnaliser, d'automatiser certaines tâches ou de retrouver des questions. À définir ultérieurement.

Garder l'ordre des réponses. (Non activé par défaut.) Ajouter la commande : `\qkeeporder`. Utile par exemple lorsque les réponses sont dans un ordre naturel et qu'on ne veut pas qu'il soit changé. Par exemple : "Quelle est la date de la révolution française ? 1689, 1789, 1889, 1989 ?"

Case "Je ne sais pas". (Non activé par défaut.) Ajouter la commande : `\qidontknow`. Rajoute une case supplémentaire à la fin : "Je ne sais pas".

Réponses sur une seule ligne. (Non activé par défaut.) Ajouter la commande : `\qoneline`. Cela donne :

Question 2
Combien font 2^{10} ?

☐ [Faux] 1000 ☐ [Vrai] 1024 ☐ [Faux] 2048

Explications. Ajouter des explications après les réponses à l'aide de `\begin{explanations}` ... `\end{explanations}`

Images. Avec la commande `\qimage{monfichier}`, qui est exactement la commande `includegraphics`, avec possibilité d'utiliser les options usuelles, par exemple `\qimage[height=3cm]{monfichier}`. Dans la philosophie actuelle, c'est une seule image par question, et en plus l'image est à la fin de l'énoncé. (Pas d'image dans les réponses, ni dans les choix).

Commentaires. Vous pouvez insérer des commentaires dans votre fichier \LaTeX , mais ils ne seront pas exportés.

3 Exemple sophistiqué

Question 3 (Somme des entiers)

Combien vaut $\sum_{k=2}^n k$?

- ☐ [Faux] $\frac{n(n+1)}{2}$.
- ☐ [Faux] $\frac{n(n-1)}{2}$.
- ☐ [Vrai] $\frac{(n+2)(n-1)}{2}$.
- ☐ I don't know

[id: 5555] [author: exo7] [class: 127.01,132.08]

Explanations: Attention, la somme démarre à 2 et pas à 1 !

```
\begin{question}[Somme des entiers]
\qid{5555}
\qauthor{exo7}
\qclassification{127.01,132.08}
\qidontknow
```

Combien vaut $\sum_{k=2}^n k$?

```
\begin{answers}
  \bad{\frac{n(n+1)}{2}}$.}

  \bad{\frac{n(n-1)}{2}}$.}

  \good{\frac{(n+2)(n-1)}{2}}$.}

\end{answers}
```

```
\begin{explanations}
Attention, la somme démarre à 2 et pas à 1 !
\end{explanations}
```

```
\end{question}
```

4 Type de question/réponses

Il est possible de préciser le type de réponse(s) attendue(s) par l'option `\qtype{...}`.

- **onetoall**. **Option par défaut**. Une ou plusieurs réponses (voir toutes) sont bonnes.
- **onlyone**. Une unique réponse est vraie
- **zerotoall**. Zéro ou plusieurs réponses (voir toutes) sont bonnes. (Avis personnel : à déconseiller !)
- **truefalse**. Seulement deux choix possibles : vrai ou faux (compatible avec un troisième choix, fourni par l'option `qidontknow`).
- **open**. Question ouverte.

5 Conversion YAML

Le format *human-friendly* par défaut est donc la structure L^AT_EX décrite ci-dessus. Cependant ce n'est pas une format *computer-friendly*. Le format choisi pour une question facilement exploitable par un ordinateur est `yaml`.

Voici le code `yaml` d'une question :

```
---
question: |
    Combien font  $(2^{10})$  ?

answers:
  - value: |
      1000
    correct: False

  - value: |
      1024
    correct: True

  - value: |
      2048
    correct: False
```

Conversion L^AT_EX vers `yaml` :

```
python3 latextoyaml.py toto.tex
```

6 Autres conversions

Depuis le format `yaml`, il est facile de convertir une question en à peu près n'importe quoi :

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} \longleftrightarrow \text{yaml} \longrightarrow \text{amc}, \text{xml}, \text{moodle}, \dots$

— Conversion `yaml` vers L^AT_EX :

```
python3 yamltoall.py toto.yaml newname.tex
```

Ou plus simplement `python3 yamltoall.py toto.yaml`

— Conversion `yaml` vers autre, exemple avec `amc` :

```
python3 yamltoall.py -f amc toto.yaml newname.amc
```

Ou plus simplement `python3 yamltoall.py -f amc toto.yaml`

— Conversion `yaml` vers moodle : `yamltoall.py -f moodle toto.yaml`

— Conversion `yaml` vers scenari : `yamltoall.py -f f2s toto.yaml`

Par contre à l'exportation vous pouvez perdre de l'information (auteur, tags, classification,...)