

DocAléa JinjaPyLaTeX

Après avoir créé un modèle, le professeur de mathématiques pourra obtenir différentes versions de ses exercices avec des données différentes et les corrigés

Prérequis

Sous Windows

- [Python 3](#) (à installer "pour tous les utilisateurs")
- Ouvrir "l'invite de commande" en tant qu'administrateur (avec cortana, chercher "cmd" puis cliquer droit sur "Invite de commandes" puis "Exécuter en tant qu'administrateur").
- Installer Jinja2 : dans l'invite de commande taper

```
"C:\Program Files (x86)\Python36-32\pip3 install Jinja2" .<!--(chemin approximatif à préciser) -->
```

Sous Mac Os

- [Python 3](#) (à installer)
- Installer Jinja2 : dans un terminal taper `(sudo) pip3 install Jinja2` (FACULTATIF : lorsque Python2 est installé sur l'ordinateur, vous pouvez saisir `pip install Jinja2` pour rendre jinja disponible dans cet environnement)

Création du modèle

Le fichier `.tex` peut contenir du code qui sera interprété par Jinja2

- Bloc : `## Bloc ##`
- Variables et calculs `<< N >>`
- Commande sur une ligne : `%% Commande`
- Commentaire sur plusieurs lignes : `\#{ Commentaire }`
- Commentaire court : `## Commentaire`

En-tête

Il ne faut pas écrire `\begin{document}` (ni `\end{document}`), le script va chercher un fichier `preamble-perso.tex`, s'il le trouve il sera inséré au début du document sinon il insèrera :

```
\documentclass[a4paper,11pt,fleqn]{article}
\input{preamble}
\begin{document}
\pagestyle{empty}
```

Question

Pour générer des questions aléatoires, on peut commencer par utiliser des variables prédéfinies n et m (voir ci-dessous) dans le document exemple.tex :

```
Calculer :  $\langle n[0] \rangle \times \langle m[0] \rangle = \ldots$ 
```

Correction

Pour générer une correction au fichier exemple.tex, créer un fichier exemple-cor.tex avec le contenu :

```
Le résultat est :  $\langle n[0] \rangle \times \langle m[0] \rangle = \langle n[0] * m[0] \rangle$ 
```

Boucle

On peut transformer la question précédente pour générer quatre questions (de 0 à 3) aléatoires.

```
\begin{itemize}
%% for i in range(0,3)
  \item Calculer :  $\langle n[i] \rangle \times \langle m[i] \rangle = \ldots$ 
%% endfor
\end{itemize}
```

Le corrigé est alors adapté en conséquence :

```
\begin{itemize}
%% for i in range(0,3)
  \item Le résultat est :  $\langle n[i] \rangle \times \langle m[i] \rangle = \langle n[i] * m[i] \rangle$ 
%% endfor
\end{itemize}
```

Test

```
%% if i==1
    blabla
%% elif i==2
    blibli
%% else
    blublu
%% endif
```

Affectation

```
%% set titre = 'Mon Titre'
```

[Documentation Jinja2](#)

Variables prédéfinies

Il s'agit systématiquement de tableaux de 100 nombres (indice de 0 à 99) qui doivent donc être utilisés sous la forme `<<N[indice]>>` dans le document LaTeX.

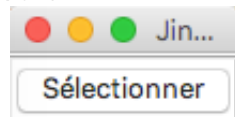
- N : entier entre 1 et 9
- M : entier entre 1 et 9
- n : entier entre 2 et 9
- m : entier entre 2 et 9
- nZ : entier relatif dont la valeur absolue est entre 2 et 9
- mZ : entier relatif dont la valeur absolue est entre 2 et 9
- N3 : 1, 2 ou 3
- N6 : 1, 2, 3, 4, 5 ou 6
- NE : entier entre 1 et 9
- Z : relatif entre -10 et 10
- Z2 : relatif entre -10 et 10
- ZE : relatif non nul entre -10 et 10
- ZE2 : relatif non nul entre -10 et 10
- DCM : 10, 100 ou 1 000
- DCM2 : 10, 100 ou 1 000
- D : 10, 20, 30...90
- S : -1 ou 1
- NN : entier entre 1 et 99
- NNN : entier entre 1 et 999
- NNO : 10, 20, 30... 990
- h : entier entre 1 et 12
- H : entier entre 1 et 24

- min : entier entre 1 et 60
- s : entier entre 1 et 60
- L1, L2, L3 et L4 : 4 lettres majuscules différentes
- prenom : un prénom au hasard (on n'a pas encore distingué les prénoms masculins et féminins)

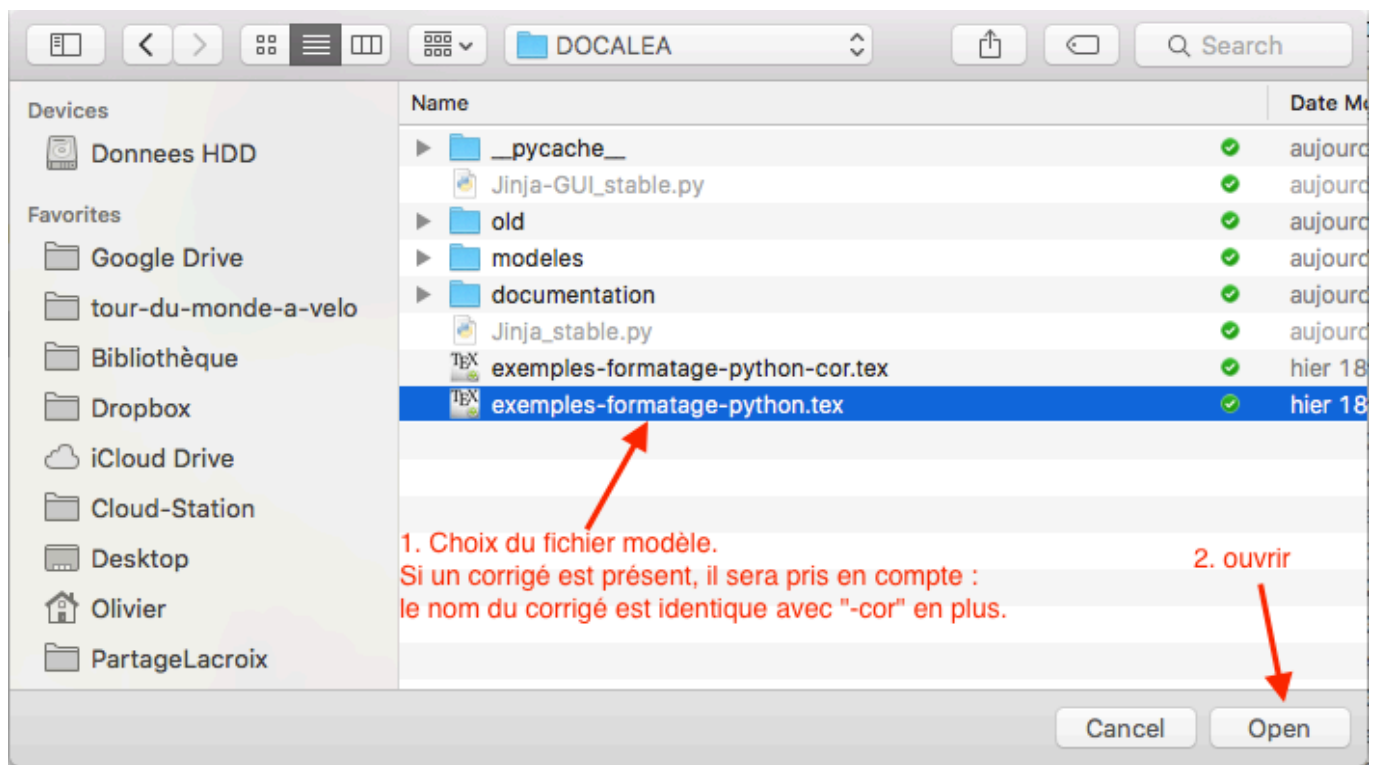
Utilisation d'un modèle pour générer des sujets aléatoires

L'utilisation de la version graphique du script python est recommandée.

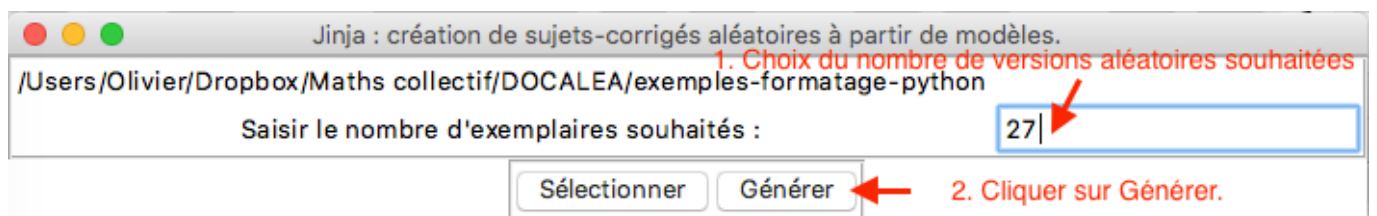
- Exécuter Jinja-GUI_stable.py (F5 dans l'éditeur IDLE)



- Cliquer sur "sélectionner" :
- Boîte de sélection du fichier tex :



- Choix du nombre de versions aléatorisées souhaitées :



- Le fichier aléatoire créé est placé dans le même dossier que le document d'origine. Il portera le même nom avec "_aleatoirise" à la fin. Il restera à effectuer la compilation LaTeX avec votre éditeur préféré.

Fonctions

terme(a)

- `terme(0)` => `0`
- `terme(3)` => `3`
- `terme(-4)` => `-4`

Exemple d'utilisation dans des calculs sur les relatifs :

```
\begin{itemize}
%% for i in range(0,3)
  \item $<nZ[i]>> <<terme(mZ[i])>>=\ldots$
%% endfor
\end{itemize}
```

Le signe sera systématiquement explicite pour les nombres `mZ[i]`, ce qui est souhaitable pour une addition de relatifs. Ce code génèrera des questions de la forme `3 - 4 = ...`.

facteur(a)

- `facteur(3)` => `3`
- `facteur(-4)` => `(-4)`
- `facteur(0)` => `0`

Exemple d'utilisation dans des calculs sur les relatifs :

```
\begin{itemize}
%% for i in range(4,8)
  \item $<nZ[i]>>\times<<facteur(mZ[i])>>=\ldots$
%% endfor
\end{itemize}
```

Les parenthèses seront ajoutées, uniquement si `mZ[i]` est négatif. Ce code génèrera des questions de la forme `-3 × (- 4) = ...` ou `-5 × 3=...`, etc...

prix(a)

Affiche le nombre entier (s'il est entier) ou (s'il est décimal) avec deux décimales et la virgule en séparateur décimale.

Attention, le résultat est de type string

- `prix(54)` => `54`
- `prix(54.3)` => `54,30`