

# TexAlea

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

```
cd "C:\Program Files (x86)\Python36-32" puis pip3 install Jinja2 .
```

### Sous Mac Os

- [Python 3](#) (à installer)
- Installer Jinja2 : dans un terminal taper `(sudo) pip3 install Jinja2`

## 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 automatiquement insérer :

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

Lorsqu'on souhaite personnaliser le fichier LaTeX, on peut écrire, dans le même dossier, un fichier `preamble-perso.tex` qui sera utilisé comme préambule de tous les documents du dossier. On peut aussi créer un préambule personnalisé pour chaque modèle de fichier .tex en ajoutant "-preamble" au nom du fichier. Exemple : pour un fichier modèle "DS.tex", le fichier préambule "DS-preamble.tex" sera utilisé en priorité sur le préambule `preamble-perso.tex` commun au dossier, lui même prioritaire sur le préambule par défaut.

Remarque : si un fichier corrigé est généré, il prend en compte le même préambule que le fichier d'origine (inutile de dupliquer celui-ci).

## 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 :  $\llbracket n[0] \rrbracket \times \llbracket m[0] \rrbracket = \ldots$ 
```

## Correction

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

```
Le résultat est :  $\llbracket n[0] \rrbracket \times \llbracket m[0] \rrbracket = \llbracket n[0] * m[0] \rrbracket$ 
```

## 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 :  $\llbracket n[i] \rrbracket \times \llbracket m[i] \rrbracket = \ldots$ 
%% endfor
\end{itemize}
```

Le corrigé est alors adapté en conséquence :

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

## Test

---

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

## Affectation

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

## Changement de version.

Entre deux versions du document aléatoirisé, sont insérées les lignes suivantes :

```
\newpage
\setcounter{section}{0}
```

Cela peut s'avérer insuffisant dans le cas d'utilisation de commandes personnalisées pour gérer correctement les numéros d'exercices. Dans ce cas, il est possible de créer un fichier "changement-version.tex" dans le même dossier que jinja-stable.py. C'est alors celui-ci qui sera utilisé pour les changements de versions de documents.

## Corrigé des documents générés

Si un modèle se nomme "exemple.tex", la présence dans le même dossier d'un "exemple-cor.tex" sera détectée automatiquement. Le même jeu de variables sera utilisé, ce qui permet de créer un corrigé pour le document en question. La syntaxe du corrigé est la même que celle du document principal.

## [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
- N2 : 1 ou 2
- N3 : 1, 2 ou 3
- N4 : 1, 2, 3 ou 4
- N5 : 1, 2, 3, 4 ou 5
- N6 : 1, 2, 3, 4, 5 ou 6
- N7 : 1 à 7
- N8 : 1 à 8
- N9 : 1 à 9
- N10 : 1 à 10
- 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, L4, ..., L9 : des lettres majuscules différentes
- prenom : un prénom au hasard
- garçon : un prénom de garçon au hasard
- fille : un prénom de fille au hasard

## Utilisation d'un modèle pour générer des sujets aléatoires

---

### Jinja\_stable.py

L'utilisation de la version non graphique Jinja\_stable.py est recommandée pour des tests. ATTENTION : elle n'utilise des modèles que s'ils sont présents dans un sous-dossier "modeles" de l'arborescence.

```

===== RESTART: /Users/Olivier/GitHub/TeXAlea/Jinja_stable.py =====
0 - modeles/conversions.tex
1 - modeles/demo.tex
2 - modeles/durees.tex
3 - modeles/exemples-formatage-python.tex
4 - modeles/multiplications_prefixes.tex
5 - modeles/pourcentages.tex
6 - modeles/relatifs.tex
7 - modeles/test.tex
8 - modeles/test1.tex
9 - modeles/test_r_mem.tex
10 - modeles/calcul-mental/multiplications.tex
11 - modeles/fractions/_fractions-niveau1-v1.tex
12 - modeles/fractions/preambule-perso.tex
Nom du fichier modèle (sans extension) ou numéro de celui-ci :3
Fichier choisi : modeles/exemples-formatage-python
Nombre d'exemplaires souhaités :27
Création du fichier aléatoirisé en 27 version(s) réalisée.

```

1. Choisir l'un des fichiers listés.

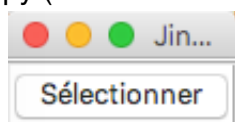
2. Choisir le nombre d'exemplaires.

Le fichier aléatoire créé est placé dans le dossier "fichiers-aleatoirises", sous-dossier de celui contenant Jinjastable.py. Il portera le même nom que le modèle d'origine avec "aleatoirise" à la fin. Il restera à effectuer la compilation LaTeX avec votre éditeur préféré.

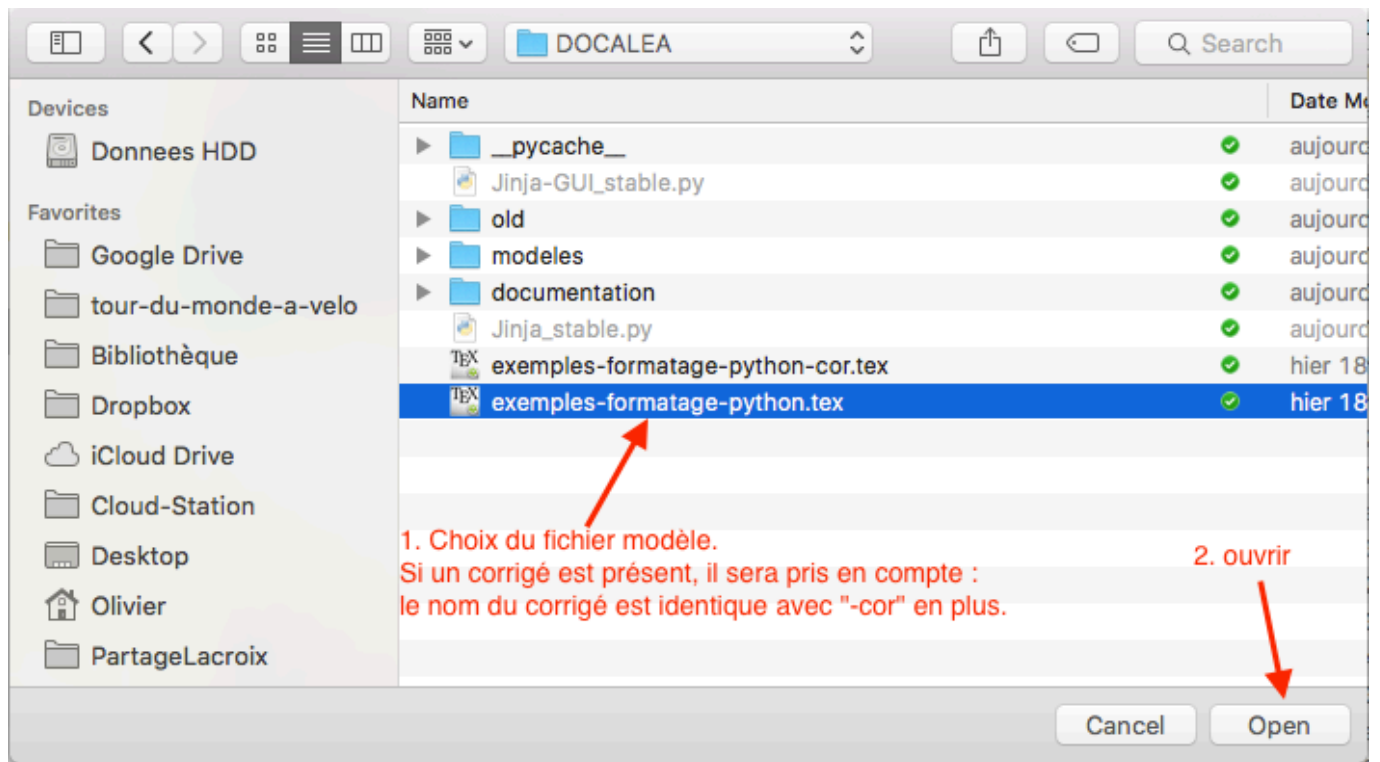
## Jinja-GUI\_stable.py

L'utilisation de la version graphique du script python est recommandée pour utiliser des modèles discéminés sur votre ordinateur.

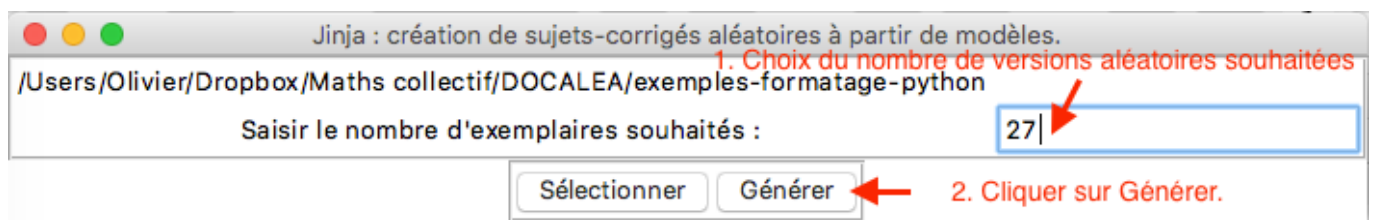
- Exécuter Jinja-GUI\_stable.py (F5 dans l'éditeur IDLE)



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



- Choix du nombre de versions aléatoirées souhaitées :



- Le fichier aléatoire créé est placé dans le même dossier que le modèle d'origine. Il portera le même nom que le modèle d'origine 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 générera des questions de la forme  $3 - 4 = \dots$ .

## 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  $\ll nZ[i] \gg \times \ll \text{facteur}(mZ[i]) \gg = \ldots$ 
%% endfor
\end{itemize}
```

Les parenthèses seront ajoutées, uniquement si  $mZ[i]$  est négatif. Ce code générera des questions de la forme  $-3 \times (-4) = \dots$  ou  $-5 \times 3 = \dots$ , 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`

## HMS(h,m,s)

Affiche la durée en mode mathématique au format HMS.

```
HMS(2,31,15) =>  $\$2\sim\text{h}\sim31\sim\text{min}\sim15\sim\text{s}\$$ 
```

## ecriture\_decimale(a)

Formate un nombre décimal obtenu aléatoirement pour un affichage correct dans le document.

Exemple : `\ll NN[1]/10 \gg` est un nombre décimal compris entre 0,1 et 9,9 par incréments de 0,1. Le séparateur décimal est un point. Si ce nombre est entier, il est affiché sous la forme `x.y` ( `3.0` par exemple). `\ll ecriture_decimale(NN[1]/10) \gg` générera comme séparateur décimal la virgule et supprime la partie décimale, si elle est nulle.

Des options existent (le deuxième argument peut être 1 (par défaut), 2 ou 3 :

- `<<ecriture_decimale(12.345,2)>>` => écrit le nombre sous forme d'une fraction décimale.
- `<<ecriture_decimale(12.345,3)>>` => écrit le nombre sous forme de la somme de la partie entière et d'une fraction décimale correspondant à la partie décimale.