

devoir.py

Nicolas Mesnier

25 février 2023

`devoir.py` est un script écrit en python 3 qui permet d'initialiser un rapport d'épreuve à partir d'une feuille de calcul libreoffice et d'ajouter en fin de correction les statistiques du devoir avant publication en PDF. C'est un logiciel libre, distribué sous licence [GPL v3](#).

Table des matières

Introduction	2
Téléchargement	2
Installation	2
Introduction	2
GNU/Linux (Debian et filles rebelles...) ou Mac OS X	2
Configuration	2
Utilisation	3
Ressources	5

Introduction

`devoir.py` est un script écrit en python 3 qui permet d'initialiser un rapport d'épreuve à partir d'une feuille de calcul libreoffice et d'ajouter en fin de correction les statistiques du devoir avant publication en PDF.

`devoir.py` permet de :

- d'initialiser un rapport d'épreuve au format markdown avec autant de sections que de parties avec chacune autant de sous-sections que de questions définies dans la feuille de calcul ;
- d'ajouter les statistiques par question au rapport puis le compiler en PDF pour publication.

`devoir.py` s'appuie sur :

- une distribution python 3 avec les bibliothèques standard `os`, `sys`, `subprocess`, `shutil`, `glob`, `re`, `argparse`, `datetime`, `math` et `textwrap` ;
- une distribution LaTeX, par exemple [TeX Live](#) pour la commande `pdflatex` ;
- le script [panpdf](#) qui s'appuie sur le convertisseur de documents [pandoc](#) pour générer le rapport en PDF.

C'est un logiciel libre, distribué sous licence [GPL v3](#).

Téléchargement

La dernière version est la 0.1.6 (25/02/2023).

Type	Fichier
code python	devoir.py
feuille de calcul	devoir.ods
documentation	devoir.pdf

Installation

Introduction

Pour utiliser `devoir.py`, il est nécessaire d'avoir sur son ordinateur personnel :

- une distribution python 3 ;
- une distribution LaTeX ;
- [pandoc](#) ;
- [panpdf](#) qui est écrit en [bash](#).

GNU/Linux (Debian et filles rebelles...) ou Mac OS X

1. Copier le fichier `devoir.py` dans le répertoire `/home/$(whoami)/bin` ;
2. Le rendre exécutable

```
$ chmod +x devoir.py
```

3. Copier le fichier `devoir.ods` dans un répertoire « stable » puis renseigner dans le fichier `devoir.py` sa localisation, par exemple :

```
feuille = /home/$(whoami)/bin/devoir.ods
```

4. Modifier la matière par défaut (paramètre `matiere`), les classes (paramètre `classes`) ou les abréviations définies par défaut (dictionnaire `short`) selon vos besoins.

Configuration

Par défaut, la matière des devoirs est définie ligne 46 comme :

```
matiere = "Sciences industrielles de l'ingénieur"
```

Vous pouvez évidemment la modifier selon vos souhaits. Sont pour l'instant définies les matières :

```
short={  
  "Sciences industrielles de l'ingénieur": "SII",  
  "Informatique commune": "Info",  
  "Mathématiques": "Maths",  
  "Physique-Chimie": "PC",  
}
```

où les valeurs dans le dictionnaire sont les diminutifs des noms complets donnés en clés. Les clés servent au titre du rapport, et les diminutifs aux noms de fichiers.

Les types sont renseignés ligne 48 :

```
typesEval = ["DS", "DM", "I"]
```

Outre le titre des rapports, ils permettent de moduler le recalage des notes dans le cas d'un DS.

Utilisation

1. Copier le fichier `devoir.ods` dans le répertoire courant avec le nom `<classe>-<typeEval><num>.ods` :

```
$ devoir.py -f -c <classe> -t <typeEval> -n <num>
```

par exemple :

```
$ devoir.py -f -c PCSI -t DS -n 2
```

ou en spécifiant directement le nom de fichier

```
$ devoir.py -f -o PCSI-DS2.ods
```

par défaut

```
$ devoir.py -f
```

copie simplement le fichier `devoir.ods` dans le répertoire courant avec le même nom.

2. Modifier le champ **A1** selon que vous corrigez un **DS** ou un **DM**. Dans le cas d'un **DM** on se satisfait de notes brutes alors que pour un **DS** on les ajustera avec une formule du type

$$N = \left(20 \times \left(\frac{n}{20} \right)^a - M \right) \times \frac{\Sigma}{\sigma} + M, \quad a = \frac{\ln \left(\frac{M}{20} \right)}{\ln \left(\frac{m}{20} \right)}$$

où n est la note brute, N la note modifiée et avec m la moyenne et σ l'écart-type des notes brutes, et M et Σ des estimateurs raisonnables de la moyenne et de l'écart-type de la distribution de notes désirée.

3. Saisissez les questions par partie et le barème ;
4. Copier les noms et prénoms de vos élèves dans les colonnes **A** et **B**. Par défaut, le fichier est calibré pour une classe de 48 élèves.
5. Générer pour chacun un identifiant personnel, par exemple un nombre entier compris entre 100 et 300 avec [ring](#)

```
$ ring 48 100 300 > codes.csv
```

ou [RandomPrime](#)

```
$ RandomPrime 48 100 300 > codes.csv
```

et les ajouter à la colonne **C** ;

6. Exécuter le script `devoir.py` pour initialiser le rapport

```
$ devoir.py -i <classe>-<typeEval><num>.ods
```

qui générera un rapport pour la matière définie par défaut (variable `matiere` ligne 46, voir la section [configuration](#)) ou, si la matière n'est pas celle par défaut :

```
$ devoir.py -i <classe>-<typeEval><num>.ods -m <matiere>
```

Si `<matiere>` est parmi la liste des diminutifs donnés en clés du dictionnaire `short` :

```
short={
    "Sciences industrielles de l'ingénieur": "SII",
    "Informatique commune": "Info",
    "Mathématiques": "Maths",
    "Physique-Chimie": "PC",
}
```

alors le fichier du rapport sera nommé :

```
<matiere>-<classe>-<typeEval><num>_rapport.md
```

sinon il sera nommé :

```
short[<matiere>]-<classe>-<typeEval><num>_rapport.md
```

par exemples :

```
— $ devoir.py -i DS3.ods -m PC
   génère le fichier PC-DS3_rapport.md;
— $ devoir.py -i DM4.ods -m Info
   génère le fichier Info-DM4_rapport.md;
— $ devoir.py -i DS5.ods -m Mathématiques -c PTSI
   génère le fichier Maths-PTSI-DS5_rapport.md.
```

Dans tous les cas, on peut initialiser un rapport de façon complètement générique :

```
devoir.py -i <file.ods> -m <matiere> -c <classe> -t <typeEval> -n <num>
```

7. Corriger les copies en attribuant à chaque question traitée une note en **points entiers entre 0 et 5** (pour éviter toute demie-mesure), son poids étant donné par le barème. Rédiger simultanément le rapport d'épreuve pour capitaliser les erreurs vues, etc. Hélas, je n'ai pas encore de script pour cela...
8. Ajuster les paramètres M (cellule AE11 du fichier `devoir.ods`) et Σ (cellule AE12) pour obtenir l'ajustement des notes (au besoin s'appuyer sur les extrema et l'allure de la bijection).
9. Une fois que les notes satisfont aux critères de votre évaluation, figez les valeurs en copiant les valeurs des cellules AE15 et AE16 respectivement vers celles des cellules AF15 et AF16, puis faire pointer les cellules AD15 et AD16 respectivement vers ces dernières (par défaut elles pointent vers les valeurs calculées, cellules AE15 et AE16). C'est une méthode archaïque, mais elle évite de modifier toutes les notes finales en cas d'erreur d'évaluation d'une copie.
10. Exécuter le script `devoir.py` pour ajouter les statistiques du devoir au rapport

```
$ devoir.py -s <classe>-<typeEval><num>.ods
```

ou, si la matière n'est pas celle par défaut :

```
$ devoir.py -s <classe>-<typeEval><num>.ods -m <matiere>
```

ou de façon complètement générique :

```
$ devoir.py -i <file.ods> -m <matiere> -c <classe> -t <typeEval> -n <num>
```

11. Créer un fichier anonymé en omettant les noms et prénoms, puis le trier par identifiants croissants.

Ressources

1. GPL v3
<https://www.gnu.org/licenses/gpl-3.0.html>
2. TeX Live
<http://tug.org/texlive/>
3. pandoc
<https://pandoc.org/>
4. bash
<https://www.gnu.org/software/bash/>
5. panpdf
<http://nmesnier.free.fr/files/panpdf>
6. ring
<http://nmesnier.free.fr/files/ring>
7. RandomPrime
<http://nmesnier.free.fr/files/RandomPrime.c>